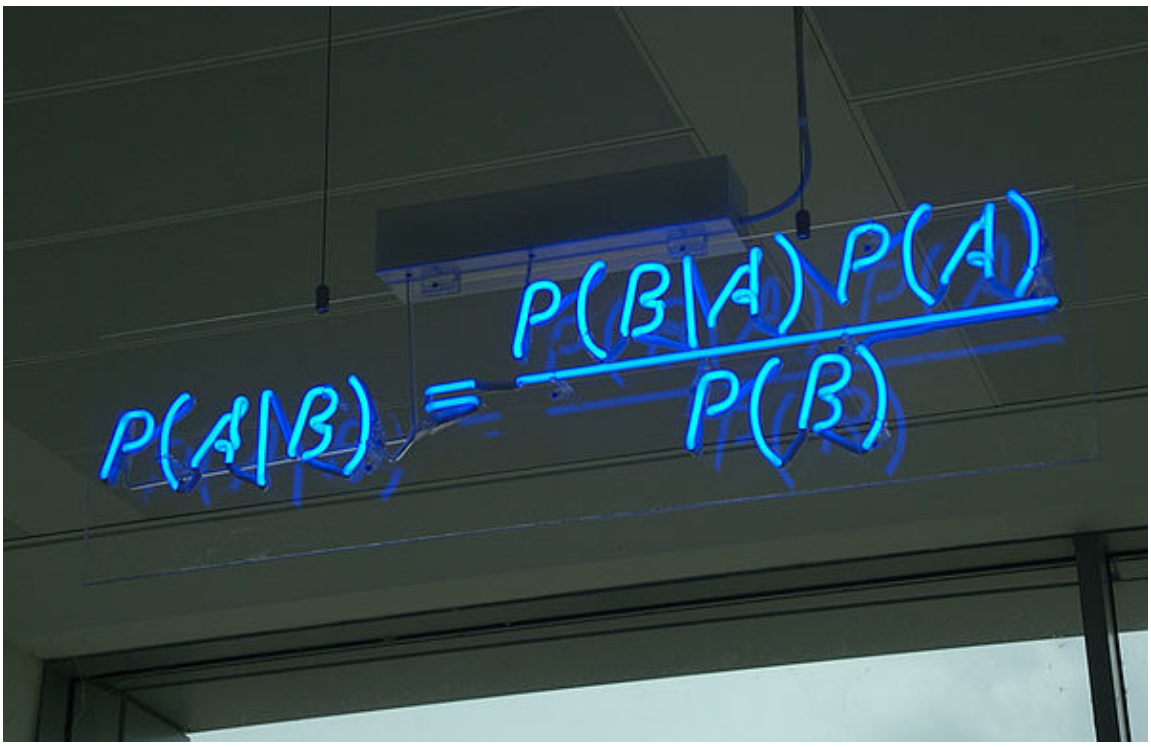
Graphical Models

see more at http://ml.memect.com

Contents

Chapter 1

Bayes' theorem



A blue neon sign, showing the simple statement of Bayes' theorem

In probability theory and statistics, **Bayes' theorem** (alternatively **Bayes' law** or Bayes' rule) relates current probability to prior probability. It is important in the mathematical manipulation of conditional probabilities.

When applied, the probabilities involved in Bayes' theorem may have different interpretations. In one of these interpretations, the theorem is used directly as part of a particular approach to statistical inference. In particular, with the Bayesian interpretation of probability, the theorem expresses how a subjective degree of belief should rationally change to account for evidence: this is Bayesian inference, which is fundamental to Bayesian statistics. However, Bayes' theorem has applications in a wide range of calculations involving probabilities, not just in Bayesian inference.

Bayes' theorem is named after Rev. Thomas Bayes (/ˈbeɪz/; 1701–1761), who first showed how to use new evidence to update beliefs. It was further developed by Pierre-Simon Laplace, who first published the modern formulation in his 1812 *Théorie analytique des probabilités*. Sir Harold Jeffreys put Bayes' algorithm and Laplace's formulation on an axiomatic basis. Jeffreys wrote that Bayes' theorem "is to the theory of probability what Pythagoras's theorem is to geometry".[1]

1.1 Statement of theorem

Bayes' theorem is stated mathematically as the following equation:[2]

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)},$$

where A and B are events.

- $P(A)$ and $P(B)$ are the probabilities of A and B without regard to one other.
- $P(A \mid B)$, a conditional probability, is the probability of A given that B is true.
- $P(B \mid A)$, is the probability of B given that A is true.

1.2 Introductory example

The entire output of a factory is produced on three machines. The three machines account for 20%, 30%, and 50% of the output, respectively. The fraction of defective items produced is this: for the first machine, 5%; for the second machine, 3%; for the third machine, 1%. If an item is chosen at random from the total output and is found to be defective, what is the probability that it was produced by the third machine?

A solution is as follows. Let Ai denote the event that a randomly chosen item was made by the ith machine (for i = 1,2,3). Let B denote the event that a randomly chosen item is defective. Then, we are given the following information:

$P(A_1) = 0.2$, $P(A_2) = 0.3$, $P(A_3) = 0.5$.

If the item was made by machine A_1, then the probability that it is defective is 0.05; that is, $P(B \mid A_1) = 0.05$. Overall, we have

$P(B \mid A_1) = 0.05$, $P(B \mid A_2) = 0.03$, $P(B \mid A_3) = 0.01$.

To answer the original question, we first find $P(B)$. That can be done in the following way:

$P(B) = \Sigma i\ P(B \mid Ai)\ P(Ai) = (0.05)(0.2) + (0.03)(0.3) + (0.01)(0.5) = 0.024$.

Hence 2.4% of the total output of the factory is defective.

We are given that B has occurred, and we want to calculate the conditional probability of A_3. By Bayes' theorem,

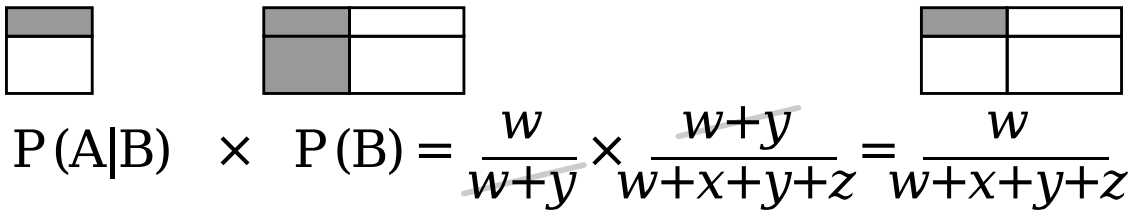
$P(A_3 \mid B) = P(B \mid A_3)\ P(A_3)/P(B) = (0.01)(0.50)/(0.024) = 5/24$.

Given that the item is defective, the probability that it was made by the third machine is only 5/24. Although machine 3 produces half of the total output, it produces a much smaller fraction of the defective items. Hence the knowledge that the item selected was defective enables us to replace the prior probability $P(A_3) = 1/2$ by the smaller posterior probability $P(A_3 \mid B) = 5/24$.

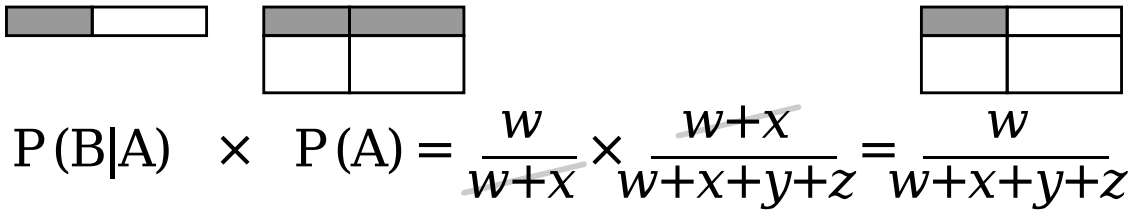
1.3 Interpretations

The interpretation of Bayes' theorem depends on the interpretation of probability ascribed to the terms. The two main interpretations are described below.

Relative size	Case B	Case $\bar{B}$	Total
Condition A	w	x	$w+x$
Condition $\bar{A}$	y	z	$y+z$
Total	$w+y$	$x+z$	$w+x+y+z$



$$P(A|B) \times P(B) = \frac{w}{w+y} \times \frac{w+y}{w+x+y+z} = \frac{w}{w+x+y+z}$$



$$P(B|A) \times P(A) = \frac{w}{w+x} \times \frac{w+x}{w+x+y+z} = \frac{w}{w+x+y+z}$$

B) = P(B $\bar{A}$) P($\bar{A}$) / P(B) etc.

1.3.1 Bayesian interpretation

In the Bayesian (or epistemological) interpretation, probability measures a *degree of belief*. Bayes' theorem then links the degree of belief in a proposition before and after accounting for evidence. For example, suppose it is believed with 50% certainty that a coin is twice as likely to land heads than tails. If the coin is flipped a number of times and the outcomes observed, that degree of belief may rise, fall or remain the same depending on the results.

For proposition A and evidence B,

- $P(A)$, the *prior*, is the initial degree of belief in A.
- $P(A \mid B)$, the *posterior*, is the degree of belief having accounted for B.
- the quotient $P(B \mid A)/P(B)$ represents the support B provides for A.

For more on the application of Bayes' theorem under the Bayesian interpretation of probability, see Bayesian inference.

1.3.2 Frequentist interpretation

In the frequentist interpretation, probability measures a *proportion of outcomes*. For example, suppose an experiment is performed many times. $P(A)$ is the proportion of outcomes with property A, and $P(B)$ that with property B. $P(B \mid A)$ is the proportion of outcomes with property B *out of* outcomes with property A, and $P(A \mid B)$ the proportion of those with A *out of* those with B.

The role of Bayes' theorem is best visualized with tree diagrams, as shown to the right. The two diagrams partition the same outcomes by A and B in opposite orders, to obtain the inverse probabilities. Bayes' theorem serves as the link between these different partitionings.

1.4 Forms

1.4.1 Events

Simple form

For events A and B, provided that $P(B) \neq 0$,

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}.$$

In many applications, for instance in Bayesian inference, the event B is fixed in the discussion, and we wish to consider the impact of its having been observed on our belief in various possible events A. In such a situation the denominator of the last expression, the probability of the given evidence B, is fixed; what we want to vary is A. Bayes' theorem then shows that the posterior probabilities are proportional to the numerator:

$$P(A \mid B) \propto P(A) \cdot P(B \mid A) \text{ (proportionality over } A \text{ for given } B).$$

In words: **posterior is proportional to prior times likelihood**.[3]

If events A_1, A_2, ..., are mutually exclusive and exhaustive, i.e., one of them is certain to occur but no two can occur together, and we know their probabilities up to proportionality, then we can determine the proportionality constant by using the fact that their probabilities must add up to one. For instance, for a given event A, the event A itself and its complement $\neg A$ are exclusive and exhaustive. Denoting the constant of proportionality by c we have

$$P(A \mid B) = c \cdot P(A) \cdot P(B \mid A) \text{ and } P(\neg A \mid B) = c \cdot P(\neg A) \cdot P(B \mid \neg A).$$

Adding these two formulas we deduce that

$$c = \frac{1}{P(A) \cdot P(B \mid A) + P(\neg A) \cdot P(B \mid \neg A)}.$$

Alternative form

Another form of Bayes' Theorem that is generally encountered when looking at two competing statements or hypotheses is:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A)}.$$

For an epistemological interpretation:

For proposition A and evidence or background B,[4]

- $P(A)$,the prior probability, is the initial degree of belief in A.
- $P(-A)$, is the corresponding probability of the initial degree of belief against A: $1 - P(A) = P(-A)$
- $P(B \mid A)$, the conditional probability or likelihood, is the degree of belief in B, given that the proposition A is true.
- $P(B \mid -A)$, the conditional probability or likelihood, is the degree of belief in B, given that the proposition A is false.
- $P(A \mid B)$, the posterior probability, is the probability for A after taking into account B for and against A.

Extended form

Often, for some partition $\{Aj\}$ of the event space, the event space is given or conceptualized in terms of $P(Aj)$ and $P(B \mid Aj)$. It is then useful to compute $P(B)$ using the law of total probability:

$$P(B) = \sum_j P(B \mid A_j)P(A_j),$$

$$\Rightarrow P(A_i \mid B) = \frac{P(B \mid A_i)\, P(A_i)}{\sum_j P(B \mid A_j)\, P(A_j)}.$$

In the special case where A is a binary variable:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A)}.$$

1.4.2 Random variables

Consider a sample space Ω generated by two random variables X and Y. In principle, Bayes' theorem applies to the events $A = \{X = x\}$ and $B = \{Y = y\}$. However, terms become 0 at points where either variable has finite probability density. To remain useful, Bayes' theorem may be formulated in terms of the relevant densities (see Derivation).

Simple form

If X is continuous and Y is discrete,

$$f_X(x \mid Y = y) = \frac{P(Y = y \mid X = x)\, f_X(x)}{P(Y = y)}.$$

If X is discrete and Y is continuous,

$$P(X = x \mid Y = y) = \frac{f_Y(y \mid X = x)\, P(X = x)}{f_Y(y)}.$$

If both X and Y are continuous,

$$f_X(x \mid Y = y) = \frac{f_Y(y \mid X = x)\, f_X(x)}{f_Y(y)}.$$

Extended form

A continuous event space is often conceptualized in terms of the numerator terms. It is then useful to eliminate the denominator using the law of total probability. For $fY(y)$, this becomes an integral:

$$f_Y(y) = \int_{-\infty}^{\infty} f_Y(y \mid X = \xi)\, f_X(\xi)\, d\xi.$$

1.4.3 Bayes' rule

Main article: Bayes' rule

Bayes' rule is Bayes' theorem in odds form.

$$O(A_1 : A_2 \mid B) = O(A_1 : A_2) \cdot \Lambda(A_1 : A_2 \mid B)$$

where

$$\Lambda(A_1 : A_2 \mid B) = \frac{P(B \mid A_1)}{P(B \mid A_2)}$$

is called the Bayes factor or likelihood ratio and the odds between two events is simply the ratio of the probabilities of the two events. Thus

$$O(A_1 : A_2) = \frac{P(A_1)}{P(A_2)},$$

$$O(A_1 : A_2 \mid B) = \frac{P(A_1 \mid B)}{P(A_2 \mid B)},$$

So the rule says that the posterior odds are the prior odds times the Bayes factor, or in other words, posterior is proportional to prior times likelihood.

1.5 Derivation

1.5.1 For events

Bayes' theorem may be derived from the definition of conditional probability:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \text{ if } P(B) \neq 0,$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}, \text{ if } P(A) \neq 0,$$

$$\Rightarrow P(A \cap B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A),$$

$$\Rightarrow P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}, \text{ if } P(B) \neq 0.$$

1.5.2 For random variables

For two continuous random variables *X* and *Y*, Bayes' theorem may be analogously derived from the definition of conditional density:

$$f_X(x \mid Y = y) = \frac{f_{X,Y}(x, y)}{f_Y(y)}$$

$$f_Y(y \mid X = x) = \frac{f_{X,Y}(x, y)}{f_X(x)}$$

$$\Rightarrow f_X(x \mid Y = y) = \frac{f_Y(y \mid X = x)\, f_X(x)}{f_Y(y)}.$$

1.6 Examples

1.6.1 Frequentist example

An entomologist spots what might be a rare subspecies of beetle, due to the pattern on its back. In the rare subspecies, 98% have the pattern, or *P*(Pattern | Rare) = 98%. In the common subspecies, 5% have the pattern. The rare subspecies accounts for only 0.1% of the population. How likely is the beetle having the pattern to be rare, or what is *P*(Rare | Pattern)?

From the extended form of Bayes' theorem (since any beetle can be only rare or common),

$$\begin{aligned} P(\text{Rare} \mid \text{Pattern}) &= \frac{P(\text{Pattern} \mid \text{Rare})P(\text{Rare})}{P(\text{Pattern} \mid \text{Rare})P(\text{Rare}) \ + \ P(\text{Pattern} \mid \text{Common})P(\text{Common})} \\ &= \frac{0.98 \times 0.001}{0.98 \times 0.001 + 0.05 \times 0.999} \\ &\approx 1.9\%. \end{aligned}$$

1.6.2 Drug testing

Suppose a drug test is 99% sensitive and 99% specific. That is, the test will produce 99% true positive results for drug users and 99% true negative results for non-drug users. Suppose that 0.5% of people are users of the drug. If a randomly selected individual tests positive, what is the probability he or she is a user?

$$\begin{aligned} P(\text{User} \mid +) &= \frac{P(+ \mid \text{User})P(\text{User})}{P(+ \mid \text{User})P(\text{User}) + P(+ \mid \text{Non-user})P(\text{Non-user})} \\ &= \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995} \\ &\approx 33.2\% \end{aligned}$$

Despite the apparent accuracy of the test, if an individual tests positive, it is more likely that they do *not* use the drug than that they do.

This surprising result arises because the number of non-users is very large compared to the number of users; thus the number of false positives (0.995%) outweighs the number of true positives (0.495%). To use concrete numbers, if 1000 individuals are tested, there are expected to be 995 non-users and 5 users. From the 995 non-users, 0.01 × 995 ≃ 10 false positives are expected. From the 5 users, 0.99 × 5 ≃ 5 true positives are expected. Out of 15 positive results, only 5, about 33%, are genuine.

Note: The importance of specificity can be illustrated by showing that even if sensitivity is 100% and specificity is at 99% the probability of the person being a drug user is ≈33% but if the specificity is changed to 99.5% and the sensitivity is dropped down to 99% the probability of the person being a drug user rises to 49.8%.

1.7 History

Bayes' theorem was named after the Reverend Thomas Bayes (1701–61), who studied how to compute a distribution for the probability parameter of a binomial distribution (in modern terminology). Bayes' unpublished manuscript was significantly edited by Richard Price before it was posthumously read at the Royal Society. Price edited[5] Bayes' major work *An Essay towards solving a Problem in the Doctrine of Chances* (1763), which appeared in *Philosophical Transactions*,[6] and contains Bayes' Theorem. Price wrote an introduction to the paper which provides some of the philosophical basis of Bayesian statistics. In 1765 he was elected a Fellow of the Royal Society in recognition of his work on the legacy of Bayes.[7][8]

The French mathematician Pierre-Simon Laplace reproduced and extended Bayes' results in 1774, apparently quite unaware of Bayes' work.[9][10] Stephen Stigler suggested in 1983 that Bayes' theorem was discovered by Nicholas Saunderson some time before Bayes;[11] that interpretation, however, has been disputed.[12]

Martyn Hooper[13] and Sharon McGrayne[14] have argued that Richard Price's contribution was substantial:

> By modern standards, we should refer to the Bayes–Price rule. Price discovered Bayes' work, recognized its importance, corrected it, contributed to the article, and found a use for it. The modern convention of employing Bayes' name alone is unfair but so entrenched that anything else makes little sense.
> —[14]

1.8 See also

- Bayesian inference
- Inductive probability
- *Grammar of Assent*
- Probabiliorism

1.9 Notes

[1] Jeffreys, Harold (1973). *Scientific Inference* (3rd ed.). Cambridge University Press. p. 31. ISBN 978-0-521-18078-8.

[2] Stuart, A.; Ord, K. (1994), *Kendall's Advanced Theory of Statistics: Volume I—Distribution Theory*, Edward Arnold, §8.7.

[3] Lee, Peter M. (2012). "Chapter 1". *Bayesian Statistics*. Wiley. ISBN 978-1-1183-3257-3.

[4] "Bayes Theorem: Introduction". *Trinity University*.

[5] Richard Allen (1999). *David Hartley on Human Nature*. SUNY Press. pp. 243–4. ISBN 978-0-7914-9451-6. Retrieved 16 June 2013.

[6] Bayes, Thomas, and Price, Richard (1763). "An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S." (PDF). *Philosophical Transactions of the Royal Society of London* **53** (0): 370–418. doi:10.1098/rstl.1763.0053.

[7] Holland, pp. 46–7.

[8] Richard Price (1991). *Price: Political Writings*. Cambridge University Press. p. xxiii. ISBN 978-0-521-40969-8. Retrieved 16 June 2013.

[9] Laplace refined Bayes' theorem over a period of decades:

- Laplace announced his independent discovery of Bayes' theorem in: Laplace (1774) "Mémoire sur la probabilité des causes par les événements," *Mémoires de l'Académie royale des Sciences de MI (Savants étrangers)*, **4**: 621–656. Reprinted in: Laplace, *Oeuvres complètes* (Paris, France: Gauthier-Villars et fils, 1841), vol. 8, pp. 27–65. Available on-line at: Gallica. Bayes' theorem appears on p. 29.
- Laplace presented a refinement of Bayes' theorem in: Laplace (read: 1783 / published: 1785) "Mémoire sur les approximations des formules qui sont fonctions de très grands nombres," *Mémoires de l'Académie royale des Sciences de Paris*, 423–467. Reprinted in: Laplace, *Oeuvres complètes* (Paris, France: Gauthier-Villars et fils, 1844), vol. 10, pp. 295–338. Available on-line at: Gallica. Bayes' theorem is stated on page 301.
- See also: Laplace, *Essai philosophique sur les probabilités* (Paris, France: Mme. Ve. Courcier [Madame veuve (i.e., widow) Courcier], 1814), page 10. English translation: Pierre Simon, Marquis de Laplace with F. W. Truscott and F. L. Emory, trans., *A Philosophical Essay on Probabilities* (New York, New York: John Wiley & Sons, 1902), page 15.

[10] Daston, Lorraine (1988). *Classical Probability in the Enlightenment*. Princeton Univ Press. p. 268. ISBN 0-691-08497-1.

[11] Stigler, Stephen M (1983). "Who Discovered Bayes' Theorem?". *The American Statistician* **37** (4): 290–296. doi:10.1080/00031305.1983.1048

[12] Edwards, A. W. F. (1986). "Is the Reference in Hartley (1749) to Bayesian Inference?". *The American Statistician* **40** (2): 109–110. doi:10.1080/00031305.1986.10475370.

[13] Hooper, Martyn (2013). "Richard Price, Bayes' theorem, and God". *Significance* **10** (1): 36–39. doi:10.1111/j.1740-9713.2013.00638.x.

[14] McGrayne, S. B. (2011). *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines & Emerged Triumphant from Two Centuries of Controversy*. Yale University Press. ISBN 978-0-300-18822-6.

1.10 Further reading

- Bruss, F. Thomas (2013), "250 years of 'An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S.' ", DOI 10.1365/s13291-013-0077-z, Jahresbericht der Deutschen Mathematiker-Vereinigung, Springer Verlag, Vol. 115, Issue 3-4 (2013), 129-133.
- Gelman, A, Carlin, JB, Stern, HS, and Rubin, DB (2003), "Bayesian Data Analysis", Second Edition, CRC Press.
- Grinstead, CM and Snell, JL (1997), "Introduction to Probability (2nd edition)", American Mathematical Society (free pdf available) .
- Hazewinkel, Michiel, ed. (2001), "Bayes formula", *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- McGrayne, SB (2011). *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines & Emerged Triumphant from Two Centuries of Controversy*. Yale University Press. ISBN 978-0-300-18822-6.
- Laplace, P (1774/1986), "Memoir on the Probability of the Causes of Events", *Statistical Science* 1(3):364–378.
- Lee, PM (2012), "Bayesian Statistics: An Introduction", Wiley.
- Rosenthal, JS (2005), "Struck by Lightning: the Curious World of Probabilities". Harper Collings.
- Stigler, SM (1986). "Laplace's 1774 Memoir on Inverse Probability". *Statistical Science* **1** (3): 359–363. doi:10.1214/ss/1177013620.
- Stone, JV (2013), download chapter 1 of "Bayes' Rule: A Tutorial Introduction to Bayesian Analysis", Sebtel Press, England.

1.11 External links

- Bayes' theorem at *Encyclopædia Britannica*
- The Theory That Would Not Die by Sharon Bertsch McGrayne New York Times Book Review by John Allen Paulos on 5 August 2011
- Visual explanation of Bayes using trees (video)
- Bayes' frequentist interpretation explained visually (video)
- Earliest Known Uses of Some of the Words of Mathematics (B). Contains origins of "Bayesian", "Bayes' Theorem", "Bayes Estimate/Risk/Solution", "Empirical Bayes", and "Bayes Factor".
- Weisstein, Eric W., "Bayes' Theorem", *MathWorld*.
- Bayes' theorem at PlanetMath.org.
- Bayes Theorem and the Folly of Prediction

- A tutorial on probability and Bayes' theorem devised for Oxford University psychology students
- An Intuitive Explanation of Bayes' Theorem by Eliezer S. Yudkowsky

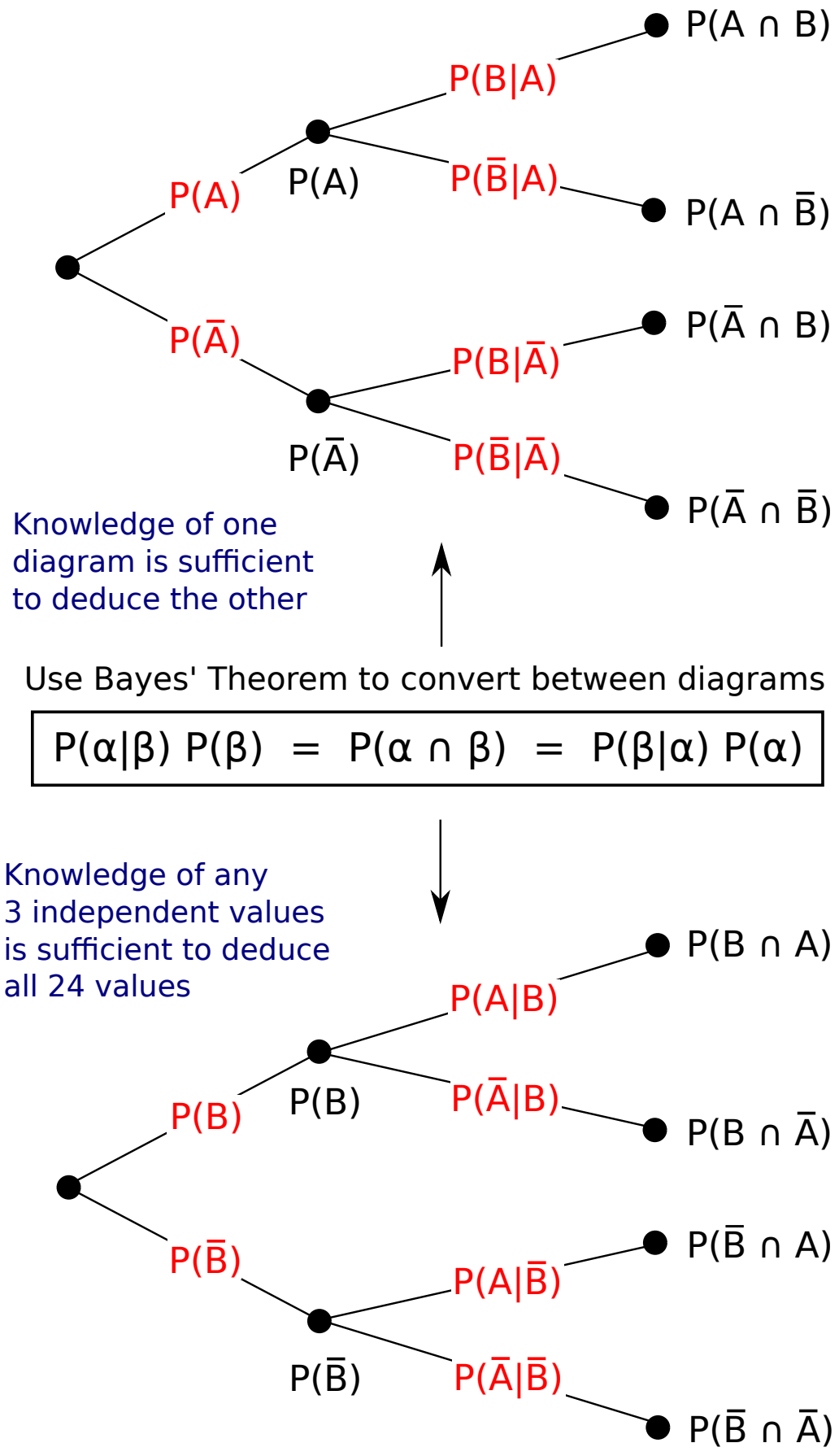
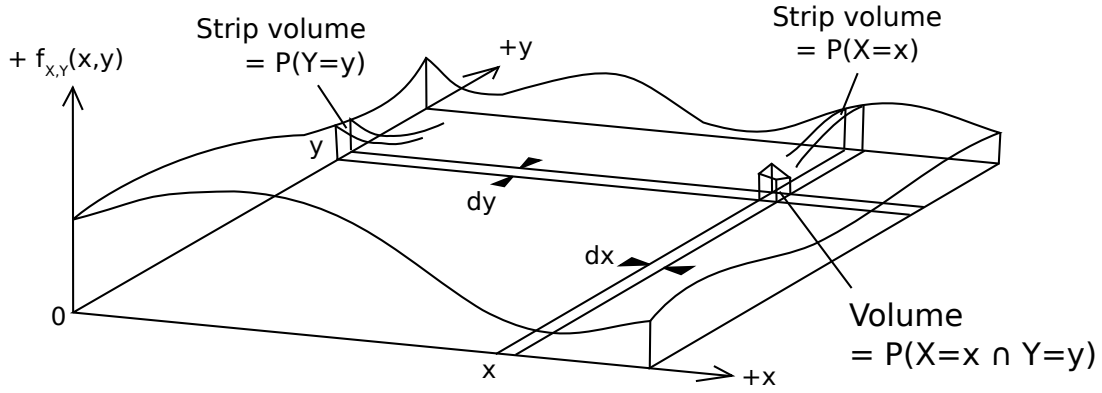


Illustration of frequentist interpretation with tree diagrams. Bayes' theorem connects conditional probabilities to their inverses.



$$P(Y=y|X=x) = \frac{P(X=x \cap Y=y)}{P(X=x)} \qquad P(X=x|Y=y) = \frac{P(X=x \cap Y=y)}{P(Y=y)}$$

Diagram illustrating the meaning of Bayes' theorem as applied to an event space generated by continuous random variables X *and* Y*. Note that there exists an instance of Bayes' theorem for each point in the domain. In practice, these instances might be parametrized by writing the specified probability densities as a function of* x *and* y*.*

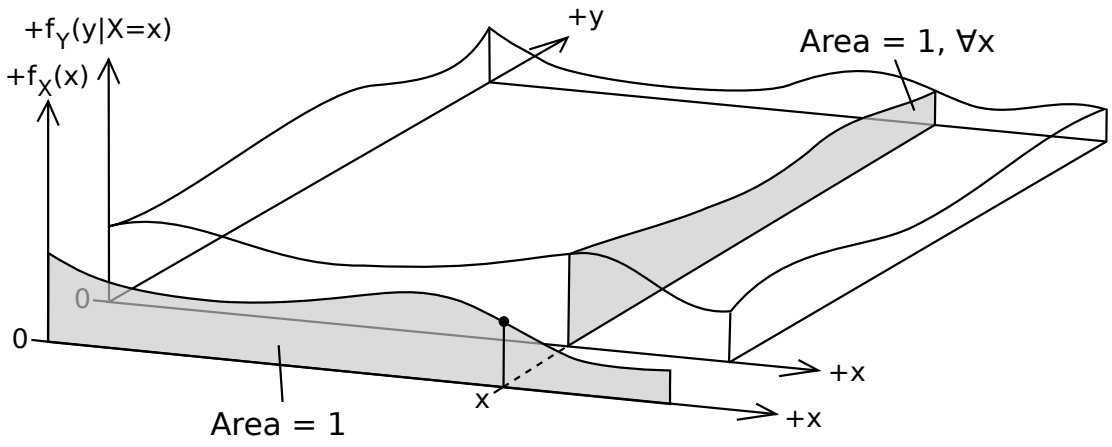
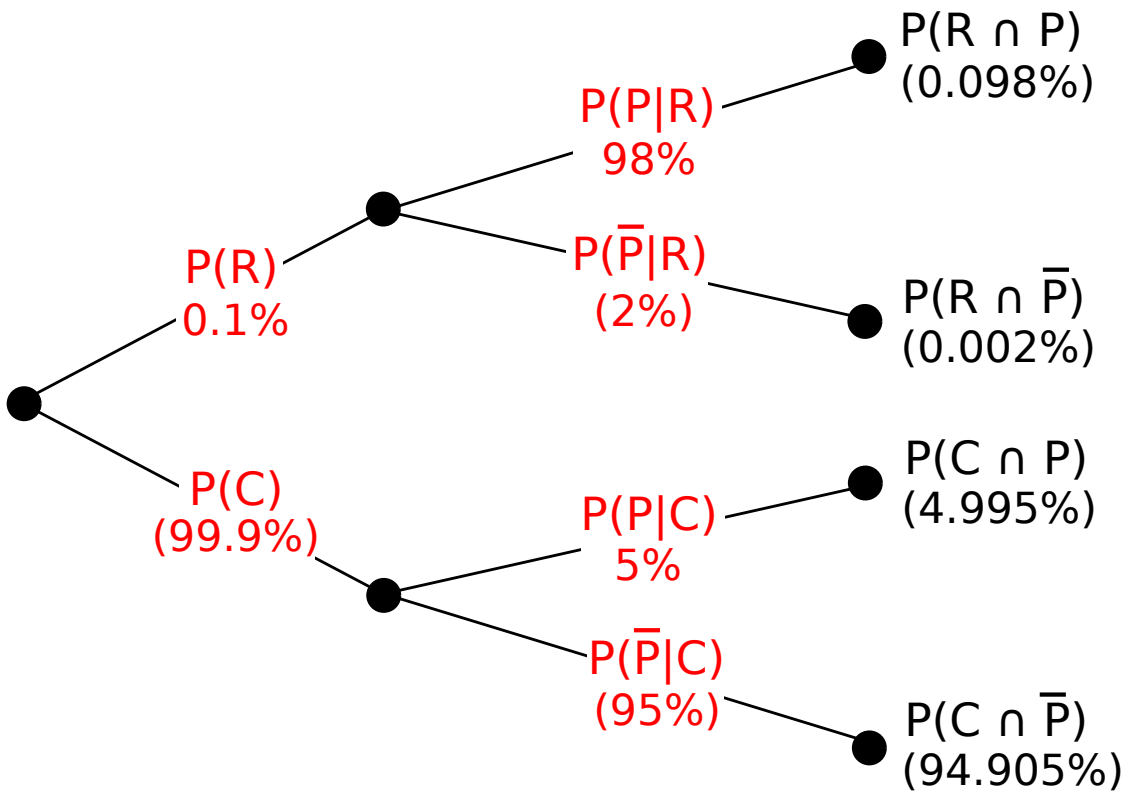
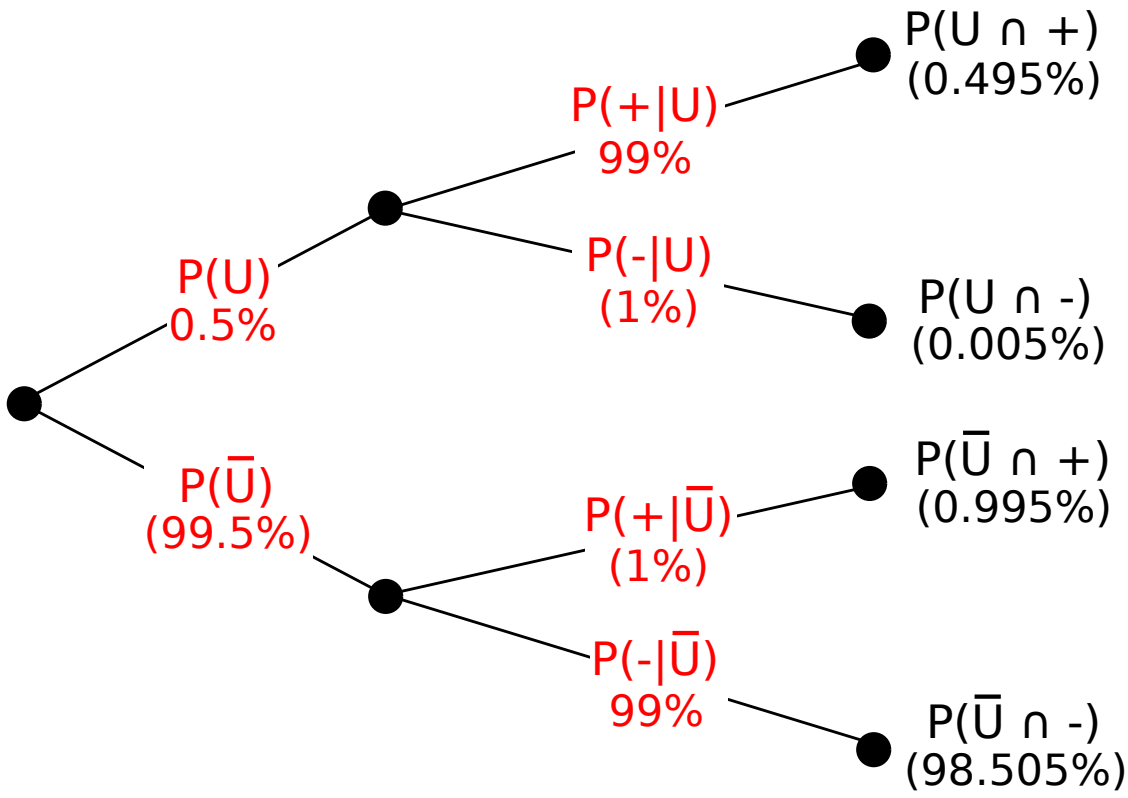


Diagram illustrating how an event space generated by continuous random variables X and Y is often conceptualized.



Tree diagram illustrating frequentist example. R, C, P and P bar are the events representing rare, common, pattern and no pattern. Percentages in parentheses are calculated. Note that three independent values are given, so it is possible to calculate the inverse tree (see figure above).



Tree diagram illustrating drug testing example. U, U bar, "+" and "−" are the events representing user, non-user, positive result and negative result. Percentages in parentheses are calculated.

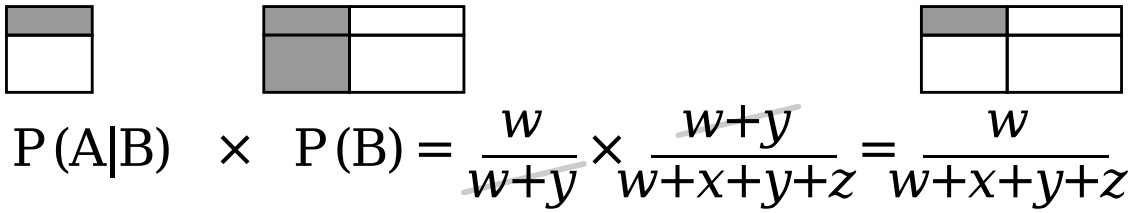
Chapter 2

Bayesian inference

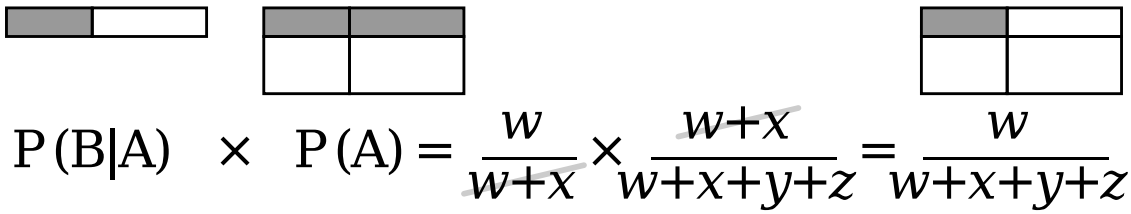
Bayesian inference is a method of statistical inference in which Bayes' rule is used to update the probability for a hypothesis as evidence is acquired. Bayesian inference is an important technique in statistics, and especially in mathematical statistics. Bayesian updating is particularly important in the dynamic analysis of a sequence of data. Bayesian inference has found application in a wide range of activities, including science, engineering, philosophy, medicine, and law. In the philosophy of decision theory, Bayesian inference is closely related to subjective probability, often called "Bayesian probability". Bayesian probability provides a rational method for updating beliefs.

2.1 Introduction to Bayes' rule

Relative size	Case B	Case $\bar{B}$	Total
Condition A	w	x	$w+x$
Condition $\bar{A}$	y	z	$y+z$
Total	$w+y$	$x+z$	$w+x+y+z$



$$P(A|B) \times P(B) = \frac{w}{w+y} \times \frac{w+y}{w+x+y+z} = \frac{w}{w+x+y+z}$$



$$P(B|A) \times P(A) = \frac{w}{w+x} \times \frac{w+x}{w+x+y+z} = \frac{w}{w+x+y+z}$$

$\bar{A}$) P($\bar{A}$)/P(B) etc.

Main article: Bayes' rule

See also: Bayesian probability

2.1.1 Formal

Bayesian inference derives the posterior probability as a consequence of two antecedents, a prior probability and a "likelihood function" derived from a statistical model for the observed data. Bayesian inference computes the posterior probability according to Bayes' theorem:

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

where

- $\mid$ denotes a conditional probability; more specifically, it means *given*.
- H stands for any *hypothesis* whose probability may be affected by data (called *evidence* below). Often there are competing hypotheses, from which one chooses the most probable.
- the *evidence* E corresponds to new data that were not used in computing the prior probability.
- $P(H)$, the *prior probability*, is the probability of H *before* E is observed. This indicates one's previous estimate of the probability that a hypothesis is true, before gaining the current evidence.
- $P(H \mid E)$, the *posterior probability*, is the probability of H *given* E , i.e., *after* E is observed. This tells us what we want to know: the probability of a hypothesis *given* the observed evidence.
- $P(E \mid H)$ is the probability of observing E *given* H . As a function of E with H fixed, this is the *likelihood*. The likelihood function should **not** be confused with $P(H \mid E)$ as a function of H rather than of E . It indicates the compatibility of the evidence with the given hypothesis.
- $P(E)$ is sometimes termed the marginal likelihood or "model evidence". This factor is the same for all possible hypotheses being considered. (This can be seen by the fact that the hypothesis H does not appear anywhere in the symbol, unlike for all the other factors.) This means that this factor does not enter into determining the relative probabilities of different hypotheses.

Note that, for different values of H , only the factors $P(H)$ and $P(E \mid H)$ affect the value of $P(H \mid E)$. As both of these factors appear in the numerator, the posterior probability is proportional to both. In words:

- (more precisely) *The posterior probability of a hypothesis is determined by a combination of the inherent likeliness of a hypothesis (the prior) and the compatibility of the observed evidence with the hypothesis (the likelihood).*
- (more concisely) *Posterior is proportional to likelihood times prior.*

Note that Bayes' rule can also be written as follows:

$$P(H \mid E) = \frac{P(E \mid H)}{P(E)} \cdot P(H)$$

where the factor $\frac{P(E|H)}{P(E)}$ represents the impact of E on the probability of H .

2.1.2 Informal

If the evidence does not match up with a hypothesis, one should reject the hypothesis. But if a hypothesis is extremely unlikely *a priori*, one should also reject it, even if the evidence does appear to match up.

For example, imagine that I have various hypotheses about the nature of a newborn baby of a friend, including:

- H_1 : the baby is a brown-haired boy.
- H_2 : the baby is a blond-haired girl.
- H_3 : the baby is a dog.

Then consider two scenarios:

1. I'm presented with evidence in the form of a picture of a blond-haired baby girl. I find this evidence supports H_2 and opposes H_1 and H_3 .
2. I'm presented with evidence in the form of a picture of a baby dog. Although this evidence, treated in isolation, supports H_3 , my prior belief in this hypothesis (that a human can give birth to a dog) is extremely small, so the posterior probability is nevertheless small.

The critical point about Bayesian inference, then, is that it provides a principled way of combining new evidence with prior beliefs, through the application of Bayes' rule. (Contrast this with frequentist inference, which relies only on the evidence as a whole, with no reference to prior beliefs.) Furthermore, Bayes' rule can be applied iteratively: after observing some evidence, the resulting posterior probability can then be treated as a prior probability, and a new posterior probability computed from new evidence. This allows for Bayesian principles to be applied to various kinds of evidence, whether viewed all at once or over time. This procedure is termed "Bayesian updating".

2.1.3 Bayesian updating

Bayesian updating is widely used and computationally convenient. However, it is not the only updating rule that might be considered "rational".

Ian Hacking noted that traditional "Dutch book" arguments did not specify Bayesian updating: they left open the possibility that non-Bayesian updating rules could avoid Dutch books. Hacking wrote[1] "And neither the Dutch book argument, nor any other in the personalist arsenal of proofs of the probability axioms, entails the dynamic assumption. Not one entails Bayesianism. So the personalist requires the dynamic assumption to be Bayesian. It is true that in consistency a personalist could abandon the Bayesian model of learning from experience. Salt could lose its savour."

Indeed, there are non-Bayesian updating rules that also avoid Dutch books (as discussed in the literature on "probability kinematics" following the publication of Richard C. Jeffrey's rule, which applies Bayes' rule to the case where the evidence itself is assigned a probability.[2] The additional hypotheses needed to uniquely require Bayesian updating have been deemed to be substantial, complicated, and unsatisfactory.[3]

2.2 Formal description of Bayesian inference

2.2.1 Definitions

- x , a data point in general. This may in fact be a vector of values.
- θ , the parameter of the data point's distribution, i.e., $x \sim p(x \mid \theta)$. This may in fact be a vector of parameters.
- α , the hyperparameter of the parameter, i.e., $\theta \sim p(\theta \mid \alpha)$. This may in fact be a vector of hyperparameters.
- $\mathbf{X}$, a set of n observed data points, i.e., $x_1, \ldots, x_n$.
- $\tilde{x}$, a new data point whose distribution is to be predicted.

2.2.2 Bayesian inference

- The prior distribution is the distribution of the parameter(s) before any data is observed, i.e. $p(\theta \mid \alpha)$.
- The prior distribution might not be easily determined. In this case, we can use the Jeffreys prior to obtain the posterior distribution before updating them with newer observations.

- The sampling distribution is the distribution of the observed data conditional on its parameters, i.e. $p(\mathbf{X} \mid \theta)$. This is also termed the likelihood, especially when viewed as a function of the parameter(s), sometimes written $\mathrm{L}(\theta \mid \mathbf{X}) = p(\mathbf{X} \mid \theta)$.

- The marginal likelihood (sometimes also termed the *evidence*) is the distribution of the observed data marginalized over the parameter(s), i.e. $p(\mathbf{X} \mid \alpha) = \int_\theta p(\mathbf{X} \mid \theta) p(\theta \mid \alpha) \, \mathrm{d}\theta$.

- The posterior distribution is the distribution of the parameter(s) after taking into account the observed data. This is determined by Bayes' rule, which forms the heart of Bayesian inference:

$$p(\theta \mid \mathbf{X}, \alpha) = \frac{p(\mathbf{X} \mid \theta) p(\theta \mid \alpha)}{p(\mathbf{X} \mid \alpha)} \propto p(\mathbf{X} \mid \theta) p(\theta \mid \alpha)$$

Note that this is expressed in words as "posterior is proportional to likelihood times prior", or sometimes as "posterior = likelihood times prior, over evidence".

2.2.3 Bayesian prediction

- The posterior predictive distribution is the distribution of a new data point, marginalized over the posterior:

$$p(\tilde{x} \mid \mathbf{X}, \alpha) = \int_\theta p(\tilde{x} \mid \theta) p(\theta \mid \mathbf{X}, \alpha) \, \mathrm{d}\theta$$

- The prior predictive distribution is the distribution of a new data point, marginalized over the prior:

$$p(\tilde{x} \mid \alpha) = \int_\theta p(\tilde{x} \mid \theta) p(\theta \mid \alpha) \, \mathrm{d}\theta$$

Bayesian theory calls for the use of the posterior predictive distribution to do predictive inference, i.e., to predict the distribution of a new, unobserved data point. That is, instead of a fixed point as a prediction, a distribution over possible points is returned. Only this way is the entire posterior distribution of the parameter(s) used. By comparison, prediction in frequentist statistics often involves finding an optimum point estimate of the parameter(s)—e.g., by maximum likelihood or maximum a posteriori estimation (MAP)—and then plugging this estimate into the formula for the distribution of a data point. This has the disadvantage that it does not account for any uncertainty in the value of the parameter, and hence will underestimate the variance of the predictive distribution.

(In some instances, frequentist statistics can work around this problem. For example, confidence intervals and prediction intervals in frequentist statistics when constructed from a normal distribution with unknown mean and variance are constructed using a Student's t-distribution. This correctly estimates the variance, due to the fact that (1) the average of normally distributed random variables is also normally distributed; (2) the predictive distribution of a normally distributed data point with unknown mean and variance, using conjugate or uninformative priors, has a student's t-distribution. In Bayesian statistics, however, the posterior predictive distribution can always be determined exactly—or at least, to an arbitrary level of precision, when numerical methods are used.)

Note that both types of predictive distributions have the form of a compound probability distribution (as does the marginal likelihood). In fact, if the prior distribution is a conjugate prior, and hence the prior and posterior distributions come from the same family, it can easily be seen that both prior and posterior predictive distributions also come from the same family of compound distributions. The only difference is that the posterior predictive distribution uses the updated values of the hyperparameters (applying the Bayesian update rules given in the conjugate prior article), while the prior predictive distribution uses the values of the hyperparameters that appear in the prior distribution.

2.3 Inference over exclusive and exhaustive possibilities

If evidence is simultaneously used to update belief over a set of exclusive and exhaustive propositions, Bayesian inference may be thought of as acting on this belief distribution as a whole.

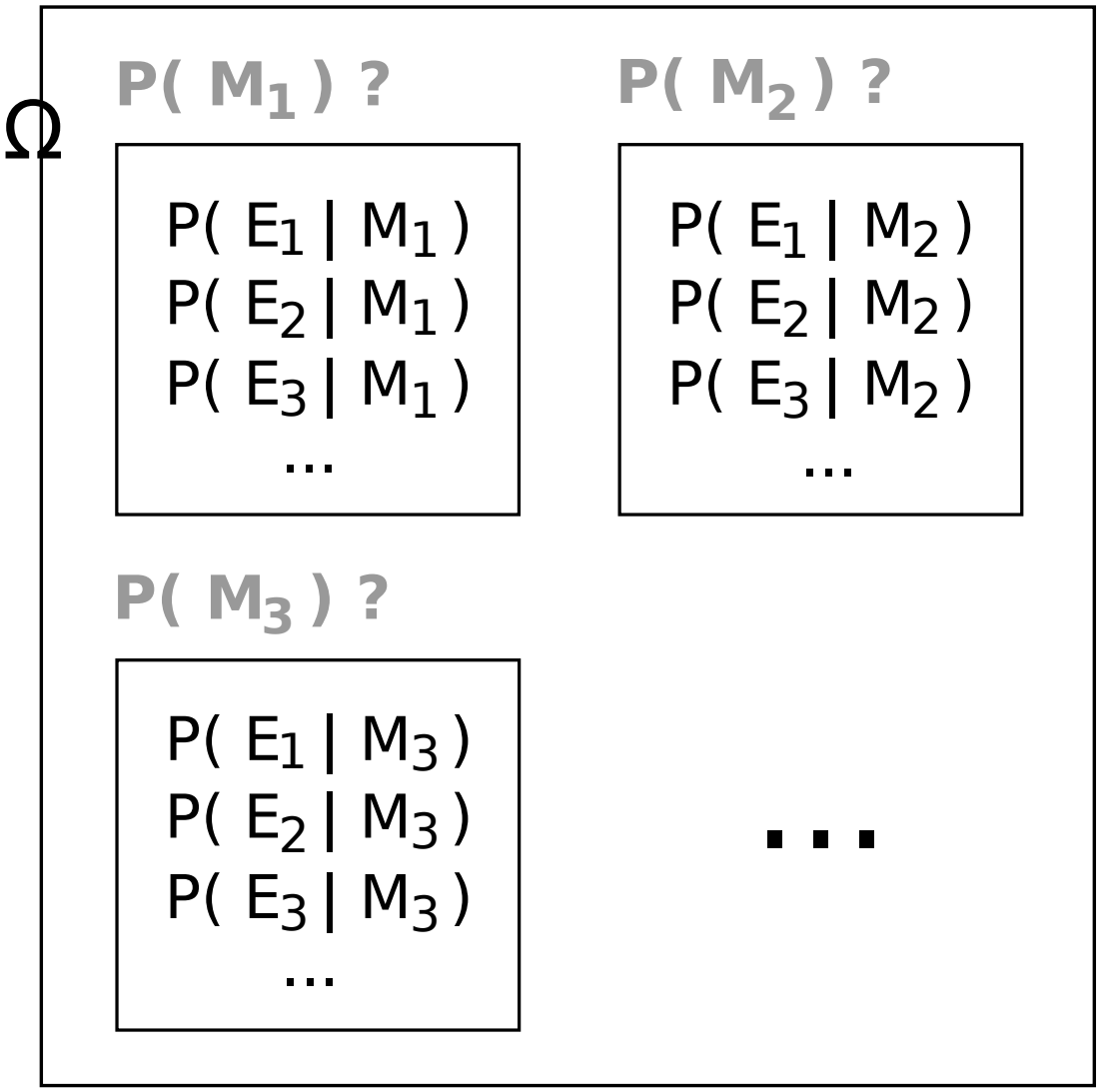


Diagram illustrating event space Ω in general formulation of Bayesian inference. Although this diagram shows discrete models and events, the continuous case may be visualized similarly using probability densities.

2.3.1 General formulation

Suppose a process is generating independent and identically distributed events E_n , but the probability distribution is unknown. Let the event space Ω represent the current state of belief for this process. Each model is represented by event M_m . The conditional probabilities $P(E_n \mid M_m)$ are specified to define the models. $P(M_m)$ is the degree of belief in M_m . Before the first inference step, $\{P(M_m)\}$ is a set of *initial prior probabilities*. These must sum to 1, but are otherwise arbitrary.

Suppose that the process is observed to generate $E \in \{E_n\}$. For each $M \in \{M_m\}$, the prior $P(M)$ is updated to the posterior $P(M \mid E)$. From Bayes' theorem:[4]

$$P(M \mid E) = \frac{P(E \mid M)}{\sum_m P(E \mid M_m) P(M_m)} \cdot P(M)$$

Upon observation of further evidence, this procedure may be repeated.

2.3.2 Multiple observations

For a set of independent and identically distributed observations $\mathbf{E} = \{e_1, \ldots, e_n\}$, it may be shown that repeated application of the above is equivalent to

$$P(M \mid \mathbf{E}) = \frac{P(\mathbf{E} \mid M)}{\sum_m P(\mathbf{E} \mid M_m)P(M_m)} \cdot P(M)$$

Where

$$P(\mathbf{E} \mid M) = \prod_k P(e_k \mid M).$$

This may be used to optimize practical calculations.

2.3.3 Parametric formulation

By parameterizing the space of models, the belief in all models may be updated in a single step. The distribution of belief over the model space may then be thought of as a distribution of belief over the parameter space. The distributions in this section are expressed as continuous, represented by probability densities, as this is the usual situation. The technique is however equally applicable to discrete distributions.

Let the vector θ span the parameter space. Let the initial prior distribution over θ be $p(\theta \mid \alpha)$, where α is a set of parameters to the prior itself, or *hyperparameters*. Let $\mathbf{E} = \{e_1, \ldots, e_n\}$ be a set of independent and identically distributed event observations, where all e_i are distributed as $p(e \mid \theta)$ for some θ . Bayes' theorem is applied to find the posterior distribution over θ :

$$\begin{aligned} p(\theta \mid \mathbf{E}, \alpha) &= \frac{p(\mathbf{E} \mid \theta, \alpha)}{p(\mathbf{E} \mid \alpha)} \cdot p(\theta \mid \alpha) \\ &= \frac{p(\mathbf{E} \mid \theta, \alpha)}{\int_\theta p(\mathbf{E}|\theta, \alpha)p(\theta \mid \alpha)\, d\theta} \cdot p(\theta \mid \alpha) \end{aligned}$$

Where

$$p(\mathbf{E} \mid \theta, \alpha) = \prod_k p(e_k \mid \theta)$$

2.4 Mathematical properties

2.4.1 Interpretation of factor

$\frac{P(E|M)}{P(E)} > 1 \Rightarrow P(E \mid M) > P(E)$. That is, if the model were true, the evidence would be more likely than is predicted by the current state of belief. The reverse applies for a decrease in belief. If the belief does not change, $\frac{P(E|M)}{P(E)} = 1 \Rightarrow P(E \mid M) = P(E)$. That is, the evidence is independent of the model. If the model were true, the evidence would be exactly as likely as predicted by the current state of belief.

2.4.2 Cromwell's rule

Main article: Cromwell's rule

If $P(M) = 0$ then $P(M \mid E) = 0$. If $P(M) = 1$, then $P(M|E) = 1$. This can be interpreted to mean that hard convictions are insensitive to counter-evidence.

The former follows directly from Bayes' theorem. The latter can be derived by applying the first rule to the event "not M " in place of " M ", yielding "if $1 - P(M) = 0$, then $1 - P(M \mid E) = 0$ ", from which the result immediately follows.

2.4.3 Asymptotic behaviour of posterior

Consider the behaviour of a belief distribution as it is updated a large number of times with independent and identically distributed trials. For sufficiently nice prior probabilities, the Bernstein-von Mises theorem gives that in the limit of infinite trials, the posterior converges to a Gaussian distribution independent of the initial prior under some conditions firstly outlined and rigorously proven by Joseph L. Doob in 1948, namely if the random variable in consideration has a finite probability space. The more general results were obtained later by the statistician David A. Freedman who published in two seminal research papers in 1963 and 1965 when and under what circumstances the asymptotic behaviour of posterior is guaranteed. His 1963 paper treats, like Doob (1949), the finite case and comes to a satisfactory conclusion. However, if the random variable has an infinite but countable probability space (i.e., corresponding to a die with infinite many faces) the 1965 paper demonstrates that for a dense subset of priors the Bernstein-von Mises theorem is not applicable. In this case there is almost surely no asymptotic convergence. Later in the 1980s and 1990s Freedman and Persi Diaconis continued to work on the case of infinite countable probability spaces.[5] To summarise, there may be insufficient trials to suppress the effects of the initial choice, and especially for large (but finite) systems the convergence might be very slow.

2.4.4 Conjugate priors

Main article: Conjugate prior

In parameterized form, the prior distribution is often assumed to come from a family of distributions called conjugate priors. The usefulness of a conjugate prior is that the corresponding posterior distribution will be in the same family, and the calculation may be expressed in closed form.

2.4.5 Estimates of parameters and predictions

It is often desired to use a posterior distribution to estimate a parameter or variable. Several methods of Bayesian estimation select measurements of central tendency from the posterior distribution.

For one-dimensional problems, a unique median exists for practical continuous problems. The posterior median is attractive as a robust estimator.[6]

If there exists a finite mean for the posterior distribution, then the posterior mean is a method of estimation.

$$\tilde{\theta} = \mathrm{E}[\theta] = \int_{\theta} \theta \, p(\theta \mid \mathbf{X}, \alpha) \, d\theta$$

Taking a value with the greatest probability defines maximum *a posteriori* (MAP) estimates:

$$\{\theta_{\mathrm{MAP}}\} \subset \arg\max_{\theta} p(\theta \mid \mathbf{X}, \alpha).$$

There are examples where no maximum is attained, in which case the set of MAP estimates is empty.

There are other methods of estimation that minimize the posterior *risk* (expected-posterior loss) with respect to a loss function, and these are of interest to statistical decision theory using the sampling distribution ("frequentist statistics").

The posterior predictive distribution of a new observation $\tilde{x}$ (that is independent of previous observations) is determined by

$$p(\tilde{x}|\mathbf{X}, \alpha) = \int_{\theta} p(\tilde{x}, \theta \mid \mathbf{X}, \alpha) \, d\theta = \int_{\theta} p(\tilde{x} \mid \theta) p(\theta \mid \mathbf{X}, \alpha) \, d\theta.$$

2.5 Examples

2.5.1 Probability of a hypothesis

Suppose there are two full bowls of cookies. Bowl #1 has 10 chocolate chip and 30 plain cookies, while bowl #2 has 20 of each. Our friend Fred picks a bowl at random, and then picks a cookie at random. We may assume there is no reason to believe Fred treats one bowl differently from another, likewise for the cookies. The cookie turns out to be a plain one. How probable is it that Fred picked it out of bowl #1?

Intuitively, it seems clear that the answer should be more than a half, since there are more plain cookies in bowl #1. The precise answer is given by Bayes' theorem. Let H_1 correspond to bowl #1, and H_2 to bowl #2. It is given that the bowls are identical from Fred's point of view, thus $P(H_1) = P(H_2)$, and the two must add up to 1, so both are equal to 0.5. The event E is the observation of a plain cookie. From the contents of the bowls, we know that $P(E \mid H_1) = 30/40 = 0.75$ and $P(E \mid H_2) = 20/40 = 0.5$. Bayes' formula then yields

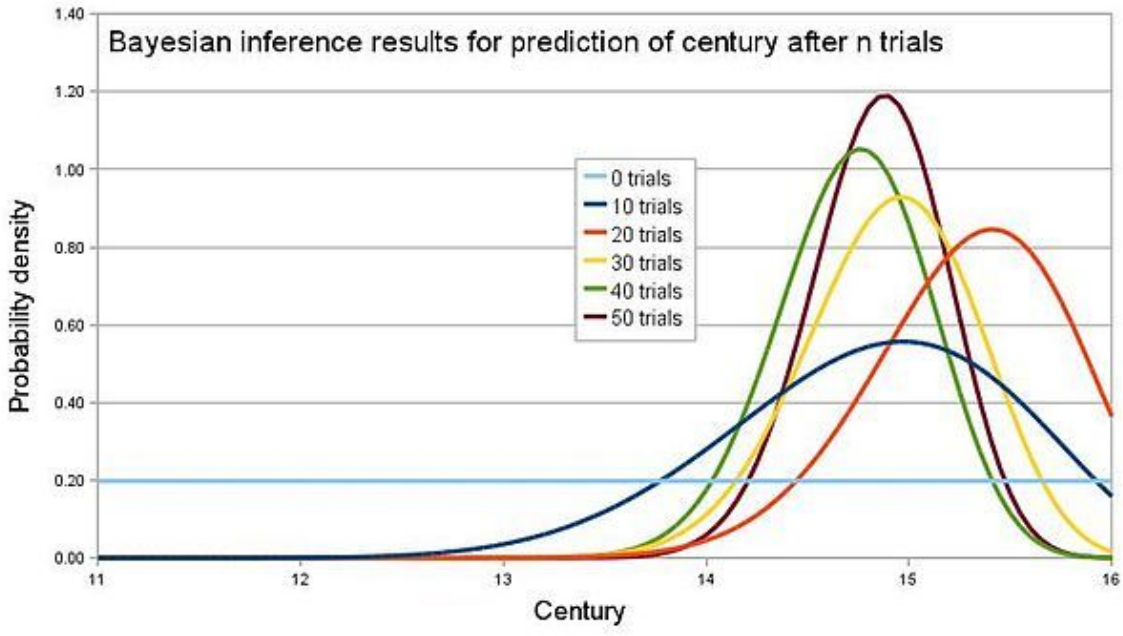
$$P(H_1 \mid E) = \frac{P(E \mid H_1)\,P(H_1)}{P(E \mid H_1)\,P(H_1) \;+\; P(E \mid H_2)\,P(H_2)}$$

$$= \frac{0.75 \times 0.5}{0.75 \times 0.5 + 0.5 \times 0.5}$$

$$= 0.6$$

Before we observed the cookie, the probability we assigned for Fred having chosen bowl #1 was the prior probability, $P(H_1)$, which was 0.5. After observing the cookie, we must revise the probability to $P(H_1 \mid E)$, which is 0.6.

2.5.2 Making a prediction



Example results for archaeology example. This simulation was generated using c=15.2.

An archaeologist is working at a site thought to be from the medieval period, between the 11th century to the 16th century. However, it is uncertain exactly when in this period the site was inhabited. Fragments of pottery are found, some of which are glazed and some of which are decorated. It is expected that if the site were inhabited during the early medieval period, then 1% of the pottery would be glazed and 50% of its area decorated, whereas if it had been

inhabited in the late medieval period then 81% would be glazed and 5% of its area decorated. How confident can the archaeologist be in the date of inhabitation as fragments are unearthed?

The degree of belief in the continuous variable C (century) is to be calculated, with the discrete set of events $\{GD, G\bar{D}, \bar{G}D, \bar{G}\bar{D}\}$ as evidence. Assuming linear variation of glaze and decoration with time, and that these variables are independent,

$$P(E = GD \mid C = c) = (0.01 + 0.16(c - 11))(0.5 - 0.09(c - 11))$$

$$P(E = G\bar{D} \mid C = c) = (0.01 + 0.16(c - 11))(0.5 + 0.09(c - 11))$$

$$P(E = \bar{G}D \mid C = c) = (0.99 - 0.16(c - 11))(0.5 - 0.09(c - 11))$$

$$P(E = \bar{G}\bar{D} \mid C = c) = (0.99 - 0.16(c - 11))(0.5 + 0.09(c - 11))$$

Assume a uniform prior of $f_C(c) = 0.2$, and that trials are independent and identically distributed. When a new fragment of type e is discovered, Bayes' theorem is applied to update the degree of belief for each c :

$$f_C(c \mid E = e) = \frac{P(E=e|C=c)}{P(E=e)} f_C(c) = \frac{P(E=e|C=c)}{\int_{11}^{16} P(E=e|C=c) f_C(c) dc} f_C(c)$$

A computer simulation of the changing belief as 50 fragments are unearthed is shown on the graph. In the simulation, the site was inhabited around 1420, or $c = 15.2$. By calculating the area under the relevant portion of the graph for 50 trials, the archaeologist can say that there is practically no chance the site was inhabited in the 11th and 12th centuries, about 1% chance that it was inhabited during the 13th century, 63% chance during the 14th century and 36% during the 15th century. Note that the Bernstein-von Mises theorem asserts here the asymptotic convergence to the "true" distribution because the probability space corresponding to the discrete set of events $\{GD, G\bar{D}, \bar{G}D, \bar{G}\bar{D}\}$ is finite (see above section on asymptotic behaviour of the posterior).

2.6 In frequentist statistics and decision theory

A decision-theoretic justification of the use of Bayesian inference was given by Abraham Wald, who proved that every unique Bayesian procedure is admissible. Conversely, every admissible statistical procedure is either a Bayesian procedure or a limit of Bayesian procedures.[7]

Wald characterized admissible procedures as Bayesian procedures (and limits of Bayesian procedures), making the Bayesian formalism a central technique in such areas of frequentist inference as parameter estimation, hypothesis testing, and computing confidence intervals.[8] For example:

- "Under some conditions, all admissible procedures are either Bayes procedures or limits of Bayes procedures (in various senses). These remarkable results, at least in their original form, are due essentially to Wald. They are useful because the property of being Bayes is easier to analyze than admissibility."[7]

- "In decision theory, a quite general method for proving admissibility consists in exhibiting a procedure as a unique Bayes solution."[9]

- "In the first chapters of this work, prior distributions with finite support and the corresponding Bayes procedures were used to establish some of the main theorems relating to the comparison of experiments. Bayes procedures with respect to more general prior distributions have played a very important role in the development of statistics, including its asymptotic theory." "There are many problems where a glance at posterior distributions, for suitable priors, yields immediately interesting information. Also, this technique can hardly be avoided in sequential analysis."[10]

- "A useful fact is that any Bayes decision rule obtained by taking a proper prior over the whole parameter space must be admissible"[11]

- "An important area of investigation in the development of admissibility ideas has been that of conventional sampling-theory procedures, and many interesting results have been obtained."[12]

2.6.1 Model selection

See Bayesian model selection

2.7 Applications

2.7.1 Computer applications

Bayesian inference has applications in artificial intelligence and expert systems. Bayesian inference techniques have been a fundamental part of computerized pattern recognition techniques since the late 1950s. There is also an ever growing connection between Bayesian methods and simulation-based Monte Carlo techniques since complex models cannot be processed in closed form by a Bayesian analysis, while a graphical model structure *may* allow for efficient simulation algorithms like the Gibbs sampling and other Metropolis–Hastings algorithm schemes.[13] Recently Bayesian inference has gained popularity amongst the phylogenetics community for these reasons; a number of applications allow many demographic and evolutionary parameters to be estimated simultaneously.

As applied to statistical classification, Bayesian inference has been used in recent years to develop algorithms for identifying e-mail spam. Applications which make use of Bayesian inference for spam filtering include CRM114, DSPAM, Bogofilter, SpamAssassin, SpamBayes, Mozilla, XEAMS, and others. Spam classification is treated in more detail in the article on the naive Bayes classifier.

Solomonoff's Inductive inference is the theory of prediction based on observations; for example, predicting the next symbol based upon a given series of symbols. The only assumption is that the environment follows some unknown but computable probability distribution. It is a formal inductive framework that combines two well-studied principles of inductive inference: Bayesian statistics and Occam's Razor.[14] Solomonoff's universal prior probability of any prefix p of a computable sequence x is the sum of the probabilities of all programs (for a universal computer) that compute something starting with p. Given some p and any computable but unknown probability distribution from which x is sampled, the universal prior and Bayes' theorem can be used to predict the yet unseen parts of x in optimal fashion.[15][16]

2.7.2 In the courtroom

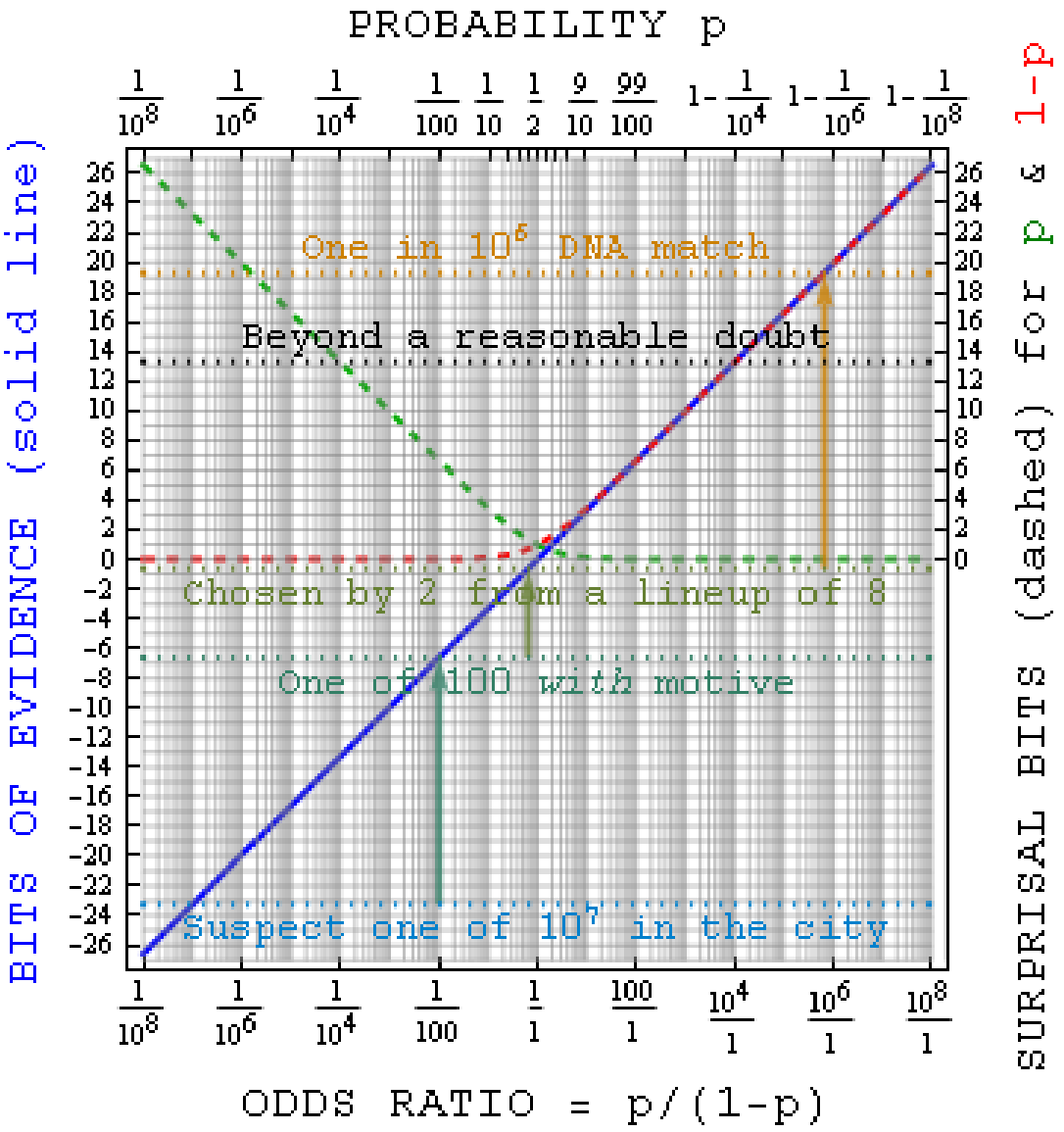
Bayesian inference can be used by jurors to coherently accumulate the evidence for and against a defendant, and to see whether, in totality, it meets their personal threshold for 'beyond a reasonable doubt'.[17][18][19] Bayes' theorem is applied successively to all evidence presented, with the posterior from one stage becoming the prior for the next. The benefit of a Bayesian approach is that it gives the juror an unbiased, rational mechanism for combining evidence. It may be appropriate to explain Bayes' theorem to jurors in odds form, as betting odds are more widely understood than probabilities. Alternatively, a logarithmic approach, replacing multiplication with addition, might be easier for a jury to handle.

If the existence of the crime is not in doubt, only the identity of the culprit, it has been suggested that the prior should be uniform over the qualifying population.[20] For example, if 1,000 people could have committed the crime, the prior probability of guilt would be 1/1000.

The use of Bayes' theorem by jurors is controversial. In the United Kingdom, a defence expert witness explained Bayes' theorem to the jury in *R v Adams*. The jury convicted, but the case went to appeal on the basis that no means of accumulating evidence had been provided for jurors who did not wish to use Bayes' theorem. The Court of Appeal upheld the conviction, but it also gave the opinion that "To introduce Bayes' Theorem, or any similar method, into a criminal trial plunges the jury into inappropriate and unnecessary realms of theory and complexity, deflecting them from their proper task."

Gardner-Medwin[21] argues that the criterion on which a verdict in a criminal trial should be based is *not* the probability of guilt, but rather the *probability of the evidence, given that the defendant is innocent* (akin to a frequentist p-value). He argues that if the posterior probability of guilt is to be computed by Bayes' theorem, the prior probability of guilt must be known. This will depend on the incidence of the crime, which is an unusual piece of evidence to consider in a criminal trial. Consider the following three propositions:

A The known facts and testimony could have arisen if the defendant is guilty



Adding up evidence.

B The known facts and testimony could have arisen if the defendant is innocent

C The defendant is guilty.

Gardner-Medwin argues that the jury should believe both A and not-B in order to convict. A and not-B implies the truth of C, but the reverse is not true. It is possible that B and C are both true, but in this case he argues that a jury should acquit, even though they know that they will be letting some guilty people go free. See also Lindley's paradox.

2.7.3 Bayesian epistemology

Bayesian epistemology is a movement that advocates for Bayesian inference as a means of justifying the rules of inductive logic.

Karl Popper and David Miller have rejected the alleged rationality of Bayesianism, i.e. using Bayes rule to make epistemological inferences:[22] It is prone to the same vicious circle as any other justificationist epistemology, because it presupposes what it attempts to justify. According to this view, a rational interpretation of Bayesian inference would

see it merely as a probabilistic version of falsification, rejecting the belief, commonly held by Bayesians, that high likelihood achieved by a series of Bayesian updates would prove the hypothesis beyond any reasonable doubt, or even with likelihood greater than 0.

2.7.4 Other

- The scientific method is sometimes interpreted as an application of Bayesian inference. In this view, Bayes' rule guides (or should guide) the updating of probabilities about hypotheses conditional on new observations or experiments.[23]
- Bayesian search theory is used to search for lost objects.
- Bayesian inference in phylogeny
- Bayesian tool for methylation analysis

2.8 Bayes and Bayesian inference

The problem considered by Bayes in Proposition 9 of his essay, "An Essay towards solving a Problem in the Doctrine of Chances", is the posterior distribution for the parameter a (the success rate) of the binomial distribution.

2.9 History

Main article: History of statistics § Bayesian statistics

The term *Bayesian* refers to Thomas Bayes (1702–1761), who proved a special case of what is now called Bayes' theorem. However, it was Pierre-Simon Laplace (1749–1827) who introduced a general version of the theorem and used it to approach problems in celestial mechanics, medical statistics, reliability, and jurisprudence.[24] Early Bayesian inference, which used uniform priors following Laplace's principle of insufficient reason, was called "inverse probability" (because it infers backwards from observations to parameters, or from effects to causes[25]). After the 1920s, "inverse probability" was largely supplanted by a collection of methods that came to be called frequentist statistics.[25]

In the 20th century, the ideas of Laplace were further developed in two different directions, giving rise to *objective* and *subjective* currents in Bayesian practice. In the objective or "non-informative" current, the statistical analysis depends on only the model assumed, the data analyzed,[26] and the method assigning the prior, which differs from one objective Bayesian to another objective Bayesian. In the subjective or "informative" current, the specification of the prior depends on the belief (that is, propositions on which the analysis is prepared to act), which can summarize information from experts, previous studies, etc.

In the 1980s, there was a dramatic growth in research and applications of Bayesian methods, mostly attributed to the discovery of Markov chain Monte Carlo methods, which removed many of the computational problems, and an increasing interest in nonstandard, complex applications.[27] Despite growth of Bayesian research, most undergraduate teaching is still based on frequentist statistics.[28] Nonetheless, Bayesian methods are widely accepted and used, such as for example in the field of machine learning.[29]

2.10 See also

- Bayes' theorem
- Bayesian hierarchical modeling
- Bayesian Analysis, the journal of the ISBA
- Inductive probability

- International Society for Bayesian Analysis (ISBA)
- Jeffreys prior

2.11 Notes

[1] Hacking (1967, Section 3, p. 316), Hacking (1988, p. 124)

[2] "Bayes' Theorem (Stanford Encyclopedia of Philosophy)". Plato.stanford.edu. Retrieved 2014-01-05.

[3] van Fraassen, B. (1989) *Laws and Symmetry*, Oxford University Press. ISBN 0-19-824860-1

[4] Gelman, Andrew; Carlin, John B.; Stern, Hal S.; Dunson, David B.;Vehtari, Aki; Rubin, Donald B. (2013). *Bayesian Data Analysis*, Third Edition. Chapman and Hall/CRC. ISBN 978-1-4398-4095-5.

[5] Larry Wasserman et alia, JASA 2000.

[6] Sen, Pranab K.; Keating, J. P.; Mason, R. L. (1993). *Pitman's measure of closeness: A comparison of statistical estimators*. Philadelphia: SIAM.

[7] Bickel & Doksum (2001, p. 32)

[8]
- Kiefer, J. and Schwartz, R. (1965). "Admissible Bayes Character of T^2-, R^2-, and Other Fully Invariant Tests for Multivariate Normal Problems". *Annals of Mathematical Statistics* **36**: 747–770. doi:10.1214/aoms/1177700051.
- Schwartz, R. (1969). "Invariant Proper Bayes Tests for Exponential Families". *Annals of Mathematical Statistics* **40**: 270–283. doi:10.1214/aoms/1177697822.
- Hwang, J. T. and Casella, George (1982). "Minimax Confidence Sets for the Mean of a Multivariate Normal Distribution". *Annals of Statistics* **10**: 868–881. doi:10.1214/aos/1176345877.

[9] Lehmann, Erich (1986). *Testing Statistical Hypotheses* (Second ed.). (see p. 309 of Chapter 6.7 "Admissibilty", and pp. 17–18 of Chapter 1.8 "Complete Classes"

[10] Le Cam, Lucien (1986). *Asymptotic Methods in Statistical Decision Theory*. Springer-Verlag. ISBN 0-387-96307-3. (From "Chapter 12 Posterior Distributions and Bayes Solutions", p. 324)

[11] Cox, D. R. and Hinkley, D.V (1974). *Theoretical Statistics*. Chapman and Hall. ISBN 0-04-121537-0. page 432

[12] Cox, D. R. and Hinkley, D. V. (1974). *Theoretical Statistics*. Chapman and Hall. ISBN 0-04-121537-0. p. 433)

[13] Jim Albert (2009). *Bayesian Computation with R, Second edition*. New York, Dordrecht, etc.: Springer. ISBN 978-0-387-92297-3.

[14] Samuel Rathmanner and Marcus Hutter. "A Philosophical Treatise of Universal Induction". *Entropy*, 13(6):1076–1136, 2011.

[15] "The Problem of Old Evidence", in §5 of "On Universal Prediction and Bayesian Confirmation", M. Hutter - Theoretical Computer Science, 2007 - Elsevier

[16] "Raymond J. Solomonoff", Peter Gacs, Paul M. B. Vitanyi, 2011 cs.bu.edu

[17] Dawid, A. P. and Mortera, J. (1996) "Coherent Analysis of Forensic Identification Evidence". *Journal of the Royal Statistical Society*, Series B, 58, 425–443.

[18] Foreman, L. A.; Smith, A. F. M., and Evett, I. W. (1997). "Bayesian analysis of deoxyribonucleic acid profiling data in forensic identification applications (with discussion)". *Journal of the Royal Statistical Society*, Series A, 160, 429–469.

[19] Robertson, B. and Vignaux, G. A. (1995) *Interpreting Evidence: Evaluating Forensic Science in the Courtroom*. John Wiley and Sons. Chichester. ISBN 978-0-471-96026-3

[20] Dawid, A. P. (2001) "Bayes' Theorem and Weighing Evidence by Juries"; http://128.40.111.250/evidence/content/dawid-paper.pdf

[21] Gardner-Medwin, A. (2005) "What Probability Should the Jury Address?". *Significance*, 2 (1), March 2005

[22] David Miller: *Critical Rationalism*

[23] Howson & Urbach (2005), Jaynes (2003)

[24] Stigler, Stephen M. (1986). "Chapter 3". *The History of Statistics*. Harvard University Press.

[25] Fienberg, Stephen E. (2006). "When did Bayesian Inference Become 'Bayesian'?" (PDF). *Bayesian Analysis* **1** (1): 1–40 [p. 5]. doi:10.1214/06-ba101.

[26] Bernardo, José-Miguel (2005). "Reference analysis". *Handbook of statistics* **25**. pp. 17–90.

[27] Wolpert, R. L. (2004). "A Conversation with James O. Berger". *Statistical Science* **19** (1): 205–218. doi:10.1214/088342304000000053. MR 2082155.

[28] Bernardo, José M. (2006). "A Bayesian mathematical statistics primer" (PDF). *ICOTS-7*.

[29] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. New York: Springer. ISBN 0387310738.

2.12 References

- Aster, Richard; Borchers, Brian, and Thurber, Clifford (2012). *Parameter Estimation and Inverse Problems*, Second Edition, Elsevier. ISBN 0123850487, ISBN 978-0123850485
- Bickel, Peter J. and Doksum, Kjell A. (2001). *Mathematical Statistics, Volume 1: Basic and Selected Topics* (Second (updated printing 2007) ed.). Pearson Prentice–Hall. ISBN 0-13-850363-X.
- Box, G. E. P. and Tiao, G. C. (1973) *Bayesian Inference in Statistical Analysis*, Wiley, ISBN 0-471-57428-7
- Edwards, Ward (1968). "Conservatism in Human Information Processing". In Kleinmuntz, B. *Formal Representation of Human Judgment*. Wiley.
- Edwards, Ward (1982). "Conservatism in Human Information Processing (excerpted)". In Daniel Kahneman, Paul Slovic and Amos Tversky. *Judgment under uncertainty: Heuristics and biases*. Cambridge University Press.
- Jaynes E. T. (2003) *Probability Theory: The Logic of Science*, CUP. ISBN 978-0-521-59271-0 (Link to Fragmentary Edition of March 1996).
- Howson, C. and Urbach, P. (2005). *Scientific Reasoning: the Bayesian Approach* (3rd ed.). Open Court Publishing Company. ISBN 978-0-8126-9578-6.
- Phillips, L. D.; Edwards, Ward (October 2008). "Chapter 6: Conservatism in a Simple Probability Inference Task (*Journal of Experimental Psychology* (1966) 72: 346-354)". In Jie W. Weiss and David J. Weiss. *A Science of Decision Making:The Legacy of Ward Edwards*. Oxford University Press. p. 536. ISBN 978-0-19-532298-9.

2.13 Further reading

2.13.1 Elementary

The following books are listed in ascending order of probabilistic sophistication:

- Stone, JV (2013), "Bayes' Rule: A Tutorial Introduction to Bayesian Analysis", Download first chapter here, Sebtel Press, England.
- Colin Howson and Peter Urbach (2005). *Scientific Reasoning: The Bayesian Approach* (3rd ed.). Open Court Publishing Company. ISBN 978-0-8126-9578-6.
- Berry, Donald A. (1996). *Statistics: A Bayesian Perspective*. Duxbury. ISBN 0-534-23476-3.
- Morris H. DeGroot and Mark J. Schervish (2002). *Probability and Statistics* (third ed.). Addison-Wesley. ISBN 978-0-201-52488-8.
- Bolstad, William M. (2007) *Introduction to Bayesian Statistics*: Second Edition, John Wiley ISBN 0-471-27020-2

- Winkler, Robert L (2003). *Introduction to Bayesian Inference and Decision* (2nd ed.). Probabilistic. ISBN 0-9647938-4-9. Updated classic textbook. Bayesian theory clearly presented.
- Lee, Peter M. *Bayesian Statistics: An Introduction*. Fourth Edition (2012), John Wiley ISBN 978-1-1183-3257-3
- Carlin, Bradley P. and Louis, Thomas A. (2008). *Bayesian Methods for Data Analysis, Third Edition*. Boca Raton, FL: Chapman and Hall/CRC. ISBN 1-58488-697-8.
- Gelman, Andrew; Carlin, John B.; Stern, Hal S.; Dunson, David B.; Vehtari, Aki; Rubin, Donald B. (2013). *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC. ISBN 978-1-4398-4095-5.

2.13.2 Intermediate or advanced

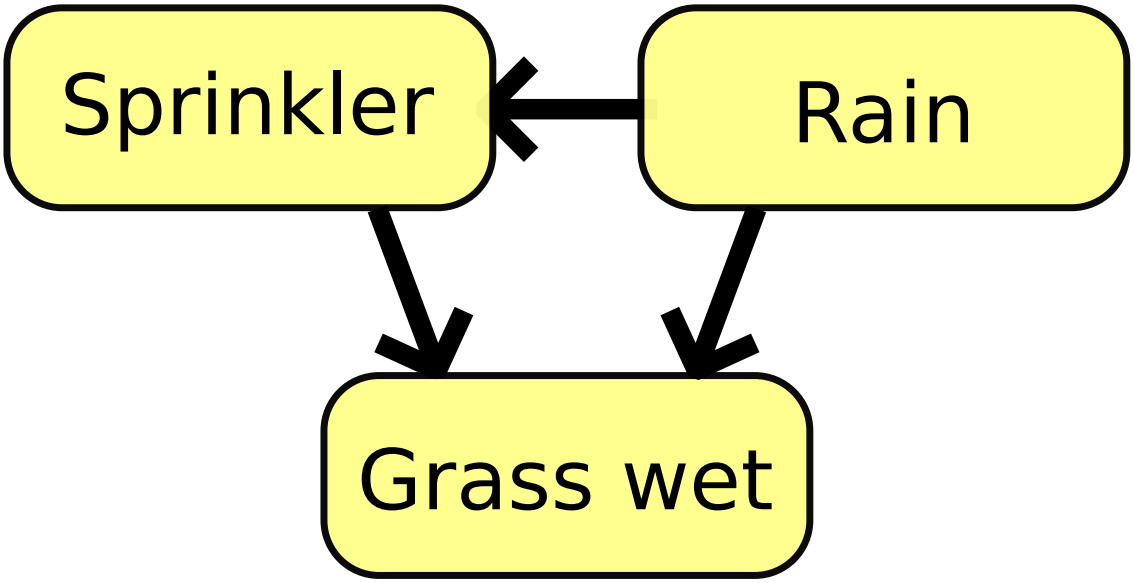
- Berger, James O (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics (Second ed.). Springer-Verlag. ISBN 0-387-96098-8.
- Bernardo, José M.; Smith, Adrian F. M. (1994). *Bayesian Theory*. Wiley.
- DeGroot, Morris H., *Optimal Statistical Decisions*. Wiley Classics Library. 2004. (Originally published (1970) by McGraw-Hill.) ISBN 0-471-68029-X.
- Schervish, Mark J. (1995). *Theory of statistics*. Springer-Verlag. ISBN 0-387-94546-6.
- Jaynes, E. T. (1998) *Probability Theory: The Logic of Science*.
- O'Hagan, A. and Forster, J. (2003) *Kendall's Advanced Theory of Statistics*, Volume 2B: *Bayesian Inference*. Arnold, New York. ISBN 0-340-52922-9.
- Robert, Christian P (2001). *The Bayesian Choice – A Decision-Theoretic Motivation* (second ed.). Springer. ISBN 0-387-94296-3.
- Glenn Shafer and Pearl, Judea, eds. (1988) *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA: Morgan Kaufmann.
- Pierre Bessière et al. (2013), "Bayesian Programming", CRC Press. ISBN 9781439880326

2.14 External links

- Hazewinkel, Michiel, ed. (2001), "Bayesian approach to statistical problems", *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- Bayesian Statistics from Scholarpedia.
- Introduction to Bayesian probability from Queen Mary University of London
- Mathematical Notes on Bayesian Statistics and Markov Chain Monte Carlo
- Bayesian reading list, categorized and annotated by Tom Griffiths
- A. Hajek and S. Hartmann: Bayesian Epistemology, in: J. Dancy et al. (eds.), A Companion to Epistemology. Oxford: Blackwell 2010, 93-106.
- S. Hartmann and J. Sprenger: Bayesian Epistemology, in: S. Bernecker and D. Pritchard (eds.), Routledge Companion to Epistemology. London: Routledge 2010, 609-620.
- *Stanford Encyclopedia of Philosophy*: "Inductive Logic"
- Bayesian Confirmation Theory
- What Is Bayesian Learning?

Chapter 3

Bayesian network



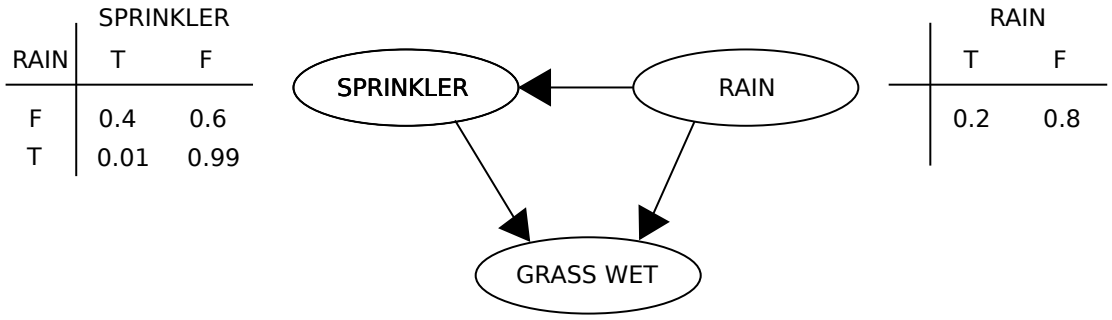
A simple Bayesian network. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.

A **Bayesian network**, **Bayes network**, **belief network**, **Bayes(ian) model** or **probabilistic directed acyclic graphical model** is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Formally, Bayesian networks are DAGs whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not connected represent variables that are conditionally independent of each other. Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node. For example, if m parent nodes represent m Boolean variables then the probability function could be represented by a table of 2^m entries, one entry for each of the 2^m possible combinations of its parents being true or false. Similar ideas may be applied to undirected, and possibly cyclic, graphs; such are called Markov networks.

Efficient algorithms exist that perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (*e.g.* speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

3.1 Example



		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

A simple Bayesian network with conditional probability tables

Suppose that there are two events which could cause grass to be wet: either the sprinkler is on or it's raining. Also, suppose that the rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler is usually not turned on). Then the situation can be modeled with a Bayesian network (shown). All three variables have two possible values, T (for true) and F (for false).

The joint probability function is:

$$\mathrm{P}(G, S, R) = \mathrm{P}(G|S, R)\mathrm{P}(S|R)\mathrm{P}(R)$$

where the names of the variables have been abbreviated to *G* = *Grass wet (yes/no)*, *S* = *Sprinkler turned on (yes/no)*, and *R* = *Raining (yes/no)*.

The model can answer questions like "What is the probability that it is raining, given the grass is wet?" by using the conditional probability formula and summing over all nuisance variables:

$$\mathrm{P}(R = T \mid G = T) = \frac{\mathrm{P}(G = T, R = T)}{\mathrm{P}(G = T)} = \frac{\sum_{S \in \{T,F\}} \mathrm{P}(G = T, S, R = T)}{\sum_{S,R \in \{T,F\}} \mathrm{P}(G = T, S, R)}$$

Using the expansion for the joint probability function $\mathrm{P}(G, S, R)$ and the conditional probabilities from the conditional probability tables (CPTs) stated in the diagram, one can evaluate each term in the sums in the numerator and denominator. For example,

$$\begin{aligned}
&\mathrm{P}(G = T, S = T, R = T) \\
&\quad = \mathrm{P}(G = T|S = T, R = T)\mathrm{P}(S = T|R = T)\mathrm{P}(R = T) \\
&\quad = 0.99 \times 0.01 \times 0.2 \\
&\quad = 0.00198.
\end{aligned}$$

Then the numerical results (subscripted by the associated variable values) are

$$\begin{aligned}\mathrm{P}(R = T \mid G = T) &= \frac{0.00198_{TTT} + 0.1584_{TFT}}{0.00198_{TTT} + 0.288_{TTF} + 0.1584_{TFT} + 0.0_{TFF}} \\ &= \frac{891}{2491} \approx 35.77\%.\end{aligned}$$

If, on the other hand, we wish to answer an interventional question: "What is the likelihood that it would rain, given that we wet the grass?" the answer would be governed by the post-intervention joint distribution function $\mathrm{P}(S, R|do(G = T)) = P(S|R)P(R)$ obtained by removing the factor $\mathrm{P}(G|S, R)$ from the pre-intervention distribution. As expected, the likelihood of rain is unaffected by the action: $\mathrm{P}(R|do(G = T)) = P(R)$.

If, moreover, we wish to predict the impact of turning the sprinkler on, we have

$$P(R, G|do(S = T)) = P(R)P(G|R, S = T)$$

with the term $P(S = T|R)$ removed, showing that the action has an effect on the grass but not on the rain.

These predictions may not be feasible when some of the variables are unobserved, as in most policy evaluation problems. The effect of the action $do(x)$ can still be predicted, however, whenever a criterion called "back-door" is satisfied.[1][2] It states that, if a set *Z* of nodes can be observed that *d*-separates[3] (or blocks) all back-door paths from *X* to *Y* then $P(Y, Z|do(x)) = P(Y, Z, X = x)/P(X = x|Z)$. A back-door path is one that ends with an arrow into *X*. Sets that satisfy the back-door criterion are called "sufficient" or "admissible." For example, the set *Z* = *R* is admissible for predicting the effect of *S* = *T* on *G*, because *R* *d*-separate the (only) back-door path $S \leftarrow R \rightarrow G$. However, if *S* is not observed, there is no other set that *d*-separates this path and the effect of turning the sprinkler on (*S* = *T*) on the grass (*G*) cannot be predicted from passive observations. We then say that *P*(*G*|*do*(*S* = *T*)) is not "identified." This reflects the fact that, lacking interventional data, we cannot determine if the observed dependence between *S* and *G* is due to a causal connection or is spurious (apparent dependence arising from a common cause, *R*). (see Simpson's paradox)

To determine whether a causal relation is identified from an arbitrary Bayesian network with unobserved variables, one can use the three rules of "*do*-calculus"[1][4] and test whether all *do* terms can be removed from the expression of that relation, thus confirming that the desired quantity is estimable from frequency data.[5]

Using a Bayesian network can save considerable amounts of memory, if the dependencies in the joint distribution are sparse. For example, a naive way of storing the conditional probabilities of 10 two-valued variables as a table requires storage space for $2^{10} = 1024$ values. If the local distributions of no variable depends on more than 3 parent variables, the Bayesian network representation only needs to store at most $10 \cdot 2^3 = 80$ values.

One advantage of Bayesian networks is that it is intuitively easier for a human to understand (a sparse set of) direct dependencies and local distributions than complete joint distributions.

3.2 Inference and learning

There are three main inference tasks for Bayesian networks.

3.2.1 Inferring unobserved variables

Because a Bayesian network is a complete model for the variables and their relationships, it can be used to answer probabilistic queries about them. For example, the network can be used to find out updated knowledge of the state of a subset of variables when other variables (the *evidence* variables) are observed. This process of computing the *posterior* distribution of variables given evidence is called probabilistic inference. The posterior gives a universal sufficient statistic for detection applications, when one wants to choose values for the variable subset which minimize some expected loss function, for instance the probability of decision error. A Bayesian network can thus be considered a mechanism for automatically applying Bayes' theorem to complex problems.

The most common exact inference methods are: variable elimination, which eliminates (by integration or summation) the non-observed non-query variables one by one by distributing the sum over the product; clique tree propagation, which caches the computation so that many variables can be queried at one time and new evidence can be propagated quickly; and recursive conditioning and AND/OR search, which allow for a space-time tradeoff and match the

efficiency of variable elimination when enough space is used. All of these methods have complexity that is exponential in the network's treewidth. The most common approximate inference algorithms are importance sampling, stochastic MCMC simulation, mini-bucket elimination, loopy belief propagation, generalized belief propagation, and variational methods.

3.2.2 Parameter learning

In order to fully specify the Bayesian network and thus fully represent the joint probability distribution, it is necessary to specify for each node X the probability distribution for X conditional upon X's parents. The distribution of X conditional upon its parents may have any form. It is common to work with discrete or Gaussian distributions since that simplifies calculations. Sometimes only constraints on a distribution are known; one can then use the principle of maximum entropy to determine a single distribution, the one with the greatest entropy given the constraints. (Analogously, in the specific context of a dynamic Bayesian network, one commonly specifies the conditional distribution for the hidden state's temporal evolution to maximize the entropy rate of the implied stochastic process.)

Often these conditional distributions include parameters which are unknown and must be estimated from data, sometimes using the maximum likelihood approach. Direct maximization of the likelihood (or of the posterior probability) is often complex when there are unobserved variables. A classical approach to this problem is the expectation-maximization algorithm which alternates computing expected values of the unobserved variables conditional on observed data, with maximizing the complete likelihood (or posterior) assuming that previously computed expected values are correct. Under mild regularity conditions this process converges on maximum likelihood (or maximum posterior) values for parameters.

A more fully Bayesian approach to parameters is to treat parameters as additional unobserved variables and to compute a full posterior distribution over all nodes conditional upon observed data, then to integrate out the parameters. This approach can be expensive and lead to large dimension models, so in practice classical parameter-setting approaches are more common.

3.2.3 Structure learning

In the simplest case, a Bayesian network is specified by an expert and is then used to perform inference. In other applications the task of defining the network is too complex for humans. In this case the network structure and the parameters of the local distributions must be learned from data.

Automatically learning the graph structure of a Bayesian network is a challenge pursued within machine learning. The basic idea goes back to a recovery algorithm developed by Rebane and Pearl (1987)[6] and rests on the distinction between the three possible types of adjacent triplets allowed in a directed acyclic graph (DAG):

1. $X \rightarrow Y \rightarrow Z$
2. $X \leftarrow Y \rightarrow Z$
3. $X \rightarrow Y \leftarrow Z$

Type 1 and type 2 represent the same dependencies (X and Z are independent given Y) and are, therefore, indistinguishable. Type 3, however, can be uniquely identified, since X and Z are marginally independent and all other pairs are dependent. Thus, while the *skeletons* (the graphs stripped of arrows) of these three triplets are identical, the directionality of the arrows is partially identifiable. The same distinction applies when X and Z have common parents, except that one must first condition on those parents. Algorithms have been developed to systematically determine the skeleton of the underlying graph and, then, orient all arrows whose directionality is dictated by the conditional independencies observed.[1][7][8][9]

An alternative method of structural learning uses optimization based search. It requires a scoring function and a search strategy. A common scoring function is posterior probability of the structure given the training data. The time requirement of an exhaustive search returning a structure that maximizes the score is superexponential in the number of variables. A local search strategy makes incremental changes aimed at improving the score of the structure. A global search algorithm like Markov chain Monte Carlo can avoid getting trapped in local minima. Friedman et al.[10][11] discuss using mutual information between variables and finding a structure that maximizes this. They do this by restricting the parent candidate set to k nodes and exhaustively searching therein.

Another method consists of focusing on the sub-class of decomposable models, for which the MLE have a closed form. It is then possible to discover a consistent structure for hundreds of variables.[12]

A Bayesian network can be augmented with nodes and edges using rule-based machine learning techniques. Inductive logic programming can be used to mine rules and create new nodes.[13] Statistical relational learning (SRL) approaches use a scoring function based on the Bayes network structure to guide the structural search and augment the network.[14] A common SRL scoring function is the area under the ROC curve.

3.3 Statistical introduction

Given data x and parameter θ , a simple Bayesian analysis starts with a prior probability (*prior*) $p(\theta)$ and likelihood $p(x|\theta)$ to compute a posterior probability $p(\theta|x) \propto p(x|\theta)p(\theta)$.

Often the prior on θ depends in turn on other parameters φ that are not mentioned in the likelihood. So, the prior $p(\theta)$ must be replaced by a likelihood $p(\theta|\varphi)$, and a prior $p(\varphi)$ on the newly introduced parameters φ is required, resulting in a posterior probability

$$p(\theta, \varphi|x) \propto p(x|\theta)p(\theta|\varphi)p(\varphi).$$

This is the simplest example of a *hierarchical Bayes model*.

The process may be repeated; for example, the parameters φ may depend in turn on additional parameters ψ , which will require their own prior. Eventually the process must terminate, with priors that do not depend on any other unmentioned parameters.

3.3.1 Introductory examples

Suppose we have measured the quantities $x_1, \ldots, x_n$ each with normally distributed errors of known standard deviation σ ,

$$x_i \sim N(\theta_i, \sigma^2)$$

Suppose we are interested in estimating the θ_i . An approach would be to estimate the θ_i using a maximum likelihood approach; since the observations are independent, the likelihood factorizes and the maximum likelihood estimate is simply

$$\theta_i = x_i$$

However, if the quantities are related, so that for example we may think that the individual θ_i have themselves been drawn from an underlying distribution, then this relationship destroys the independence and suggests a more complex model, e.g.,

$$x_i \sim N(\theta_i, \sigma^2),$$
$$\theta_i \sim N(\varphi, \tau^2)$$

with improper priors $\varphi \sim$ flat, $\tau \sim$ flat $\in (0, \infty)$. When $n \geq 3$, this is an identified model (i.e. there exists a unique solution for the model's parameters), and the posterior distributions of the individual θ_i will tend to move, or *shrink* away from the maximum likelihood estimates towards their common mean. This *shrinkage* is a typical behavior in hierarchical Bayes models.

3.3.2 Restrictions on priors

Some care is needed when choosing priors in a hierarchical model, particularly on scale variables at higher levels of the hierarchy such as the variable τ in the example. The usual priors such as the Jeffreys prior often do not work, because the posterior distribution will be improper (not normalizable), and estimates made by minimizing the expected loss will be inadmissible.

3.4 Definitions and concepts

See also: Glossary of graph theory § Directed acyclic graphs

There are several equivalent definitions of a Bayesian network. For all the following, let $G = (V,E)$ be a directed acyclic graph (or DAG), and let $X = (Xv)v \in V$ be a set of random variables indexed by V.

3.4.1 Factorization definition

X is a Bayesian network with respect to G if its joint probability density function (with respect to a product measure) can be written as a product of the individual density functions, conditional on their parent variables:[15]

$p(x) = \prod_{v \in V} p\left(x_v \mid x_{\mathrm{pa}(v)}\right)$

where pa(v) is the set of parents of v (i.e. those vertices pointing directly to v via a single edge).

For any set of random variables, the probability of any member of a joint distribution can be calculated from conditional probabilities using the chain rule (given a topological ordering of X) as follows:[15]

$\mathrm{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{v=1}^{n} \mathrm{P}\left(X_v = x_v \mid X_{v+1} = x_{v+1}, \ldots, X_n = x_n\right)$

Compare this with the definition above, which can be written as:

$\mathrm{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{v=1}^{n} \mathrm{P}(X_v = x_v \mid X_j = x_j \text{ for each } X_j \text{ which is a parent of } X_v)$

The difference between the two expressions is the conditional independence of the variables from any of their non-descendants, given the values of their parent variables.

3.4.2 Local Markov property

X is a Bayesian network with respect to G if it satisfies the *local Markov property*: each variable is conditionally independent of its non-descendants given its parent variables:[16]

$X_v \perp\!\!\!\perp X_{V \setminus \mathrm{de}(v)} \mid X_{\mathrm{pa}(v)} \quad \text{all for} v \in V$

where de(v) is the set of descendants and $V \setminus$ de(v) is the set of non-descendants of v.

This can also be expressed in terms similar to the first definition, as

> $\mathrm{P}(X_v = x_v \mid X_i = x_i \text{ for each } X_i \text{ which is not a descendant of } X_v) = P(X_v = x_v \mid X_j = x_j \text{ for each } X_j \text{ which is a parent of } X_v)$

Note that the set of parents is a subset of the set of non-descendants because the graph is acyclic.

3.4.3 Developing Bayesian networks

To develop a Bayesian network, we often first develop a DAG G such that we believe X satisfies the local Markov property with respect to G. Sometimes this is done by creating a causal DAG. We then ascertain the conditional probability distributions of each variable given its parents in G. In many cases, in particular in the case where the variables are discrete, if we define the joint distribution of X to be the product of these conditional distributions, then X is a Bayesian network with respect to G.[17]

3.4.4 Markov blanket

The Markov blanket of a node is the set of nodes consisting of its parents, its children, and any other parents of its children. This set renders it independent of the rest of the network; the joint distribution of the variables in the Markov blanket of a node is sufficient knowledge for calculating the distribution of the node. X is a Bayesian network with respect to G if every node is conditionally independent of all other nodes in the network, given its Markov blanket.[16]

d-separation

This definition can be made more general by defining the "d"-separation of two nodes, where d stands for directional.[18][19] Let P be a trail (that is, a collection of edges which is like a path, but each of whose edges may have any direction) from node u to v. Then P is said to be d-separated by a set of nodes Z if and only if (at least) one of the following holds:

1. P contains a *chain*, $u \leftarrow m \leftarrow v$, such that the middle node m is in Z,
2. P contains a *fork*, $u \leftarrow m \rightarrow v$, such that the middle node m is in Z, or
3. P contains an *inverted fork* (or *collider*), $u \rightarrow m \leftarrow v$, such that the middle node m is **not** in Z and no descendant of m is in Z.

Thus u and v are said to be d-separated by Z if all trails between them are d-separated. If u and v are not d-separated, they are called d-connected.

X is a Bayesian network with respect to G if, for any two nodes u, v:

$$X_u \perp\!\!\!\perp X_v \mid X_Z$$

where Z is a set which d-separates u and v. (The Markov blanket is the minimal set of nodes which d-separates node v from all other nodes.)

3.4.5 Hierarchical models

The term *hierarchical model* is sometimes considered a particular type of Bayesian network, but has no formal definition. Sometimes the term is reserved for models with three or more levels of random variables; other times, it is reserved for models with latent variables. In general, however, any moderately complex Bayesian network is usually termed "hierarchical".

3.4.6 Causal networks

Although Bayesian networks are often used to represent causal relationships, this need not be the case: a directed edge from u to v does not require that Xv is causally dependent on Xu. This is demonstrated by the fact that Bayesian networks on the graphs:

$$a \longrightarrow b \longrightarrow c \qquad \text{and} \qquad a \longleftarrow b \longleftarrow c$$

are equivalent: that is they impose exactly the same conditional independence requirements.

A **causal network** is a Bayesian network with an explicit requirement that the relationships be causal. The additional semantics of the causal networks specify that if a node X is actively caused to be in a given state x (an action written as $do(X{=}x)$), then the probability density function changes to the one of the network obtained by cutting the links from the parents of X to X, and setting X to the caused value x.[1] Using these semantics, one can predict the impact of external interventions from data obtained prior to intervention.

3.5 Applications

Bayesian networks are used for modelling beliefs in computational biology and bioinformatics (gene regulatory networks, protein structure, gene expression analysis,[20] learning epistasis from GWAS data sets[21]) medicine,[22] biomonitoring,[23] document classification, information retrieval,[24] semantic search,[25] image processing, data fusion, decision support systems,[26] engineering, sports betting,[27][28] gaming, law,[29][30][31] study design[32] and risk analysis.[33][34][35] There are texts applying Bayesian networks to bioinformatics[36] and financial and marketing informatics.[37]

3.5.1 Software

- WinBUGS
- OpenBUGS (website), further (open source) development of WinBUGS.
- OpenMarkov, open source software and API implemented in Java
- Graphical Models Toolkit (GMTK) — GMTK is an open source, publicly available toolkit for rapidly prototyping statistical models using dynamic graphical models (DGMs) and dynamic Bayesian networks (DBNs). GMTK can be used for applications and research in speech and language processing, bioinformatics, activity recognition, and any time series application.
- Just another Gibbs sampler (JAGS) (website)
- Stan (software) (website) — Stan is an open-source package for obtaining Bayesian inference using the No-U-Turn sampler, a variant of Hamiltonian Monte Carlo. It's somewhat like BUGS, but with a different language for expressing models and a different sampler for sampling from their posteriors. RStan is the R interface to Stan.
- PyMC — PyMC is a python module that implements Bayesian statistical models and fitting algorithms, including Markov chain Monte Carlo. Its flexibility and extensibility make it applicable to a large suite of problems. Along with core sampling functionality, PyMC includes methods for summarizing output, plotting, goodness-of-fit and convergence diagnostics.
- GeNIe&Smile (website) — SMILE is a C++ library for BN and ID, and GeNIe is a GUI for it
- SamIam (website), a Java-based system with GUI and Java API
- Bayes Server - User Interface and API for Bayesian networks, includes support for time series and sequences
- Belief and Decision Networks on AIspace
- BayesiaLab by Bayesia
- Hugin
- Netica by Norsys
- dVelox by Apara Software
- System Modeler by Inatas AB
- UnBBayes by GIA-UnB (Intelligence Artificial Group - University of Brasilia)

3.6 History

The term "Bayesian networks" was coined by Judea Pearl in 1985 to emphasize three aspects:[38]

1. The often subjective nature of the input information.
2. The reliance on Bayes' conditioning as the basis for updating information.
3. The distinction between causal and evidential modes of reasoning, which underscores Thomas Bayes' posthumously published paper of 1763.[39]

In the late 1980s Judea Pearl's text *Probabilistic Reasoning in Intelligent Systems*[40] and Richard E. Neapolitan's text *Probabilistic Reasoning in Expert Systems*[41] summarized the properties of Bayesian networks and established Bayesian networks as a field of study.

Informal variants of such networks were first used by legal scholar John Henry Wigmore, in the form of Wigmore charts, to analyse trial evidence in 1913.[30]:66–76 Another variant, called path diagrams, was developed by the geneticist Sewall Wright[42] and used in social and behavioral sciences (mostly with linear parametric models).

3.7 See also

- Artificial intelligence
- Bayes' theorem
- Bayesian inference
- Bayesian probability
- Bayesian programming
- Belief propagation
- Causal loop diagram
- Chow–Liu tree
- Computational intelligence
- Computational phylogenetics
- Deep belief network
- Dempster–Shafer theory – a Generalization of Bayes' theorem
- Dynamic Bayesian network
- Expectation–maximization algorithm
- Factor graph
- Graphical model
- Hierarchical temporal memory
- Influence diagram
- Judea Pearl
- Kalman filter
- Machine learning
- Memory-prediction framework
- Mixture distribution
- Mixture model
- Naive Bayes classifier
- Path analysis
- Polytree
- Sensor fusion
- Sequence alignment
- Speech recognition
- Structural equation modeling
- Subjective logic
- Variable-order Bayesian network
- Wigmore chart
- World view

3.8 Notes

[1] Pearl, Judea (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press. ISBN 0-521-77362-8. OCLC 42291253.

[2] "The Back-Door Criterion" (PDF). Retrieved 2014-09-18.

[3] "d-Separation without Tears" (PDF). Retrieved 2014-09-18.

[4] J., Pearl (1994). "A Probabilistic Calculus of Actions". In Lopez de Mantaras, R.; Poole, D. *UAI'94 Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. San Mateo CA: Morgan Kaufman. pp. 454–462. ISBN 1-55860-332-8.

[5] I. Shpitser, J. Pearl, "Identification of Conditional Interventional Distributions" In R. Dechter and T.S. Richardson (Eds.), *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 437–444, Corvallis, OR: AUAI Press, 2006.

[6] Rebane, G. and Pearl, J., "The Recovery of Causal Poly-trees from Statistical Data," *Proceedings, 3rd Workshop on Uncertainty in AI,* (Seattle, WA) pages 222–228, 1987

[7] Spirtes, P.; Glymour, C. (1991). "An algorithm for fast recovery of sparse causal graphs" (PDF). *Social Science Computer Review* **9** (1): 62–72. doi:10.1177/089443939100900106.

[8] Spirtes, Peter; Glymour, Clark N.; Scheines, Richard (1993). *Causation, Prediction, and Search* (1st ed.). Springer-Verlag. ISBN 978-0-387-97979-3.

[9] Verma, Thomas; Pearl, Judea (1991). "Equivalence and synthesis of causal models". In Bonissone, P.; Henrion, M.; Kanal, L.N.; Lemmer, J.F. *UAI '90 Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*. Elsevier. pp. 255–270. ISBN 0-444-89264-8.

[10] Friedman, Nir; Geiger, Dan; Goldszmidt, Moises (November 1997). "Bayesian Network Classifiers". *Machine Learning* **29** (2-3): 131–163. doi:10.1023/A:1007465528199. Retrieved 24 February 2015.

[11] Friedman, Nir; Linial, Michal; Nachman, Iftach; Pe'er, Dana (August 2000). "Using Bayesian Networks to Analyze Expression Data". *Journal of Computational Biology* **7** (3-4): 601–620. doi:10.1089/106652700750050961. Retrieved 24 February 2015.

[12] Petitjean, F.; Webb, G.I.; Nicholson, A.E. (2013). *Scaling log-linear analysis to high-dimensional data* (PDF). International Conference on Data Mining. Dallas, TX, USA: IEEE.

[13] Nassif, Houssam; Wu, Yirong; Page, David; Burnside, Elizabeth (2012). "Logical Differential Prediction Bayes Net, Improving Breast Cancer Diagnosis for Older Women" (PDF). *American Medical Informatics Association Symposium (AMIA'12)* (Chicago): 1330–1339. Retrieved 18 July 2014.

[14] Nassif, Houssam; Kuusisto, Finn; Burnside, Elizabeth S; Page, David; Shavlik, Jude; Santos Costa, Vitor (2013). "Score As You Lift (SAYL): A Statistical Relational Learning Approach to Uplift Modeling" (PDF). *European Conference on Machine Learning (ECML'13)* (Prague): 595–611.

[15] Russell & Norvig 2003, p. 496.

[16] Russell & Norvig 2003, p. 499.

[17] Neapolitan, Richard E. (2004). *Learning Bayesian networks*. Prentice Hall. ISBN 978-0-13-012534-7.

[18] Geiger, Dan; Verma, Thomas; Pearl, Judea (1990). "Identifying independence in Bayesian Networks" (PDF). *Networks* **20**: 507–534. doi:10.1177/089443939100900106.

[19] Richard Scheines, *D-separation*

[20] Friedman, N.; Linial, M.; Nachman, I.; Pe'er, D. (2000). "Using Bayesian Networks to Analyze Expression Data". *Journal of Computational Biology* **7** (3–4): 601–620. doi:10.1089/106652700750050961. PMID 11108481.

[21] Jiang, X.; Neapolitan, R.E.; Barmada, M.M.; Visweswaran, S. (2011). "Learning Genetic Epistasis using Bayesian Network Scoring Criteria". *BMC Bioinformatics* **12**: 89. doi:10.1186/1471-2105-12-89. PMC 3080825. PMID 21453508.

[22] J. Uebersax (2004). *Genetic Counseling and Cancer Risk Modeling: An Application of Bayes Nets*. Marbella, Spain: Ravenpack International.

[23] Jiang X, Cooper GF. (July–August 2010). "A Bayesian spatio-temporal method for disease outbreak detection". *J Am Med Inform Assoc* **17** (4): 462–71. doi:10.1136/jamia.2009.000356. PMC 2995651. PMID 20595315.

[24] Luis M. de Campos, Juan M. Fernández-Luna and Juan F. Huete (2004). "Bayesian networks and information retrieval: an introduction to the special issue". *Information Processing & Management* (Elsevier) **40** (5): 727–733. doi:10.1016/j.ipm.2004.03.001. ISBN 0-471-14182-8.

[25] Christos L. Koumenides and Nigel R. Shadbolt. 2012. Combining link and content-based information in a Bayesian inference model for entity search. In Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search (JIWES '12). ACM, New York, NY, USA, , Article 3 , 6 pages. doi:10.1145/2379307.2379310

[26] F.J. Díez, J. Mira, E. Iturralde and S. Zubillaga (1997). "DIAVAL, a Bayesian expert system for echocardiography". *Artificial Intelligence in Medicine* (Elsevier) **10** (1): 59–73. doi:10.1016/s0933-3657(97)00384-9. PMID 9177816.

[27] Constantinou, Anthony; Fenton, N.; Neil, M. (2012). "pi-football: A Bayesian network model for forecasting Association Football match outcomes". *Knowledge-Based Systems* **36**: 322–339. doi:10.1016/j.knosys.2012.07.008. Retrieved 25 March 2014.

[28] Constantinou, Anthony; Fenton, N.; Neil, M. (2013). "Profiting from an inefficient Association Football gambling market: Prediction, Risk and Uncertainty using Bayesian networks.". *Knowledge-Based Systems* **50**: 60–86. doi:10.1016/j.knosys.2013.05.008. Retrieved 25 March 2014.

[29] G. A. Davis (2003). "Bayesian reconstruction of traffic accidents". *Law, Probability and Risk* **2** (2): 69–89. doi:10.1093/lpr/2.2.69.

[30] J. B. Kadane and D. A. Schum (1996). *A Probabilistic Analysis of the Sacco and Vanzetti Evidence*. New York: Wiley. ISBN 0-471-14182-8.

[31] O. Pourret, P. Naim and B. Marcot (2008). *Bayesian Networks: A Practical Guide to Applications*. Chichester, UK: Wiley. ISBN 978-0-470-06030-8.

[32] Karvanen, Juha (2014). "Study design in causal models". *Scandinavian Journal of Statistics*. doi:10.1111/sjos.12110.

[33] Cardenas, IC; Al-Jibouri, SHS; Halman, JIM; van Tol, FA (2014). "Modeling Risk-Related Knowledge in Tunneling Projects". *Risk Analysis* **34** (2): 323–339. doi:10.1111/risa.12094.

[34] Cardenas, IC; Al-Jibouri, SHS; Halman, JIM; van de Linde, W; Kaalberg, F (2014). "Using Prior Risk-Related Knowledge to Support Risk Management Decisions: Lessons Learnt from a Tunneling Project". *Risk Analysis* **34** (8). doi:10.1111/risa.12213.

[35] Cardenas, IC (2015). "Modeling the Influence of Unknown Factors in Risk Analysis Using Bayesian Networks". *Under review by a refereed journal*.

[36] Neapolitan, Richard (2009). *Probabilistic Methods for Bioinformatics*. Burlington, MA: Morgan Kaufmann. p. 406. ISBN 9780123704764.

[37] Neapolitan, Richard, and Xia Jiang (2007). *Probabilistic Methods for Financial and Marketing Informatics*. Burlingon, MA: Morgan Kaufmann. p. 432. ISBN 0123704774.

[38] Pearl, J. (1985). *Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning* (UCLA TECHNICAL REPORT CSD-850017). Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA. pp. 329–334. Retrieved 2009-05-01.

[39] Bayes, T.; Price, Mr. (1763). "An Essay towards solving a Problem in the Doctrine of Chances". *Philosophical Transactions of the Royal Society* **53**: 370–418. doi:10.1098/rstl.1763.0053.

[40] Pearl, J. *Probabilistic Reasoning in Intelligent Systems*. San Francisco CA: Morgan Kaufmann. p. 1988. ISBN 1558604790.

[41] Neapolitan, Richard E. (1989). *Probabilistic reasoning in expert systems: theory and algorithms*. Wiley. ISBN 978-0-471-61840-9.

[42] Wright, S. (1921). "Correlation and Causation" (PDF). *Journal of Agricultural Research* **20** (7): 557–585.

3.9 References

- Ben-Gal, Irad (2007). "Bayesian Networks". In Ruggeri, Fabrizio; Kennett, Ron S.; Faltin, Frederick W. *Encyclopedia of Statistics in Quality and Reliability* (PDF). *Encyclopedia of Statistics in Quality and Reliability*. John Wiley & Sons. doi:10.1002/9780470061572.eqr089. ISBN 978-0-470-01861-3.
- Bertsch McGrayne, Sharon. *The Theory That Would not Die*. Yale.

- Borgelt, Christian; Kruse, Rudolf (March 2002). *Graphical Models: Methods for Data Analysis and Mining*. Chichester, UK: Wiley. ISBN 0-470-84337-3.

- Borsuk, Mark Edward (2008). "Ecological informatics: Bayesian networks". In Jørgensen , Sven Erik, Fath, Brian. *Encyclopedia of Ecology*. Elsevier. ISBN 978-0-444-52033-3.

- Cardenas, I. et al. (April 2015). "Modeling the Influence of Unknown Factors in Risk Analysis using Bayesian Networks" (PDF). *Under review by a refereed journal*.

- Castillo, Enrique; Gutiérrez, José Manuel; Hadi, Ali S. (1997). "Learning Bayesian Networks". *Expert Systems and Probabilistic Network Models*. Monographs in computer science. New York: Springer-Verlag. pp. 481–528. ISBN 0-387-94858-9.

- Comley, Joshua W.; Dowe, David L. (October 2003). "Minimum Message Length and Generalized Bayesian Nets with Asymmetric Languages". Written at Victoria, Australia. In Grünwald, Peter D.; Myung, In Jae; Pitt, Mark A. *Advances in Minimum Description Length: Theory and Applications*. Neural information processing series. Cambridge, Massachusetts: Bradford Books (MIT Press) (published April 2005). pp. 265–294. ISBN 0-262-07262-9. (This paper puts decision trees in internal nodes of Bayes networks using Minimum Message Length (MML). An earlier version is Comley and Dowe (2003), .pdf.)

- Darwiche, Adnan (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press. ISBN 978-0521884389.

- Dowe, David L. (2010). MML, hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness, in Handbook of Philosophy of Science (Volume 7: Handbook of Philosophy of Statistics), Elsevier, ISBN 978-0-444-51862-0, pp 901–982.

- Fenton, Norman; Neil, Martin E. (November 2007). *Managing Risk in the Modern World: Applications of Bayesian Networks* – A Knowledge Transfer Report from the London Mathematical Society and the Knowledge Transfer Network for Industrial Mathematics. London (England): London Mathematical Society.

- Fenton, Norman; Neil, Martin E. (July 23, 2004). "Combining evidence in risk analysis using Bayesian Networks" (PDF). *Safety Critical Systems Club Newsletter* **13** (4) (Newcastle upon Tyne, England). pp. 8–13.

- Andrew Gelman; John B Carlin; Hal S Stern; Donald B Rubin (2003). "Part II: Fundamentals of Bayesian Data Analysis: Ch.5 Hierarchical models". *Bayesian Data Analysis*. CRC Press. pp. 120–. ISBN 978-1-58488-388-3.

- Heckerman, David (March 1, 1995). "Tutorial on Learning with Bayesian Networks". In Jordan, Michael Irwin. *Learning in Graphical Models*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: MIT Press (published 1998). pp. 301–354. ISBN 0-262-60032-3..

  Also appears as Heckerman, David (March 1997). "Bayesian Networks for Data Mining". *Data Mining and Knowledge Discovery* (Netherlands: Springer Netherlands) **1** (1): 79–119. doi:10.1023/A:1009730122752. ISSN 1384-5810.

  An earlier version appears as Technical Report MSR-TR-95-06, Microsoft Research, March 1, 1995. The paper is about both parameter and structure learning in Bayesian networks.

- Jensen, Finn V; Nielsen, Thomas D. (June 6, 2007). *Bayesian Networks and Decision Graphs*. Information Science and Statistics series (2nd ed.). New York: Springer-Verlag. ISBN 978-0-387-68281-5.

- Karimi, Kamran; Hamilton, Howard J. (2000). "Finding temporal relations: Causal bayesian networks vs. C4.5" (PDF). *Twelfth International Symposium on Methodologies for Intelligent Systems*.

- Korb, Kevin B.; Nicholson, Ann E. (December 2010). *Bayesian Artificial Intelligence*. CRC Computer Science & Data Analysis (2nd ed.). Chapman & Hall (CRC Press). doi:10.1007/s10044-004-0214-5. ISBN 1-58488-387-1.

- Lunn, D.; Thomas, A; Best, N et al. (2009). "The BUGS project: Evolution, critique and future directions". *Statistics in Medicine* **28** (25): 3049–3067. doi:10.1002/sim.3680. PMID 19630097. |first2= missing |last2= in Authors list (help)

- Neil, Martin; Fenton, Norman E.; Tailor, Manesh (August 2005). Greenberg, Michael R., ed. "Using Bayesian Networks to Model Expected and Unexpected Operational Losses" (PDF). *Risk Analysis: an International Journal* (John Wiley & Sons) **25** (4): 963–972. doi:10.1111/j.1539-6924.2005.00641.x. PMID 16268944.
- Pearl, Judea (September 1986). "Fusion, propagation, and structuring in belief networks". *Artificial Intelligence* (Elsevier) **29** (3): 241–288. doi:10.1016/0004-3702(86)90072-X. ISSN 0004-3702.
- Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Representation and Reasoning Series (2nd printing ed.). San Francisco, California: Morgan Kaufmann. ISBN 0-934613-73-7.
- Pearl, Judea; Russell, Stuart (November 2002). "Bayesian Networks". In Arbib, Michael A. *Handbook of Brain Theory and Neural Networks*. Cambridge, Massachusetts: Bradford Books (MIT Press). pp. 157–160. ISBN 0-262-01197-2.
- Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.
- Zhang, Nevin Lianwen; Poole, David (May 1994). "A simple approach to Bayesian network computations". *Proceedings of the Tenth Biennial Canadian Artificial Intelligence Conference (AI-94).* (Banff, Alberta): 171–178. This paper presents variable elimination for belief networks.

3.10 Further reading

- *Computational Intelligence: A Methodological Introduction* by Kruse, Borgelt, Klawonn, Moewes, Steinbrecher, Held, 2013, Springer, ISBN 9781447150121
- *Graphical Models - Representations for Learning, Reasoning and Data Mining*, 2nd Edition, by Borgelt, Steinbrecher, Kruse, 2009, J. Wiley & Sons, ISBN 9780470749562

3.11 External links

- A tutorial on learning with Bayesian Networks
- An Introduction to Bayesian Networks and their Contemporary Applications
- On-line Tutorial on Bayesian nets and probability
- Web-App to create Bayesian nets and run it with a Monte Carlo method
- Continuous Time Bayesian Networks
- Bayesian Networks: Explanation and Analogy
- A live tutorial on learning Bayesian networks
- A hierarchical Bayes Model for handling sample heterogeneity in classification problems, provides a classification model taking into consideration the uncertainty associated with measuring replicate samples.
- Hierarchical Naive Bayes Model for handling sample uncertainty, shows how to perform classification and learning with continuous and discrete variables with replicated measurements.

Chapter 4

Bayesian probability

Bayesian probability is one interpretation of the concept of probability. The Bayesian interpretation of probability can be seen as an extension of propositional logic that enables reasoning with hypotheses, i.e., the propositions whose truth or falsity is uncertain.

Bayesian probability belongs to the category of evidential probabilities; to evaluate the probability of a hypothesis, the Bayesian probabilist specifies some prior probability, which is then updated in the light of new, relevant data (evidence).[1] The Bayesian interpretation provides a standard set of procedures and formulae to perform this calculation.

In contrast to interpreting probability as the "frequency" or "propensity" of some phenomenon, Bayesian probability is a quantity that we assign for the purpose of representing a state of knowledge,[2] or a state of belief.[3] In the Bayesian view, a probability is assigned to a hypothesis, whereas under the frequentist view, a hypothesis is typically tested without being assigned a probability.

The term "Bayesian" refers to the 18th century mathematician and theologian Thomas Bayes, who provided the first mathematical treatment of a non-trivial problem of Bayesian inference.[4] Mathematician Pierre-Simon Laplace pioneered and popularised what is now called Bayesian probability.[5]

Broadly speaking, there are two views on Bayesian probability that interpret the *probability* concept in different ways. According to the *objectivist view*, the rules of Bayesian statistics can be justified by requirements of rationality and consistency and interpreted as an extension of logic.[2][6] According to the *subjectivist view*, probability quantifies a "personal belief".[3]

4.1 Bayesian methodology

Bayesian methods are characterized by the following concepts and procedures:

- The use of random variables, or, more generally, unknown quantities,[7] to model all sources of uncertainty in statistical models. This also includes uncertainty resulting from lack of information (see also the aleatoric and epistemic uncertainty).
- The need to determine the *prior probability distribution* taking into account the available (prior) information.
- The *sequential use of the Bayes' formula*: when more data becomes available, calculate the *posterior distribution* using the Bayes' formula; subsequently, the posterior distribution becomes the next prior.
- For the frequentist a hypothesis is a proposition (which must be either true or false), so that the frequentist probability of a hypothesis is either one or zero. In Bayesian statistics, a probability can be assigned to a hypothesis that can differ from 0 or 1 if the truth value is uncertain.

4.2 Objective and subjective Bayesian probabilities

Broadly speaking, there are two views on Bayesian probability that interpret the 'probability' concept in different ways. For **objectivists**, *probability* objectively measures the plausibility of propositions, i.e. the probability of a proposition corresponds to a reasonable belief everyone (even a “robot”) sharing the same knowledge should share in accordance with the rules of Bayesian statistics, which can be justified by requirements of rationality and consistency.[2][6] For **subjectivists**, probability corresponds to a 'personal belief'.[3] For subjectivists, rationality and coherence constrain the probabilities a subject may have, but allow for substantial variation within those constraints. The objective and subjective variants of Bayesian probability differ mainly in their interpretation and construction of the prior probability.

4.3 History

Main article: History of statistics § Bayesian statistics

The term *Bayesian* refers to Thomas Bayes (1702–1761), who proved a special case of what is now called Bayes' theorem in a paper titled "An Essay towards solving a Problem in the Doctrine of Chances".[8] In that special case, the prior and posterior distributions were Beta distributions and the data came from Bernoulli trials. It was Pierre-Simon Laplace (1749–1827) who introduced a general version of the theorem and used it to approach problems in celestial mechanics, medical statistics, reliability, and jurisprudence.[9] Early Bayesian inference, which used uniform priors following Laplace's principle of insufficient reason, was called "inverse probability" (because it infers backwards from observations to parameters, or from effects to causes).[10] After the 1920s, “inverse probability” was largely supplanted by a collection of methods that came to be called frequentist statistics.[10]

In the 20th century, the ideas of Laplace were further developed in two different directions, giving rise to *objective* and *subjective* currents in Bayesian practice. Harold Jeffreys' *Theory of Probability* (first published in 1939) played an important role in the revival of the Bayesian view of probability, followed by works by Abraham Wald (1950) and Leonard J. Savage (1954). The adjective *Bayesian* itself dates to the 1950s; the derived *Bayesianism*, *neo-Bayesianism* is of 1960s coinage.[11] In the objectivist stream, the statistical analysis depends on only the model assumed and the data analysed.[12] No subjective decisions need to be involved. In contrast, “subjectivist” statisticians deny the possibility of fully objective analysis for the general case.

In the 1980s, there was a dramatic growth in research and applications of Bayesian methods, mostly attributed to the discovery of Markov chain Monte Carlo methods, which removed many of the computational problems, and an increasing interest in nonstandard, complex applications.[13] Despite the growth of Bayesian research, most undergraduate teaching is still based on frequentist statistics.[14] Nonetheless, Bayesian methods are widely accepted and used, such as in the field of machine learning.[15]

4.4 Justification of Bayesian probabilities

The use of Bayesian probabilities as the basis of Bayesian inference has been supported by several arguments, such as the Cox axioms, the Dutch book argument, arguments based on decision theory and de Finetti's theorem.

4.4.1 Axiomatic approach

Richard T. Cox showed that[6] Bayesian updating follows from several axioms, including two functional equations and a controversial hypothesis of differentiability. It is known that Cox's 1961 development (mainly copied by Jaynes) is non-rigorous, and in fact a counterexample has been found by Halpern.[16] The assumption of differentiability or even continuity is questionable since the Boolean algebra of statements may only be finite.[7] Other axiomatizations have been suggested by various authors to make the theory more rigorous.[7]

4.4.2 Dutch book approach

The Dutch book argument was proposed by de Finetti, and is based on betting. A Dutch book is made when a clever gambler places a set of bets that guarantee a profit, no matter what the outcome of the bets. If a bookmaker follows the rules of the Bayesian calculus in the construction of his odds, a Dutch book cannot be made.

However, Ian Hacking noted that traditional Dutch book arguments did not specify Bayesian updating: they left open the possibility that non-Bayesian updating rules could avoid Dutch books. For example, Hacking writes[17] "And neither the Dutch book argument, nor any other in the personalist arsenal of proofs of the probability axioms, entails the dynamic assumption. Not one entails Bayesianism. So the personalist requires the dynamic assumption to be Bayesian. It is true that in consistency a personalist could abandon the Bayesian model of learning from experience. Salt could lose its savour."

In fact, there are non-Bayesian updating rules that also avoid Dutch books (as discussed in the literature on "probability kinematics" following the publication of Richard C. Jeffreys' rule, which is itself regarded as Bayesian [18]). The additional hypotheses sufficient to (uniquely) specify Bayesian updating are substantial, complicated, and unsatisfactory.[19]

4.4.3 Decision theory approach

A decision-theoretic justification of the use of Bayesian inference (and hence of Bayesian probabilities) was given by Abraham Wald, who proved that every admissible statistical procedure is either a Bayesian procedure or a limit of Bayesian procedures.[20] Conversely, every Bayesian procedure is admissible.[21]

4.5 Personal probabilities and objective methods for constructing priors

Following the work on expected utility theory of Ramsey and von Neumann, decision-theorists have accounted for rational behavior using a probability distribution for the agent. Johann Pfanzagl completed the *Theory of Games and Economic Behavior* by providing an axiomatization of subjective probability and utility, a task left uncompleted by von Neumann and Oskar Morgenstern: their original theory supposed that all the agents had the same probability distribution, as a convenience.[22] Pfanzagl's axiomatization was endorsed by Oskar Morgenstern: "Von Neumann and I have anticipated" the question whether probabilities "might, perhaps more typically, be subjective and have stated specifically that in the latter case axioms could be found from which could derive the desired numerical utility together with a number for the probabilities (cf. p. 19 of The Theory of Games and Economic Behavior). We did not carry this out; it was demonstrated by Pfanzagl ... with all the necessary rigor".[23]

Ramsey and Savage noted that the individual agent's probability distribution could be objectively studied in experiments. The role of judgment and disagreement in science has been recognized since Aristotle and even more clearly with Francis Bacon. The objectivity of science lies not in the psychology of individual scientists, but in the process of science and especially in statistical methods, as noted by C. S. Peirce.[24] Recall that the objective methods for falsifying propositions about personal probabilities have been used for a half century, as noted previously. Procedures for testing hypotheses about probabilities (using finite samples) are due to Ramsey (1931) and de Finetti (1931, 1937, 1964, 1970). Both Bruno de Finetti and Frank P. Ramsey acknowledge their debts to pragmatic philosophy, particularly (for Ramsey) to Charles S. Peirce.

The "Ramsey test" for evaluating probability distributions is implementable in theory, and has kept experimental psychologists occupied for a half century.[25] This work demonstrates that Bayesian-probability propositions can be falsified, and so meet an empirical criterion of Charles S. Peirce, whose work inspired Ramsey. (This falsifiability-criterion was popularized by Karl Popper.[26][27])

Modern work on the experimental evaluation of personal probabilities uses the randomization, blinding, and Boolean-decision procedures of the Peirce-Jastrow experiment.[28] Since individuals act according to different probability judgments, these agents' probabilities are "personal" (but amenable to objective study).

Personal probabilities are problematic for science and for some applications where decision-makers lack the knowledge or time to specify an informed probability-distribution (on which they are prepared to act). To meet the needs of science and of human limitations, Bayesian statisticians have developed "objective" methods for specifying prior probabilities.

Indeed, some Bayesians have argued the prior state of knowledge defines *the* (unique) prior probability-distribution for "regular" statistical problems; cf. well-posed problems. Finding the right method for constructing such "objective"

priors (for appropriate classes of regular problems) has been the quest of statistical theorists from Laplace to John Maynard Keynes, Harold Jeffreys, and Edwin Thompson Jaynes: These theorists and their successors have suggested several methods for constructing "objective" priors:

- Maximum entropy
- Transformation group analysis
- Reference analysis

Each of these methods contributes useful priors for "regular" one-parameter problems, and each prior can handle some challenging statistical models (with "irregularity" or several parameters). Each of these methods has been useful in Bayesian practice. Indeed, methods for constructing "objective" (alternatively, "default" or "ignorance") priors have been developed by avowed subjective (or "personal") Bayesians like James Berger (Duke University) and José-Miguel Bernardo (Universitat de València), simply because such priors are needed for Bayesian practice, particularly in science.[29] The quest for "the universal method for constructing priors" continues to attract statistical theorists.[29]

Thus, the Bayesian statistician needs either to use informed priors (using relevant expertise or previous data) or to choose among the competing methods for constructing "objective" priors.

4.6 Bayesian average

A **Bayesian average** is a method of estimating the mean of a population consistent with Bayesian interpretation, where instead of estimating the mean strictly from any or all available data set, other existing information related to that data set may also be incorporated into the calculation in order to minimize the impact of large deviations, or to assert a default value when the data set is small.

Calculating the Bayesian average uses the prior mean m and a constant C. C is assigned a value that is proportional to the typical data set size. The value is larger when the expected variation between data sets (within the larger population) is small. It is smaller, when the data sets are expected to vary substantially from one another.

$\bar{x} = \frac{Cm+\sum_{i=1}^{n} x_i}{C+n}$ [30]

4.7 See also

- Bertrand's paradox — a paradox in classical probability, solved by E.T. Jaynes in the context of Bayesian probability
- De Finetti's game — a procedure for evaluating someone's subjective probability
- QBism — a controversial application of Bayesian probabilities to quantum mechanics
- Uncertainty
- *An Essay towards solving a Problem in the Doctrine of Chances*

4.8 References

[1] Paulos, John Allen. *The Mathematics of Changing Your Mind,* New York Times (US). August 5, 2011; retrieved 2011-08-06

[2] Jaynes, E.T. "Bayesian Methods: General Background." In Maximum-Entropy and Bayesian Methods in Applied Statistics, by J. H. Justice (ed.). Cambridge: Cambridge Univ. Press, 1986

[3] de Finetti, B. (1974) *Theory of probability* (2 vols.), J. Wiley & Sons, Inc., New York

[4] Stigler, Stephen M. (1986) *The history of statistics.* Harvard University press. pg 131.

[5] Stigler, Stephen M. (1986) *The history of statistics.*, Harvard University press. pp97-98, 131.

[6] Cox, Richard T. *Algebra of Probable Inference*, The Johns Hopkins University Press, 2001

[7] Dupré, Maurice J., Tipler, Frank T. *New Axioms For Bayesian Probability*, Bayesian Analysis (2009), Number 3, pp. 599-606

[8] McGrayne, Sharon Bertsch. (2011). The Theory That Would Not Die, *p. 10.*, p. 10, at Google Books

[9] Stigler, Stephen M. (1986) *The history of statistics.* Harvard University press. Chapter 3.

[10] Fienberg, Stephen. E. (2006) *When did Bayesian Inference become "Bayesian"? Bayesian Analysis*, 1 (1), 1–40. See page 5.

[11] "The works of Wald, *Statistical Decision Functions* (1950) and Savage, *The Foundation of Statistics* (1954) are commonly regarded starting points for current Bayesian approaches"; "Recent developments of the so-called Bayesian approach to statistics" Marshall Dees Harris, *Legal-economic research*, University of Iowa. Agricultural Law Center (1959), p. 125 (fn. 52); p. 126. "This revolution, which may or may not succeed, is neo-Bayesianism. Jeffreys tried to introduce this approach, but did not succeed at the time in giving it general appeal." *Annals of the Computation Laboratory of Harvard University* 31 (1962), p. 180. "It is curious that even in its activities unrelated to ethics, humanity searches for a religion. At the present time, the religion being 'pushed' the hardest is Bayesianism." Oscar Kempthorne, 'The Classical Problem of Inference—Goodness of Fit', *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), p. 235.

[12] Bernardo, J.M. (2005), *Reference analysis*, *Handbook of statistics*, 25, 17–90

[13] Wolpert, R.L. (2004) *A conversation with James O. Berger*, Statistical science, 9, 205–218

[14] Bernardo, José M. (2006) *A Bayesian mathematical statistics primer*. ICOTS-7

[15] Bishop, C.M. *Pattern Recognition and Machine Learning.* Springer, 2007

[16] Halpern, J. *A counterexample to theorems of Cox and Fine*, Journal of Artificial Intelligence Research, 10: 67-85.

[17] Hacking (1967, Section 3, page 316), Hacking (1988, page 124)

[18] "Bayes' Theorem". *stanford.edu*.

[19] van Frassen, B. (1989) *Laws and Symmetry*, Oxford University Press. ISBN 0-19-824860-1

[20] Wald, Abraham. *Statistical Decision Functions.* Wiley 1950.

[21] Bernardo, José M., Smith, Adrian F.M. *Bayesian Theory.* John Wiley 1994. ISBN 0-471-92416-4.

[22] Pfanzagl (1967, 1968)

[23] Morgenstern (1976, page 65)

[24] Stigler, Stephen M. (1978). "Mathematical statistics in the early States". *Annals of Statistics* **6** (March): 239–265 esp. p. 248. doi:10.1214/aos/1176344123. JSTOR 2958876. MR 483118.

[25] Davidson et al. (1957)

[26] "Karl Popper" in *Stanford Encyclopedia of Philosophy*

[27] Popper, Karl. (2002) *The Logic of Scientific Discovery* 2nd Edition, Routledge ISBN 0-415-27843-0 (Reprint of 1959 translation of 1935 original) Page 57.

[28] Peirce & Jastrow (1885)

[29] Bernardo, J. M. (2005). *Reference Analysis*. *Handbook of Statistics* 25 (D. K. Dey and C. R. Rao eds). Amsterdam: Elsevier, 17-90

[30] Yang, Xiao; Zhang, Zhaoxin (2013). "Combining Prestige and Relevance Ranking for Personalized Recommendation". *Proceedings of the 22nd ACM international conference on information & knowledge management (CIKM)*: 1877–1880. doi:10.1145/2505515.2507885.

4.9 Bibliography

- Berger, James O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics (Second ed.). Springer-Verlag. ISBN 0-387-96098-8.
- Bessière, Pierre; Mazer, E., Ahuacatzin, J-M, Mekhnacha, K. (2013). *Bayesian Programming*. CRC Press. ISBN 9781439880326.
- Bernardo, José M.; Smith, Adrian F. M. (1994). *Bayesian Theory*. Wiley. ISBN 0-471-49464-X.
- Bickel, Peter J.; Doksum, Kjell A. (2001). *Mathematical statistics, Volume 1: Basic and selected topics* (Second (updated printing 2007) of the Holden-Day 1976 ed.). Pearson Prentice–Hall. ISBN 0-13-850363-X. MR 443141.
- Davidson, Donald; Suppes, Patrick; Siegel, Sidney (1957). *Decision-Making: An Experimental Approach*. Stanford University Press.
- de Finetti, Bruno. "Probabilism: A Critical Essay on the Theory of Probability and on the Value of Science," (translation of 1931 article) in *Erkenntnis,* volume 31, September 1989.
- de Finetti, Bruno (1937) "La Prévision: ses lois logiques, ses sources subjectives," Annales de l'Institut Henri Poincaré,
- de Finetti, Bruno. "Foresight: its Logical Laws, Its Subjective Sources," (translation of the 1937 article in French) in H. E. Kyburg and H. E. Smokler (eds), *Studies in Subjective Probability,* New York: Wiley, 1964.
- de Finetti, Bruno (1974–5). *Theory of Probability. A Critical Introductory Treatment*, (translation by A.Machi and AFM Smith of 1970 book) 2 volumes. Wiley ISBN 0-471-20141-3, ISBN 0-471-20142-1
- DeGroot, Morris (2004) *Optimal Statistical Decisions*. Wiley Classics Library. (Originally published 1970.) ISBN 0-471-68029-X.
- Hacking, Ian (December 1967). "Slightly More Realistic Personal Probability". *Philosophy of Science* **34** (4): 311–325. doi:10.1086/288169. JSTOR 186120. Partly reprinted in: Gärdenfors, Peter and Sahlin, Nils-Eric. (1988) *Decision, Probability, and Utility: Selected Readings*. 1988. Cambridge University Press. ISBN 0-521-33658-9
- Hajek, A. and Hartmann, S. (2010): "Bayesian Epistemology", in: Dancy, J., Sosa, E., Steup, M. (Eds.) (2001) *A Companion to Epistemology*, Wiley. ISBN 1-4051-3900-5 Preprint
- Hald, Anders (1998). *A History of Mathematical Statistics from 1750 to 1930*. New York: Wiley. ISBN 0-471-17912-4.
- Hartmann, S. and Sprenger, J. (2011) "Bayesian Epistemology", in: Bernecker, S. and Pritchard, D. (Eds.) (2011) *Routledge Companion to Epistemology*. Routledge. ISBN 978-0-415-96219-3 (Preprint)
- Hazewinkel, Michiel, ed. (2001), "Bayesian approach to statistical problems", *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- Howson, C.; Urbach, P. (2005). *Scientific Reasoning: the Bayesian Approach* (3rd ed.). Open Court Publishing Company. ISBN 978-0-8126-9578-6.
- Jaynes E.T. (2003) *Probability Theory: The Logic of Science*, CUP. ISBN 978-0-521-59271-0 (Link to Fragmentary Edition of March 1996).
- McGrayne, SB. (2011). *The Theory That Would Not Die: How Bayes' Rule Cracked The Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of Controversy.* New Haven: Yale University Press. 13-ISBN 9780300169690/10-ISBN 0300169698; OCLC 670481486
- Morgenstern, Oskar (1978). "Some Reflections on Utility". In Andrew Schotter. *Selected Economic Writings of Oskar Morgenstern*. New York University Press. pp. 65–70. ISBN 978-0-8147-7771-8.
- Peirce, C.S. and Jastrow J. (1885). "On Small Differences in Sensation". *Memoirs of the National Academy of Sciences* **3**: 73–83.

- Pfanzagl, J (1967). "Subjective Probability Derived from the Morgenstern-von Neumann Utility Theory". In Martin Shubik. *Essays in Mathematical Economics In Honor of Oskar Morgenstern*. Princeton University Press. pp. 237–251.
- Pfanzagl, J. in cooperation with V. Baumann and H. Huber (1968). "Events, Utility and Subjective Probability". *Theory of Measurement*. Wiley. pp. 195–220.
- Ramsey, Frank Plumpton (1931) "Truth and Probability" (PDF), Chapter VII in *The Foundations of Mathematics and other Logical Essays*, Reprinted 2001, Routledge. ISBN 0-415-22546-9,
- Stigler, SM. (1990). *The History of Statistics: The Measurement of Uncertainty before 1900*. Belknap Press/Harvard University Press. ISBN 0-674-40341-X.
- Stigler, SM. (1999) *Statistics on the Table: The History of Statistical Concepts and Methods*. Harvard University Press. ISBN 0-674-83601-4
- Stone, JV (2013). Download chapter 1 of book "Bayes' Rule: A Tutorial Introduction to Bayesian Analysis", Sebtel Press, England.
- Winkler, RL (2003). *Introduction to Bayesian Inference and Decision* (2nd ed.). Probabilistic. ISBN 0-9647938-4-9. Updated classic textbook. Bayesian theory clearly presented.

Chapter 5

Bayesian programming

Bayesian programming is a formalism and a methodology to specify probabilistic models and solve problems when all the necessary information is not available.

Edwin T. Jaynes proposed that probability could be considered as an alternative and an extension of logic for rational reasoning with incomplete and uncertain information. In his founding book *Probability Theory: The Logic of Science*[1] he developed this theory and proposed what he called "the robot," which was not a physical device, but an inference engine to automate probabilistic reasoning — a kind of Prolog for probability instead of logic. Bayesian Programming[2] is a formal and concrete implementation of this "robot".

Bayesian programming may also be seen as an algebraic formalism to specify graphical models such as, for instance, Bayesian networks, dynamic Bayesian networks, Kalman filters or hidden Markov models. Indeed, Bayesian Programming is more general than Bayesian networks and has a power of expression equivalent to probabilistic factor graphs.

5.1 Formalism

A Bayesian program is a means of specifying a family of probability distributions.

The constituent elements of a Bayesian program are presented below:

$$\text{Program}\begin{cases}\text{Description}\begin{cases}\text{Specification}(\pi)\begin{cases}\text{Variables}\\ \text{Decomposition}\\ \text{Forms}\end{cases}\\ \text{on (based Identification}\delta)\end{cases}\\ \text{Question}\end{cases}$$

1. A program is constructed from a description and a question.
2. A description is constructed using some specification (π) as given by the programmer and an identification or learning process for the parameters not completely specified by the specification, using a data set (δ).
3. A specification is constructed from a set of pertinent variables, a decomposition and a set of forms.
4. Forms are either parametric forms or questions to other Bayesian programs.
5. A question specifies which probability distribution has to be computed.

5.1.1 Description

The purpose of a description is to specify an effective method of computing a joint probability distribution on a set of variables $\{X_1, X_2, \cdots, X_N\}$ given a set of experimental data δ and some specification π . This joint distribution is denoted as: $P\left(X_1 \wedge X_2 \wedge \cdots \wedge X_N \mid \delta \wedge \pi\right)$.

To specify preliminary knowledge π , the programmer must undertake the following:

1. Define the set of relevant variables $\{X_1, X_2, \cdots, X_N\}$ on which the joint distribution is defined.
2. Decompose the joint distribution (break it into relevant independent or conditional probabilities).
3. Define the forms each of the distributions (e.g., for each variable, one of the list of probability distributions).

Decomposition

Given a partition $\{X_1, X_2, \ldots, X_N\}$ containing K subsets, K variables are defined $L_1, \cdots, L_K$, each corresponding to one of these subsets. Each variable L_k is obtained as the conjunction of the variables $\{X_{k_1}, X_{k_2}, \cdots\}$ belonging to the k^{th} subset. Recursive application of Bayes' theorem leads to:

$$\begin{aligned} & P(X_1 \wedge X_2 \wedge \cdots \wedge X_N \mid \delta \wedge \pi) \\ = & P(L_1 \wedge \cdots \wedge L_K \mid \delta \wedge \pi) \\ = & P(L_1 \mid \delta \wedge \pi) \times P(L_2 \mid L_1 \wedge \delta \wedge \pi) \times \cdots \times P(L_K \mid L_{K-1} \wedge \cdots \wedge L_1 \wedge \delta \wedge \pi) \end{aligned}$$

Conditional independence hypotheses then allow further simplifications. A conditional independence hypothesis for variable L_k is defined by choosing some variable X_n among the variables appearing in the conjunction $L_{k-1} \wedge \cdots \wedge L_2 \wedge L_1$, labelling R_k as the conjunction of these chosen variables and setting:

$$P(L_k \mid L_{k-1} \wedge \cdots \wedge L_1 \wedge \delta \wedge \pi) = P(L_k \mid R_k \wedge \delta \wedge \pi)$$

We then obtain:

$$\begin{aligned} & P(X_1 \wedge X_2 \wedge \cdots \wedge X_N \mid \delta \wedge \pi) \\ = & P(L_1 \mid \delta \wedge \pi) \times P(L_2 \mid R_2 \wedge \delta \wedge \pi) \times \cdots \times P(L_K \mid R_K \wedge \delta \wedge \pi) \end{aligned}$$

Such a simplification of the joint distribution as a product of simpler distributions is called a decomposition, derived using the chain rule.

This ensures that each variable appears at the most once on the left of a conditioning bar, which is the necessary and sufficient condition to write mathematically valid decompositions.

Forms

Each distribution $P(L_k \mid R_k \wedge \delta \wedge \pi)$ appearing in the product is then associated with either a parametric form (i.e., a function $f_\mu(L_k)$) or a question to another Bayesian program $P(L_k \mid R_k \wedge \delta \wedge \pi) = P\left(L \mid R \wedge \widehat{\delta} \wedge \widehat{\pi}\right)$.

When it is a form $f_\mu(L_k)$, in general, μ is a vector of parameters that may depend on R_k or δ or both. Learning takes place when some of these parameters are computed using the data set δ .

An important feature of Bayesian Programming is this capacity to use questions to other Bayesian programs as components of the definition of a new Bayesian program. $P(L_k \mid R_k \wedge \delta \wedge \pi)$ is obtained by some inferences done by another Bayesian program defined by the specifications $\widehat{\pi}$ and the data $\widehat{\delta}$. This is similar to calling a subroutine in classical programming and provides an easy way to build hierarchical models.

5.1.2 Question

Given a description (i.e., $P(X_1 \wedge X_2 \wedge \cdots \wedge X_N \mid \delta \wedge \pi)$), a question is obtained by partitioning $\{X_1, X_2, \cdots, X_N\}$ into three sets: the searched variables, the known variables and the free variables.

The 3 variables $Searched$, $Known$ and $Free$ are defined as the conjunction of the variables belonging to these sets.

A question is defined as the set of distributions:

$P\left(Searched \mid \text{Known} \wedge \delta \wedge \pi\right)$

made of many "instantiated questions" as the cardinal of $Known$, each instantiated question being the distribution:

$P\left(\text{Searched} \mid \text{Known} \wedge \delta \wedge \pi\right)$

5.1.3 Inference

Given the joint distribution $P\left(X_1 \wedge X_2 \wedge \cdots \wedge X_N \mid \delta \wedge \pi\right)$, it is always possible to compute any possible question using the following general inference:

$$
\begin{aligned}
& P\left(\text{Searched} \mid \text{Known} \wedge \delta \wedge \pi\right) \\
= & \sum_{\text{Free}} \left[P\left(\text{Searched} \wedge \text{Free} \mid \text{Known} \wedge \delta \wedge \pi\right)\right] \\
= & \frac{\sum_{\text{Free}} \left[P\left(\text{Searched} \wedge \text{Free} \wedge \text{Known} \mid \delta \wedge \pi\right)\right]}{P\left(\text{Known} \mid \delta \wedge \pi\right)} \\
= & \frac{\sum_{\text{Free}} \left[P\left(\text{Searched} \wedge \text{Free} \wedge \text{Known} \mid \delta \wedge \pi\right)\right]}{\sum_{\text{Free} \wedge \text{Searched}} \left[P\left(\text{Searched} \wedge \text{Free} \wedge \text{Known} \mid \delta \wedge \pi\right)\right]} \\
= & \frac{1}{Z} \times \sum_{\text{Free}} \left[P\left(\text{Searched} \wedge \text{Free} \wedge \text{Known} \mid \delta \wedge \pi\right)\right]
\end{aligned}
$$

where the first equality results from the marginalization rule, the second results from Bayes' theorem and the third corresponds to a second application of marginalization. The denominator appears to be a normalization term and can be replaced by a constant Z .

Theoretically, this allows to solve any Bayesian inference problem. In practice, however, the cost of computing exhaustively and exactly $P\left(\text{Searched} \mid \text{Known} \wedge \delta \wedge \pi\right)$ is too great in almost all cases.

Replacing the joint distribution by its decomposition we get:

$$
\begin{aligned}
& P\left(\text{Searched} \mid \text{Known} \wedge \delta \wedge \pi\right) \\
= & \frac{1}{Z} \sum_{\text{Free}} \left[\prod_{k=1}^{K} \left[P\left(L_i \mid K_i \wedge \pi\right)\right]\right]
\end{aligned}
$$

which is usually a much simpler expression to compute, as the dimensionality of the problem is considerably reduced by the decomposition into a product of lower dimension distributions.

5.2 Example

5.2.1 Bayesian spam detection

The purpose of Bayesian spam filtering is to eliminate junk e-mails.

The problem is very easy to formulate. E-mails should be classified into one of two categories: non-spam or spam. The only available information to classify the e-mails is their content: a set of words. Using these words without taking the order into account is commonly called a bag of words model.

The classifier should furthermore be able to adapt to its user and to learn from experience. Starting from an initial standard setting, the classifier should modify its internal parameters when the user disagrees with its own decision. It will hence adapt to the user's criteria to differentiate between non-spam and spam. It will improve its results as it encounters increasingly classified e-mails.

Variables

The variables necessary to write this program are as follows:

1. $Spam$: a binary variable, false if the e-mail is not spam and true otherwise.
2. $W_0, W_1, \ldots, W_{N-1}$: N binary variables. W_n is true if the n^{th} word of the dictionary is present in the text.

These $N + 1$ binary variables sum up all the information about an e-mail.

Decomposition

Starting from the joint distribution and applying recursively Bayes' theorem we obtain:

$$\begin{aligned} & P(\text{Spam} \wedge W_0 \wedge \cdots \wedge W_{N-1}) \\ = & P(\text{Spam}) \times P(W_0 \mid \text{Spam}) \times P(W_1 \mid \text{Spam} \wedge W_0) \\ & \times \cdots \\ & \times P\left(W_{N-1} \mid \text{Spam} \wedge W_0 \wedge \cdots \wedge W_{N-2}\right) \end{aligned}$$

This is an exact mathematical expression.

It can be drastically simplified by assuming that the probability of appearance of a word knowing the nature of the text (spam or not) is independent of the appearance of the other words. This is the naive Bayes assumption and this makes this spam filter a naive Bayes model.

For instance, the programmer can assume that:

$$P(W_1 \mid \text{Spam} \wedge W_0) = P(W_1 \mid \text{Spam})$$

to finally obtain:

$$P(\text{Spam} \wedge W_0 \wedge \ldots \wedge W_{N-1}) = P(\text{Spam}) \prod_{n=0}^{N-1} [P(W_n \mid \text{Spam})]$$

This kind of assumption is known as the naive Bayes' assumption. It is "naive" in the sense that the independence between words is clearly not completely true. For instance, it completely neglects that the appearance of pairs of words may be more significant than isolated appearances. However, the programmer may assume this hypothesis and may develop the model and the associated inferences to test how reliable and efficient it is.

Parametric forms

To be able to compute the joint distribution, the programmer must now specify the $N + 1$ distributions appearing in the decomposition:

1. $P(\text{Spam})$ is a prior defined, for instance, by $P([\text{Spam} = 1]) = 0.75$
2. Each of the N forms $P(W_n \mid \text{Spam})$ may be specified using Laplace rule of succession (this is a pseudocounts-based smoothing technique to counter the zero-frequency problem of words never-seen-before):
   (a) $P(W_n \mid [\text{Spam} = \text{false}]) = \frac{1+a_f^n}{2+a_f}$
   (b) $P(W_n \mid [\text{Spam} = \text{true}]) = \frac{1+a_t^n}{2+a_t}$

where a_f^n stands for the number of appearances of the n^{th} word in non-spam e-mails and a_f stands for the total number of non-spam e-mails. Similarly, a_t^n stands for the number of appearances of the n^{th} word in spam e-mails and a_t stands for the total number of spam e-mails.

Identification

The N forms $P(W_n \mid \text{Spam})$ are not yet completely specified because the $2N+2$ parameters $a_f^{n=0,\ldots,N-1}$, $a_t^{n=0,\ldots,N-1}$, a_f and a_t have no values yet.

The identification of these parameters could be done either by batch processing a series of classified e-mails or by an incremental updating of the parameters using the user's classifications of the e-mails as they arrive.

Both methods could be combined: the system could start with initial standard values of these parameters issued from a generic database, then some incremental learning customizes the classifier to each individual user.

Question

The question asked to the program is: "what is the probability for a given text to be spam knowing which words appear and don't appear in this text?" It can be formalized by:

$$P(\text{Spam} \mid w_0 \wedge \cdots \wedge w_{N-1})$$

which can be computed as follows:

$$\begin{aligned} & P(\text{Spam} \mid w_0 \wedge \cdots \wedge w_{N-1}) \\ = & \frac{P(\text{Spam}) \prod_{n=0}^{N-1} [P(w_n \mid \text{Spam})]}{\sum_{\text{Spam}} [P(\text{Spam}) \prod_{n=0}^{N-1} [P(w_n \mid \text{Spam})]]} \end{aligned}$$

The denominator appears to be a normalization constant. It is not necessary to compute it to decide if we are dealing with spam. For instance, an easy trick is to compute the ratio:

$$\begin{aligned} & \frac{P([\text{Spam} = \text{true}] \mid w_0 \wedge \cdots \wedge w_{N-1})}{P([\text{Spam} = \text{false}] \mid w_0 \wedge \cdots \wedge w_{N-1})} \\ = & \frac{P([\text{Spam} = \text{true}])}{P([\text{Spam} = \text{false}])} \times \prod_{n=0}^{N-1} \left[\frac{P(w_n \mid [\text{Spam} = \text{true}])}{P(w_n \mid [\text{Spam} = \text{false}])} \right] \end{aligned}$$

This computation is faster and easier because it requires only $2N$ products.

Bayesian program

The Bayesian spam filter program is completely defined by:

$$\text{Pr} \begin{cases} Ds \begin{cases} Sp(\pi) \begin{cases} Va : \text{Spam}, W_0, W_1 \ldots W_{N-1} \\ Dc : \begin{cases} P(\text{Spam} \wedge W_0 \wedge \ldots \wedge W_n \wedge \ldots \wedge W_{N-1}) \\ = P(\text{Spam}) \prod_{n=0}^{N-1} P(W_n \mid \text{Spam}) \end{cases} \\ Fo : \begin{cases} P(\text{Spam}) : \begin{cases} P([\text{Spam} = \text{false}]) = 0.25 \\ P([\text{Spam} = \text{true}]) = 0.75 \end{cases} \\ P(W_n \mid \text{Spam}) : \begin{cases} P(W_n \mid [\text{Spam} = \text{false}]) \\ = \frac{1+a_f^n}{2+a_f} \\ P(W_n \mid [\text{Spam} = \text{true}]) \\ = \frac{1+a_t^n}{2+a_t} \end{cases} \end{cases} \end{cases} \\ \text{on (based Identification} \delta) \end{cases} \\ Qu : P(\text{Spam} \mid w_0 \wedge \ldots \wedge w_n \wedge \ldots \wedge w_{N-1}) \end{cases}$$

5.2.2 Bayesian filter, Kalman filter and hidden Markov model

Bayesian filters (often called Recursive Bayesian estimation) are generic probabilistic models for time evolving processes. Numerous models are particular instances of this generic approach, for instance: the Kalman filter or the Hidden Markov model.

Variables

- Variables $S^0, \ldots, S^T$ are a time series of state variables considered to be on a time horizon ranging from 0 to T .
- Variables $O^0, \ldots, O^T$ are a time series of observation variables on the same horizon.

Decomposition

The decomposition is based:

- on $P(S^t \mid S^{t-1})$, called the system model, transition model or dynamic model, which formalizes the transition from the state at time $t-1$ to the state at time t ;
- on $P(O^t \mid S^t)$, called the observation model, which expresses what can be observed at time t when the system is in state S^t ;
- on an initial state at time 0 : $P(S^0 \wedge O^0)$.

Parametrical forms

The parametrical forms are not constrained and different choices lead to different well-known models: see Kalman filters and Hidden Markov models just below.

Question

The question usually asked of these models is $P\left(S^{t+k} \mid O^0 \wedge \cdots \wedge O^t\right)$: what is the probability distribution for the state at time $t+k$ knowing the observations from instant 0 to t ?

The most common case is Bayesian filtering where $k = 0$, which means that one searches for the present state, knowing the past observations.

However it is also possible to do a prediction $(k > 0)$, where one tries to extrapolate a future state from past observations, or to do smoothing $(k < 0)$, where one tries to recover a past state from observations made either before or after that instant.

Some more complicated questions may also be asked as shown below in the HMM section.

Bayesian filters $(k = 0)$ have a very interesting recursive property, which contributes greatly to their attractiveness. $P\left(S^t|O^0 \wedge \cdots \wedge O^t\right)$ may be computed simply from $P\left(S^{t1} \mid O^0 \wedge \cdots \wedge O^{t-1}\right)$ with the following formula:

$$\begin{aligned} & P\left(S^t|O^0 \wedge \cdots \wedge O^t\right) \\ = \; & P\left(O^t|S^t\right) \times \textstyle\sum_{S^{t-1}} \left[P\left(S^t|S^{t-1}\right) \times P\left(S^{t-1}|O^0 \wedge \cdots \wedge O^{t-1}\right)\right] \end{aligned}$$

Another interesting point of view for this equation is to consider that there are two phases: a prediction phase and an estimation phase:

- During the prediction phase, the state is predicted using the dynamic model and the estimation of the state at the previous moment:

$$\begin{aligned} & P\left(S^t|O^0 \wedge \cdots \wedge O^{t-1}\right) \\ = \; & \textstyle\sum_{S^{t-1}} \left[P\left(S^t|S^{t-1}\right) \times P\left(S^{t-1}|O^0 \wedge \cdots \wedge O^{t-1}\right)\right] \end{aligned}$$

- During the estimation phase, the prediction is either confirmed or invalidated using the last observation:

$$\begin{aligned} & P\left(S^t \mid O^0 \wedge \cdots \wedge O^t\right) \\ = & P\left(O^t \mid S^t\right) \times P\left(S^t | O^0 \wedge \cdots \wedge O^{t-1}\right) \end{aligned}$$

Bayesian program

$$Pr \begin{cases} Ds \begin{cases} Sp(\pi) \begin{cases} Va: \\ S^0, \cdots, S^T, O^0, \cdots, O^T \\ Dc: \\ \begin{cases} & P\left(S^0 \wedge \cdots \wedge S^T \wedge O^0 \wedge \cdots \wedge O^T | \pi\right) \\ = & P\left(S^0 \wedge O^0\right) \times \prod_{t=1}^{T}\left[P\left(S^t | S^{t-1}\right) \times P\left(O^t | S^t\right)\right] \end{cases} \\ Fo: \\ \begin{cases} P\left(S^0 \wedge O^0\right) \\ P\left(S^t | S^{t-1}\right) \\ P\left(O^t | S^t\right) \end{cases} \end{cases} \\ Id \end{cases} \\ Qu: \\ \begin{cases} P\left(S^{t+k} | O^0 \wedge \cdots \wedge O^t\right) \\ (k=0) \equiv \text{Filtering} \\ (k>0) \equiv \text{Prediction} \\ (k<0) \equiv \text{Smoothing} \end{cases} \end{cases}$$

Kalman filter

The very well-known Kalman filters[3] are a special case of Bayesian filters.

They are defined by the following Bayesian program:

$$Pr \begin{cases} Ds \begin{cases} Sp(\pi) \begin{cases} Va: \\ S^0, \cdots, S^T, O^0, \cdots, O^T \\ Dc: \\ \begin{cases} & P\left(S^0 \wedge \cdots \wedge O^T | \pi\right) \\ = & \begin{bmatrix} P\left(S^0 \wedge O^0 | \pi\right) \\ \prod_{t=1}^{T}\left[P\left(S^t | S^{t-1} \wedge \pi\right) \times P\left(O^t | S^t \wedge \pi\right)\right] \end{bmatrix} \end{cases} \\ Fo: \\ \begin{cases} P\left(S^t \mid S^{t-1} \wedge \pi\right) \equiv G\left(S^t, A \bullet S^{t-1}, Q\right) \\ P\left(O^t \mid S^t \wedge \pi\right) \equiv G\left(O^t, H \bullet S^t, R\right) \end{cases} \end{cases} \\ Id \end{cases} \\ Qu: \\ P\left(S^T \mid O^0 \wedge \cdots \wedge O^T \wedge \pi\right) \end{cases}$$

- Variables are continuous.
- The transition model $P(S^t \mid S^{t-1} \wedge \pi)$ and the observation model $P(O^t \mid S^t \wedge \pi)$ are both specified using Gaussian laws with means that are linear functions of the conditioning variables.

With these hypotheses and by using the recursive formula, it is possible to solve the inference problem analytically to answer the usual $P(S^T \mid O^0 \wedge \cdots \wedge O^T \wedge \pi)$ question. This leads to an extremely efficient algorithm, which explains the popularity of Kalman filters and the number of their everyday applications.

When there are no obvious linear transition and observation models, it is still often possible, using a first-order Taylor's expansion, to treat these models as locally linear. This generalization is commonly called the extended Kalman filter.

Hidden Markov model

Hidden Markov models (HMMs) are another very popular specialization of Bayesian filters.

They are defined by the following Bayesian program:

$$\Pr \begin{cases} Ds \begin{cases} Sp(\pi) \begin{cases} Va: \\ S^0, \ldots, S^T, O^0, \ldots, O^T \\ Dc: \\ \begin{cases} \quad P\left(S^0 \wedge \cdots \wedge O^T \mid \pi\right) \\ = \quad \left[\begin{array}{c} P\left(S^0 \wedge O^0 \mid \pi\right) \\ \prod_{t=1}^{T} \left[P\left(S^t \mid S^{t-1} \wedge \pi\right) \times P\left(O^t \mid S^t \wedge \pi\right)\right] \end{array} \right] \end{cases} \\ Fo: \\ \begin{cases} P\left(S^0 \wedge O^0 \mid \pi\right) \equiv \text{Matrix} \\ P\left(S^t \mid S^{t-1} \wedge \pi\right) \equiv \text{Matrix} \\ P\left(O^t \mid S^t \wedge \pi\right) \equiv \text{Matrix} \end{cases} \end{cases} \\ Id \end{cases} \\ Qu: \\ \max_{S^1 \wedge \cdots \wedge S^{T-1}} \left[P\left(S^1 \wedge \cdots \wedge S^{T-1} \mid S^T \wedge O^0 \wedge \cdots \wedge O^T \wedge \pi\right)\right] \end{cases}$$

- Variables are treated as being discrete.
- The transition model $P\left(S^t \mid S^{t-1} \wedge \pi\right)$ and the observation model $P\left(O^t \mid S^t \wedge \pi\right)$ are

both specified using probability matrices.

- The question most frequently asked of HMMs is:

$$\max_{S^1 \wedge \cdots \wedge S^{T-1}} \left[P\left(S^1 \wedge \cdots \wedge S^{T-1} \mid S^T \wedge O^0 \wedge \cdots \wedge O^T \wedge \pi\right)\right]$$

What is the most probable series of states that leads to the present state, knowing the past observations?

This particular question may be answered with a specific and very efficient algorithm called the Viterbi algorithm.

A specific learning algorithm called the Baum–Welch algorithm has also been developed for HMMs.

5.3 Applications

5.3.1 Academic applications

For the last 15 years, Bayesian programming approach has been used in various universities to develop both robotics applications and life sciences models.[4]

Robotics

In robotics, Bayesian programming has been applied to autonomous robotics,[5][6][7][8][9] robotic CAD systems,[10] Advanced driver assistance systems,[11] robotic arm control, mobile robotics,[12][13] Human-robots interactions,[14] Human-vehicle interactions (Bayesian autonomous driver models) [15] [16] [17] [18] [19] [20] video game avatar programming and training [21] and real-time strategy games (AI).[22]

Life sciences

In life sciences, Bayesian Programming has been used in vision to reconstruct shape from motion,[23] to model visuo-vestibular interaction[24] and to study saccadic eye movements;[25] in speech perception and control to study early acquisition of speech[26] and the emergence of articulatory-acoustic systems;[27] and to model handwriting perception and control.[28]

5.4 Bayesian programming versus possibility theories

The comparison between probabilistic approaches (not only Bayesian programming) and possibility theories has been debated for a long time and is, unfortunately, a very controversial matter.

Possibility theories like, for instance, fuzzy sets,[29] Fuzzy logic[30] and Possibility theory[31] propose different alternatives to probability to model uncertainty. They argue that probability is insufficient or inconvenient to model certain aspects of incomplete and uncertain knowledge.

The defense of probability is mainly based on Cox's theorem which, starting from four postulates concerning rational reasoning in the presence of uncertainty, demonstrates that the only mathematical framework that satisfies these postulates is probability theory. The argument then goes like this: if you use a different approach than probability, then you necessarily infringe on one of these postulates. Let us see which one and discuss its utility.

5.5 Bayesian programming versus probabilistic programming

The purpose of probabilistic programming is to unify the scope of classical programming languages with probabilistic modeling (especially Bayesian networks) in order to be able to deal with uncertainty but still profit from the power of expression of programming languages to describe complex models.

The extended classical programming languages can be logical languages as proposed in Probabilistic Horn Abduction,[32] Independent Choice Logic,[33] PRISM,[34] and ProbLog which propose an extension of Prolog.

It can also be extensions of functional programming languages (essentially Lisp and Scheme) such as IBAL or CHURCH. The inspiring programming languages can even be object oriented like in BLOG and FACTORIE or more standard ones like in CES and FIGARO.

The purpose of Bayesian programming is different. Jaynes' precept of "probability as logic" defends that probability is an extension of and an alternative to logic above which a complete theory of rationality, computation and programming can be rebuilt. Bayesian programming does not search to extend classical languages but rather to replace them by a new programming approach based on probability and taking fully into account incompleteness and uncertainty.

The precise comparison between the semantic and power of expression of Bayesian and probabilistic programming is still an open question.

5.6 See also

- Bayes' rule
- Bayesian inference
- Bayesian probability
- Bayesian spam filtering
- Belief propagation
- Cox's theorem
- Expectation-maximization algorithm
- Factor graph

- Graphical model
- Hidden Markov model
- Judea Pearl
- Kalman filter
- Naive Bayes classifier
- Pierre-Simon de Laplace
- Probabilistic logic
- Probabilistic programming language
- Subjective logic

5.7 References

[1] Jaynes, Edwin T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press. ISBN 0-521-59271-2.

[2] Bessière, P.; Mazer, E.; Ahuactzin, J-M. & Mekhnacha, K. (2013). *Bayesian Programming*. Chapman & Hall/CRC. ISBN 9781439880326.

[3] Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". *Transactions of the ASME--Journal of Basic Engineering* **82**: 33—45. doi:10.1115/1.3662552.

[4] Bessière, P.; Laugier, C. & Siegwart, R. (2008). *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*. Springer. ISBN 978-3-540-79007-5.

[5] Lebeltel, O.; Bessière, P.; Diard, J. & Mazer, E. (2004). "Bayesian Robot Programming". *Advanced Robotics* **16** (1): 49—79. doi:10.1023/b:auro.0000008671.38949.43.

[6] Diard, J.; Gilet, E.; Simonin, E. & Bessière, P. (2010). "Incremental learning of Bayesian sensorimotor models: from low-level behaviours to large-scale structure of the environment". *Connection Science* **22** (4): 291—312. doi:10.1080/09540091003682561.

[7] Pradalier, C.; Hermosillo, J.; Koike, C., Braillon, C.; Bessière, P. & Laugier, C. (2005). "The CyCab: a car-like robot navigating autonomously and safely among pedestrians". *Robotics and Autonomous Systems* **50** (1): 51—68. doi:10.1016/j.robot.2004.10.002.

[8] Ferreira, J.; Lobo, J.; Bessière, P.; Castelo-Branco, M. & Dias, J. (2012). "A Bayesian Framework for Active Artificial Perception". *IEEE Transactions on Systems, IEEE Transactions on Systems, Man, and Cybernetics, Part B* **99**: 1—13.

[9] Ferreira, J. F.; Dias, J. M. (2014). *Probabilistic Approaches to Robotic Perception*. Springer.

[10] Mekhnacha, K.; Mazer, E. & Bessière, P. (2001). "The design and implementation of a Bayesian CAD modeler for robotic applications". *Advanced Robotics* **15** (1): 45—69. doi:10.1163/156855301750095578.

[11] Coué, C.; Pradalier, C.; Laugier, C.; Fraichard, T. & Bessière, P. (2006). "Bayesian Occupancy Filtering for Multitarget Tracking: an Automotive Application". *International Journal of Robotics Research* **25** (1): 19—30. doi:10.1177/0278364906061158.

[12] Vasudevan, S.; Siegwart, R. (2008). "Bayesian space conceptualization and place classification for semantic maps in Bayesian space conceptualization and place classification for semantic maps in mobile robotics". *Robotics and Autonomous Systems* **56** (6): 522—537. doi:10.1016/j.robot.2008.03.005.

[13] Perrin, X.; Chavarriaga, R.; Colas, F.; Seigwart, R. & Millan, J. (2010). "Brain-coupled interaction for semi-autonomous navigation of an assistive robot". *Robotics and Autonomous Systems* **58** (12): 1246—1255. doi:10.1016/j.robot.2010.05.010.

[14] Rett, J.; Dias, J. & Ahuactzin, J-M. (2010). "Bayesian reasoning for Laban Movement Analysis used in human-machine interaction". *Int. J. of Reasoning-based Intelligent Systems* **2** (1): 13—35. doi:10.1504/IJRIS.2010.029812.

[15] Möbus, C.; Eilers, M.; Garbe, H.; Zilinski, M. (2009), http://link.springer.com/chapter/10.1007%2F978-3-642-02809-0_45 |contribution-url= missing title (help), in Duffy, Vincent G., *Probabilistic and Empirical Grounded Modeling of Agents in (Partial) Cooperative Traffic Scenarios*, Lecture Notes in Computer Science, Volume 5620, Second International Conference, ICDHM 2009, San Diego, CA, USA: Springer, pp. 423–432, doi:10.1007/978-3-642-02809-0_45, ISBN 978-3-642-02808-3

[16] Möbus, C.; Eilers, M. (2009), http://link.springer.com/chapter/10.1007%2F978-3-642-02809-0_44 |contribution-url= missing title (help), in Duffy, Vincent G., *Further Steps Towards Driver Modeling according to the Bayesian Programming Approach*, Lecture Notes in Computer Science, Volume 5620, Second International Conference, ICDHM 2009, San Diego, CA, USA: Springer, pp. 413–422, doi:10.1007/978-3-642-02809-0_44, ISBN 978-3-642-02808-3

[17] Eilers, M.; Möbus, C. (2010). "Lernen eines modularen Bayesian Autonomous Driver Mixture-of-Behaviors (BAD MoB) Modells". In Kolrep, H.; Jürgensohn, Th. *Fahrermodellierung - Zwischen kinematischen Menschmodellen und dynamisch-kognitiven Verhaltensmodellen*. Fortschrittsbericht des VDI in der Reihe 22 (Mensch-Maschine-Systeme). Düsseldorf, Germany: VDI-Verlag. pp. 61 – 74. ISBN 978-3-18-303222-8.

[18] Möbus, C.; Eilers, M. (2011). http://www.igi-global.com/chapter/prototyping-smart-assistance-bayesian-autonomous/54671 |contribution-url= missing title (help). In Mastrogiovanni, F.; Chong, N.-Y. *Prototyping Smart Assistance with Bayesian Autonomous Driver Models*. Hershey, Pennsylvania (USA): IGI Global publications. pp. 460–512. doi:10.4018/978-1-61692-857-5.ch023. ISBN 9781616928575.

[19] Eilers, M.; Möbus, C. (2011). "Learning the Relevant Percepts of Modular Hierarchical Bayesian Driver Models Using a Bayesian Information Criterion". In Duffy, V.G. *Digital Human Modeling*. LNCS 6777. Heidelberg, Germany: Springer. pp. 463–472. doi:10.1007/978-3-642-21799-9_52. ISBN 978-3-642-21798-2.

[20] Eilers, M.; Möbus, C. (2011). "Learning of a Bayesian Autonomous Driver Mixture-of-Behaviors (BAD-MoB) Model". In Duffy, V.G. *Advances in Applied Digital Human Modeling*. LNCS 6777. Boca Raton, USA: CRC Press, Taylor & Francis Group. pp. 436–445. ISBN 978-1-4398-3511-1.

[21] Le Hy, R.; Arrigoni, A.; Bessière, P. & Lebetel, O. (2004). "Teaching Bayesian Behaviours to Video Game Characters". *Robotics and Autonomous Systems* **47** (2–3): 177—185. doi:10.1016/j.robot.2004.03.012.

[22] Synnaeve, G. (2012). *Bayesian Programming and Learning for Multiplayer Video Games* (PDF).

[23] Colas, F.; Droulez, J.; Wexler, M. & Bessière, P. (2008). "A unified probabilistic model of the perception of three-dimensional structure from optic flow". *Biological Cybernetics*: 132—154.

[24] Laurens, J.; Droulez, J. (2007). "Bayesian processing of vestibular information". *Biological Cybernetics* **96** (4): 389—404. doi:10.1007/s00422-006-0133-1.

[25] Colas, F.; Flacher, F.; Tanner, T.; Bessière, P. & Girard, B. (2009). "Bayesian models of eye movement selection with retinotopic maps". *Biological Cybernetics* **100** (3): 203—214. doi:10.1007/s00422-009-0292-y.

[26] Serkhane, J.; Schwartz, J-L. & Bessière, P. (2005). "Building a talking baby robot A contribution to the study of speech acquisition and evolution". *Interaction Studies* **6** (2): 253—286. doi:10.1075/is.6.2.06ser.

[27] Moulin-Frier, C.; Laurent, R.; Bessière, P.; Schwartz, J-L. & Diard, J. (2012). "Adverse conditions improve distinguishability of auditory, motor and percep-tuo-motor theories of speech perception: an exploratory Bayesian modeling study". *Language and Cognitive Processes* **27** (7–8): 1240—1263. doi:10.1080/01690965.2011.645313.

[28] Gilet, E.; Diard, J. & Bessière, P. (2011). Sporns, Olaf, ed. "Bayesian Action–Perception Computational Model: Interaction of Production and Recognition of Cursive Letters". *Plos ONE* **6** (6): e20387. Bibcode:2011PLoSO...620387G. doi:10.1371/journal.pone.0020387.

[29] Zadeh, Lofti, A. (1965). "Fuzzy sets". *Information and Control* **8** (3): 338—353. doi:10.1016/S0019-9958(65)90241-X.

[30] Zadeh, Lofti, A. (1975). "Fuzzy logic and approximate reasoning". *Synthese* **30** (3—4): 407—428. doi:10.1007/BF00485052.

[31] Dubois, D.; Prade, H. (2001). *Ann. Math. Artif. Intell.* **32** (1—4): 35—66. doi:10.1023/A:1016740830286. Missing or empty |title= (help)

[32] Poole, D. (1993). "Probabilistic Horn abduction and Bayesian networks". *Artificial Intelligence* **64**: 81–129. doi:10.1016/0004-3702(93)90061-F.

[33] Poole, D. (1997). "The Independent Choice Logic for modelling multiple agents under uncertainty". *Artficial Intelligence* **94**: 7–56. doi:10.1016/S0004-3702(97)00027-1.

[34] Sato, T.; Kameya, Y. (2001). *Journal of Artificial Intelligence Research* **15**: 391—454. Missing or empty |title= (help)

5.8 Further reading

- Kamel Mekhnacha (2013). *Bayesian Programming*. Chapman and Hall/CRC. ISBN 978-1-4398-8032-6.

5.9 External links

- A companion site to the *Bayesian programming* book where to download ProBT an inference engine dedicated to Bayesian programming.
- The Bayesian-programming.org site for the promotion of Bayesian programming with detailed information and numerous publications.

Chapter 6

Belief propagation

Belief propagation, also known as **sum-product message passing** is a message passing algorithm for performing inference on graphical models, such as Bayesian networks and Markov random fields. It calculates the marginal distribution for each unobserved node, conditional on any observed nodes. Belief propagation is commonly used in artificial intelligence and information theory and has demonstrated empirical success in numerous applications including low-density parity-check codes, turbo codes, free energy approximation, and satisfiability.[1]

The algorithm was first proposed by Judea Pearl in 1982,[2] who formulated this algorithm on trees, and was later extended to polytrees.[3] It has since been shown to be a useful approximate algorithm on general graphs.[4]

If X={Xi} is a set of discrete random variables with a joint mass function p, the marginal distribution of a single Xi is simply the summation of p over all other variables:

$$p_{X_i}(x_i) = \sum_{\mathbf{x}' : x'_i = x_i} p(\mathbf{x}').$$

However, this quickly becomes computationally prohibitive: if there are 100 binary variables, then one needs to sum over $2^{99} \approx 6.338 \times 10^{29}$ possible values. By exploiting the polytree structure, belief propagation allows the marginals to be computed much more efficiently.

6.1 Description of the sum-product algorithm

Variants of the belief propagation algorithm exist for several types of graphical models (Bayesian networks and Markov random fields,[5] in particular). We describe here the variant that operates on a factor graph. A factor graph is a bipartite graph containing nodes corresponding to variables V and factors F, with edges between variables and the factors in which they appear. We can write the joint mass function:

$$p(\mathbf{x}) = \prod_{a \in F} f_a(\mathbf{x}_a)$$

where $\mathbf{x}a$ is the vector of neighbouring variable nodes to the factor node a. Any Bayesian network or Markov random field can be represented as a factor graph.

The algorithm works by passing real valued functions called *messages* along the edges between the hidden nodes. More precisely, if v is a variable node and a is a factor node connected to v in the factor graph, the messages from v to a, (denoted by $\mu_{v \to a}$) and from a to v ($\mu_{a \to v}$), are real-valued functions whose domain is Dom(v), the set of values that can be taken by the random variable associated with v. These messages contain the "influence" that one variable exerts on another. The messages are computed differently depending on whether the node receiving the message is a variable node or a factor node. Keeping the same notation:

- A message from a variable node v to a factor node a is the product of the messages from all other neighbouring factor nodes (except the recipient; alternatively one can say the recipient sends as message the constant function equal to "1"):

$$\forall x_v \in Dom(v),\ \mu_{v\to a}(x_v) = \prod_{a^* \in N(v)\setminus\{a\}} \mu_{a^*\to v}(x_v).$$

where $N(v)$ is the set of neighbouring (factor) nodes to v. If $N(v) \setminus \{a\}$ is empty, then $\mu_{v\to a}(x_v)$ is set to the uniform distribution.

- A message from a factor node a to a variable node v is the product of the factor with messages from all other nodes, marginalised over all variables except the one associated with v:

$$\forall x_v \in Dom(v),\ \mu_{a\to v}(x_v) = \sum_{\mathbf{x}'_a : x'_v = x_v} f_a(\mathbf{x}'_a) \prod_{v^* \in N(a)\setminus\{v\}} \mu_{v^*\to a}(x'_{v^*}).$$

where $N(a)$ is the set of neighbouring (variable) nodes to a. If $N(a) \setminus \{v\}$ is empty then $\mu_{a\to v}(x_v) = f_a(x_v)$, since in this case $x_v = x_a$.

As shown by the previous formula: the complete marginalisation is reduced to a sum of products of simpler terms than the ones appearing in the full joint distribution. This is the reason why it is called the sum-product algorithm.

In a typical run, each message will be updated iteratively from the previous value of the neighbouring messages. Different scheduling can be used for updating the messages. In the case where the graphical model is a tree, an optimal scheduling allows to reach convergence after computing each messages only once (see next sub-section). When the factor graph has cycles, such an optimal scheduling does not exist, and a typical choice is to update all messages simultaneously at each iteration.

Upon convergence (if convergence happened), the estimated marginal distribution of each node is proportional to the product of all messages from adjoining factors (missing the normalization constant):

$$p_{X_v}(x_v) \propto \prod_{a\in N(v)} \mu_{a\to v}(x_v).$$

Likewise, the estimated joint marginal distribution of the set of variables belonging to one factor is proportional to the product of the factor and the messages from the variables:

$$p_{X_a}(\mathbf{x}_a) \propto f_a(\mathbf{x}_a) \prod_{v\in N(a)} \mu_{v\to a}(x_v).$$

In the case where the factor graph is acyclic (i.e. is a tree or a forest), these estimated marginal actually converge to the true marginals in a finite number of iterations. This can be shown by mathematical induction.

6.1.1 Exact algorithm for trees

In the case when the factor graph is a tree, the belief propagation algorithm will compute the exact marginals. Furthermore, with proper scheduling of the message updates, it will terminate after 2 steps. This optimal scheduling can be described as follows:

Before starting, the graph is orientated by designating one node as the *root*; any non-root node which is connected to only one other node is called a *leaf*.

In the first step, messages are passed inwards: starting at the leaves, each node passes a message along the (unique) edge towards the root node. The tree structure guarantees that it is possible to obtain messages from all other adjoining nodes before passing the message on. This continues until the root has obtained messages from all of its adjoining nodes.

The second step involves passing the messages back out: starting at the root, messages are passed in the reverse direction. The algorithm is completed when all leaves have received their messages.

6.1.2 Approximate algorithm for general graphs

Curiously, although it was originally designed for acyclic graphical models, it was found that the Belief Propagation algorithm can be used in general graphs. The algorithm is then sometimes called "loopy" belief propagation, because graphs typically contain cycles, or loops. The initialization and scheduling of message updates must be adjusted slightly (compared with the previously described schedule for acyclic graphs) because graphs might not contain any leaves. Instead, one initializes all variable messages to 1 and uses the same message definitions above, updating all messages at every iteration (although messages coming from known leaves or tree-structured subgraphs may no longer need updating after sufficient iterations). It is easy to show that in a tree, the message definitions of this modified procedure will converge to the set of message definitions given above within a number of iterations equal to the diameter of the tree.

The precise conditions under which loopy belief propagation will converge are still not well understood; it is known that on graphs containing a single loop it converges in most cases, but the probabilities obtained might be incorrect.[6] Several sufficient (but not necessary) conditions for convergence of loopy belief propagation to a unique fixed point exist.[7] There exist graphs which will fail to converge, or which will oscillate between multiple states over repeated iterations. Techniques like EXIT charts can provide an approximate visualisation of the progress of belief propagation and an approximate test for convergence.

There are other approximate methods for marginalization including variational methods and Monte Carlo methods.

One method of exact marginalization in general graphs is called the junction tree algorithm, which is simply belief propagation on a modified graph guaranteed to be a tree. The basic premise is to eliminate cycles by clustering them into single nodes.

6.2 Related algorithm and complexity issues

A similar algorithm is commonly referred to as the Viterbi algorithm, but also known as a special case of the max-product or min-sum algorithm, which solves the related problem of maximization, or most probable explanation. Instead of attempting to solve the marginal, the goal here is to find the values $\mathbf{x}$ that maximises the global function (i.e. most probable values in a probabilistic setting), and it can be defined using the arg max:

* $arg\,max_{\mathbf{x}} g(\mathbf{x})$.

An algorithm that solves this problem is nearly identical to belief propagation, with the sums replaced by maxima in the definitions.[8]

It is worth noting that inference problems like marginalization and maximization are NP-hard to solve exactly and approximately (at least for relative error) in a graphical model. More precisely, the marginalization problem defined above is #P-complete and maximization is NP-complete.

The memory usage of belief propagation can be reduced through the use of the Island algorithm (at a small cost in time complexity).

6.3 Relation to free energy

The sum-product algorithm is related to the calculation of free energy in thermodynamics. Let Z be the partition function. A probability distribution

$$P(\mathbf{X}) = \frac{1}{Z} \prod_{f_j} f_j(x_j)$$

(as per the factor graph representation) can be viewed as a measure of the internal energy present in a system, computed as

$$E(\mathbf{X}) = \log \prod_{f_j} f_j(x_j).$$

The free energy of the system is then

$$F = U - H = \sum_{\mathbf{X}} P(\mathbf{X}) E(\mathbf{X}) + \sum_{\mathbf{X}} P(\mathbf{X}) \log P(\mathbf{X}).$$

It can then be shown that the points of convergence of the sum-product algorithm represent the points where the free energy in such a system is minimized. Similarly, it can be shown that a fixed point of the iterative belief propagation algorithm in graphs with cycles is a stationary point of a free energy approximation.[9]

6.4 Generalized belief propagation (GBP)

Belief propagation algorithms are normally presented as message update equations on a factor graph, involving messages between variable nodes and their neighboring factor nodes and vice versa. Considering messages between *regions* in a graph is one way of generalizing the belief propagation algorithm.[9] There are several ways of defining the set of regions in a graph that can exchange messages. One method uses ideas introduced by Kikuchi in the physics literature, and is known as Kikuchi's cluster variation method.

Improvements in the performance of belief propagation algorithms are also achievable by breaking the replicas symmetry in the distributions of the fields (messages). This generalization leads to a new kind of algorithm called survey propagation (SP), which have proved to be very efficient in NP-complete problems like satisfiability[1] and graph coloring.

The cluster variational method and the survey propagation algorithms are two different improvements to belief propagation. The name generalized survey propagation (GSP) is waiting to be assigned to the algorithm that merges both generalizations.

6.5 Gaussian belief propagation (GaBP)

Gaussian belief propagation is a variant of the belief propagation algorithm when the underlying distributions are Gaussian. The first work analyzing this special model was the seminal work of Weiss and Freeman [10]

The GaBP algorithm solves the following marginalization problem:

$$P(x_i) = \frac{1}{Z} \int_{j \neq i} \exp(-1/2 x^T A x + b^T x)\, dx_j$$

where Z is a normalization constant, A is a symmetric positive definite matrix (inverse covariance matrix a.k.a. precision matrix) and b is the shift vector.

Equivalently, it can be shown that using the Gaussian model, the solution of the marginalization problem is equivalent to the MAP assignment problem:

$$\underset{x}{\operatorname{argmax}}\ P(x) = \frac{1}{Z} \exp(-1/2 x^T A x + b^T x).$$

This problem is also equivalent to the following minimization problem of the quadratic form:

$$\min_x 1/2 x^T A x - b^T x.$$

Which is also equivalent to the linear system of equations

$$Ax = b.$$

Convergence of the GaBP algorithm is easier to analyze (relatively to the general BP case) and there are two known sufficient convergence conditions. The first one was formulated by Weiss et al. in the year 2000, when the information matrix A is diagonally dominant. The second convergence condition was formulated by Johnson et al.[11] in 2006, when the spectral radius of the matrix

$$\rho(I - |D^{-1/2}AD^{-1/2}|) < 1$$

where $D = \text{diag}(A)$. Later, Su and Wu established the necessary and sufficient convergence conditions for synchronous GaBP and damped GaBP, as well as another sufficient convergence condition for asynchronous GaBP. For each case, the convergence condition involves verifying 1) a set (determined by A) being non-empty, 2) the spectral radius of a certain matrix being smaller than one, and 3) the singularity issue (when converting BP message into belief) does not occur.[12]

The GaBP algorithm was linked to the linear algebra domain,[13] and it was shown that the GaBP algorithm can be viewed as an iterative algorithm for solving the linear system of equations $Ax = b$ where A is the information matrix and b is the shift vector. Empirically, the GaBP algorithm is shown to converge faster than classical iterative methods like the Jacobi method, the Gauss–Seidel method, successive over-relaxation, and others.[14] Additionally, the GaBP algorithm is shown to be immune to numerical problems of the preconditioned conjugate gradient method [15]

6.6 References

[1] Braunstein, A.; Mézard, R.; Zecchina, R. (2005). "Survey propagation: An algorithm for satisfiability". *Random Structures & Algorithms* **27** (2): 201–226. doi:10.1002/rsa.20057.

[2] Pearl, Judea (1982). "Reverend Bayes on inference engines: A distributed hierarchical approach" (PDF). *Proceedings of the Second National Conference on Artificial Intelligence*. AAAI-82: Pittsburgh, PA. Menlo Park, California: AAAI Press. pp. 133–136. Retrieved 2009-03-28.

[3] Kim, Jin H.; Pearl, Judea (1983). "A computational model for combined causal and diagnostic reasoning in inference systems" (PDF). *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*. IJCAI-83: Karlsruhe, Germany **1**. pp. 190–193. Retrieved 2013-01-03.

[4] Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (2nd ed.). San Francisco, CA: Morgan Kaufmann. ISBN 1-55860-479-0.

[5] Yedidia, J.S.; Freeman, W.T.; Y. (January 2003). "Understanding Belief Propagation and Its Generalizations". In Lakemeyer, Gerhard; Nebel, Bernhard. *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann. pp. 239–236. ISBN 1-55860-811-7. Retrieved 2009-03-30.

[6] Weiss, Yair (2000). "Correctness of Local Probability Propagation in Graphical Models with Loops". *Neural Computation* **12** (1): 1–41. doi:10.1162/089976600300015880.

[7] Mooij, J; Kappen, H (2007). "Sufficient Conditions for Convergence of the Sum–Product Algorithm". *IEEE Transactions on Information Theory* **53** (12): 4422–4437. doi:10.1109/TIT.2007.909166.

[8] Löliger, Hans-Andrea (2004). "An Introduction to Factor Graphs". *IEEE Signal Processing Magazine* **21**: 28–41. doi:10.1109/msp.2004.126704

[9] Yedidia, J.S.; Freeman, W.T.; Weiss, Y.; Y. (July 2005). "Constructing free-energy approximations and generalized belief propagation algorithms". *IEEE Transactions on Information Theory* **51** (7): 2282–2312. doi:10.1109/TIT.2005.850085. Retrieved 2009-03-28.

[10] Weiss, Yair; Freeman, William T. (October 2001). "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology". *Neural Computation* **13** (10): 2173–2200. doi:10.1162/089976601750541769. PMID 11570995.

[11] Malioutov, Dmitry M.; Johnson, Jason K.; Willsky, Alan S. (October 2006). "Walk-sums and belief propagation in Gaussian graphical models". *Journal of Machine Learning Research* **7**: 2031–2064. Retrieved 2009-03-28.

[12] Su, Qinliang; Wu, Yik-Chung (March 2015). "On convergence conditions of Gaussian belief propagation". *IEEE Trans. Signal Process.* **63** (5): 1144–1155. doi:10.1109/TSP.2015.2389755.

[13] Gaussian belief propagation solver for systems of linear equations. By O. Shental, D. Bickson, P. H. Siegel, J. K. Wolf, and D. Dolev, IEEE Int. Symp. on Inform. Theory (ISIT), Toronto, Canada, July 2008. http://www.cs.huji.ac.il/labs/danss/p2p/gabp/

[14] Linear Detection via Belief Propagation. Danny Bickson, Danny Dolev, Ori Shental, Paul H. Siegel and Jack K. Wolf. In the 45th Annual Allerton Conference on Communication, Control, and Computing, Allerton House, Illinois, 7 Sept.. http://www.cs.huji.ac.il/labs/danss/p2p/gabp/

[15] Distributed large scale network utility maximization. D. Bickson, Y. Tock, A. Zymnis, S. Boyd and D. Dolev. In the International symposium on information theory (ISIT), July 2009. http://www.cs.huji.ac.il/labs/danss/p2p/gabp/

6.7 Notes

- Frey, Brendan (1998). *Graphical Models for Machine Learning and Digital Communication*. MIT Press
- Löliger, Hans-Andrea (2004). *An Introduction to Factor Graphs.* IEEE Signal Proc. Mag. Vol.21. pages 28–41
- David J.C. MacKay (2003). Exact Marginalization in Graphs. In David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, pp. 334–340. Cambridge: Cambridge University Press.
- Mackenzie, Dana (2005). *Communication Speed Nears Terminal Velocity* New Scientist. 9 July 2005. Issue 2507 (Registration required)
- Yedidia, J.S.; Freeman, W.T.; Weiss, Y.; Y. (July 2005). "Constructing free-energy approximations and generalized belief propagation algorithms". *IEEE Transactions on Information Theory* **51** (7): 2282–2312. doi:10.1109/TIT.2005.850085. Retrieved 2009-03-28.
- Yedidia, J.S.; Freeman, W.T.; Y. (January 2003). "Understanding Belief Propagation and Its Generalizations". In Lakemeyer, Gerhard; Nebel, Bernhard. *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann. pp. 239–236. ISBN 1-55860-811-7. Retrieved 2009-03-30.
- Bishop, Christopher M (2006). "Chapter 8: Graphical models" (PDF). *Pattern Recognition and Machine Learning*. Springer. pp. 359–418. ISBN 0-387-31073-8. Retrieved 2014-03-20.
- Koch, Volker M. (2007). *A Factor Graph Approach to Model-Based Signal Separation* --- A tutorial-style dissertation
- Wymeersch, Henk (2007). *Iterative Receiver Design*. Cambridge University Press. ISBN 0-521-87315-0.
- Bickson, Danny. (2009). *Gaussian Belief Propagation Resource Page* --- Webpage containing recent publications as well as Matlab source code.
- Coughlan, James. (2009). *A Tutorial Introduction to Belief Propagation*.

Chapter 7

Causal graph

In statistics, econometrics, epidemiology, genetics and related disciplines, **causal graphs** (also known as path diagrams, causal Bayesian networks or DAGs) are graphical models used to encode assumptions about the data-generating process. They can also be viewed as a blueprint of the algorithm by which Nature assigns values to the variables in the domain of interest.

Causal graphs can be used for communication and for inference. As communication devices, the graphs provide formal and transparent representation of the causal assumptions that researchers may wish to convey and defend. As inference tools, the graphs enable researchers to estimate effect sizes from non-experimental data,[1][2][3][4][5] derive testable implications of the assumptions encoded,[1][6][7][8] test for external validity,[9] and manage missing data[10] and selection bias.[11]

Causal graphs were first used by the geneticist Sewall Wright[12] under the rubric "path diagrams". They were later adopted by social scientists[13][14][15][16][17][18] and, to a lesser extent, by economists.[19] These models were initially confined to linear equations with fixed parameters. Modern developments have extended graphical models to non-parametric analysis, and thus achieved a generality and flexibility that has transformed causal analysis in computer science, epidemiology,[20] and social science.[21]

7.1 Construction and terminology

The causal graph can be drawn in the following way. Each variable in the model has a corresponding vertex or node and an arrow is drawn from a variable X to a variable Y whenever Y is judged to respond to changes in X when all other variables are being held constant. Variables connected to Y through direct arrows are called *parents* of Y, or "direct causes of Y." and are denoted by $Pa(Y)$.

Causal models often include "error terms" or "omitted factors" which represent all unmeasured factors that influence a variable Y when $Pa(Y)$ are held constant. In most cases, error terms are excluded from the graph. However, if the graph author suspects that the error terms of any two variables are dependent (e.g. the two variables have an unobserved or latent common cause) then a bidirected arc is drawn between them. Thus, the presence of latent variables is taken into account through the correlations they induce between the error terms, as represented by bidirected arcs.

7.2 Fundamental tools

A fundamental tool in graphical analysis is d-separation, which allows researchers to determine, by inspection, whether the causal structure implies that two sets of variables are independent given a third set. In recursive models without correlated error terms (sometimes called *Markovian*), these conditional independences represent all of the model's testable implications.[22]

7.3 Example

Suppose we wish to estimate the effect of attending an elite college on future earnings. Simply regressing earnings on college rating will not give an unbiased estimate of the target effect because elite colleges are highly selective, and students attending them are likely to have qualifications for high-earning jobs prior to attending the school. Assuming that the causal relationships are linear, this background knowledge can be expressed in the following structural equation model (SEM) specification.

Model 1

$$
\begin{aligned}
Q_1 &= U_1 \\
C &= a \cdot Q_1 + U_2 \\
Q_2 &= c \cdot C + d \cdot Q_1 + U_3 \\
S &= b \cdot C + e \cdot Q_2 + U_4,
\end{aligned}
$$

where Q_1 represents the individual's qualifications prior to college, Q_2 represents qualifications after college, C contains attributes representing the quality of the college attended, and S the individual's salary.

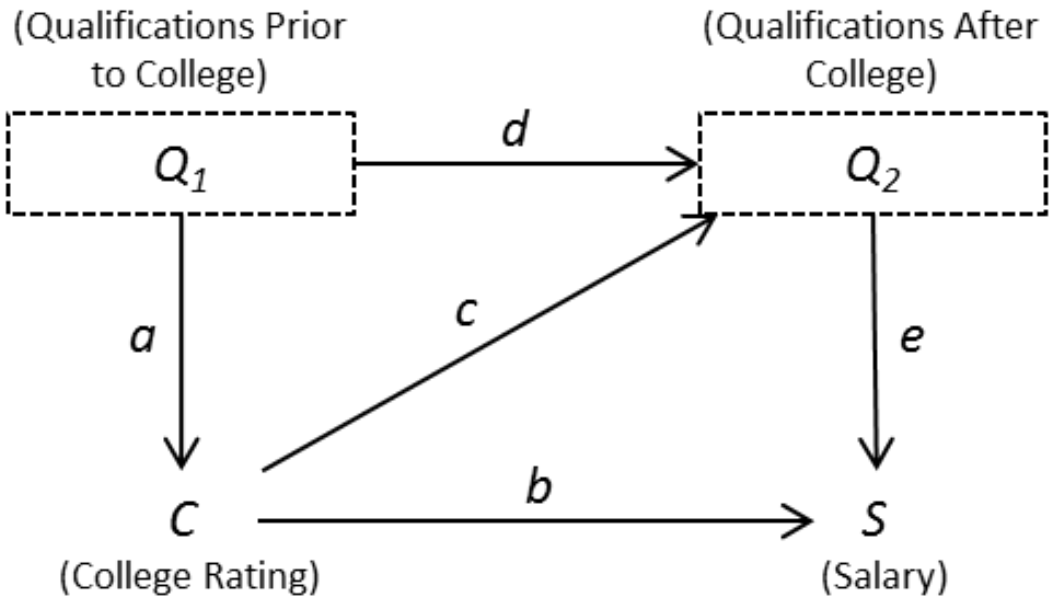


Figure 1: Unidentified model with latent variables (Q_1 and Q_2) shown explicitly

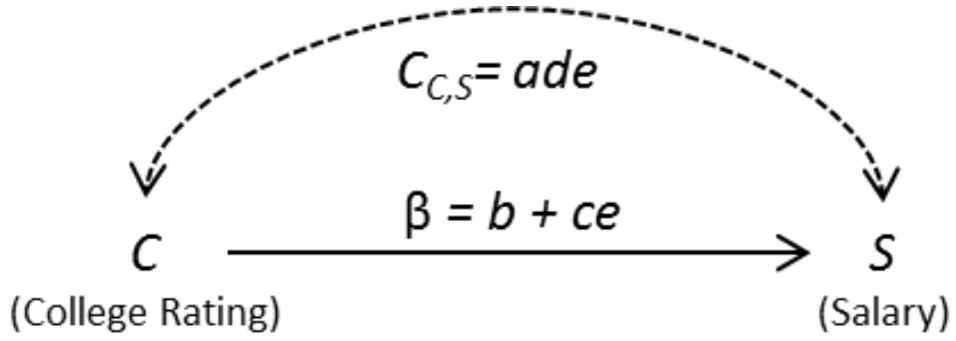


Figure 2: Unidentified model with latent variables summarized

Figure 1 is a causal graph that represents this model specification. Each variable in the model has a corresponding node or vertex in the graph. Additionally, for each equation, arrows are drawn from the independent variables to the

dependent variables. These arrows reflect the direction of causation. In some cases, we may label the arrow with its corresponding structural coefficient as in Figure 1.

If Q_1 and Q_2 are unobserved or latent variables their influence on C and S can be attributed to their error terms. By removing them, we obtain the following model specification:

Model 2

$$C = U_C$$
$$S = \beta C + U_S$$

The background information specified by Model 1 imply that the error term of S , U_S , is correlated with *C*'s error term, U_C . As a result, we add a bidirected arc between *S* and *C*, as in Figure 2.

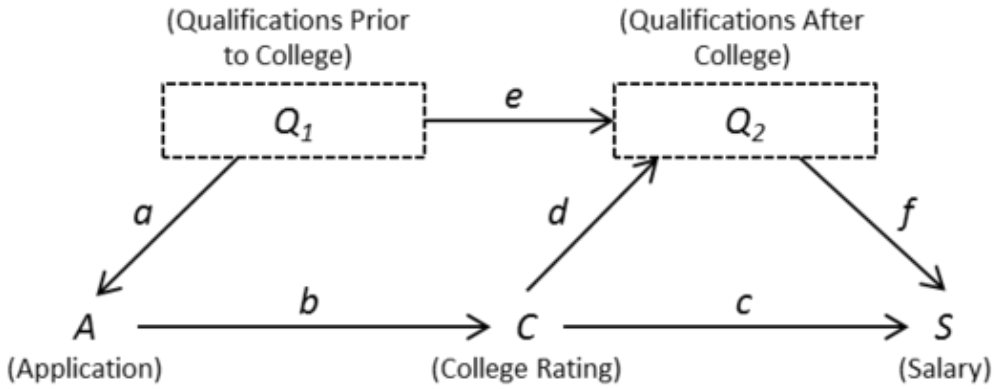


Figure 3: Identified model with latent variables (Q_1 and Q_2) shown explicitly

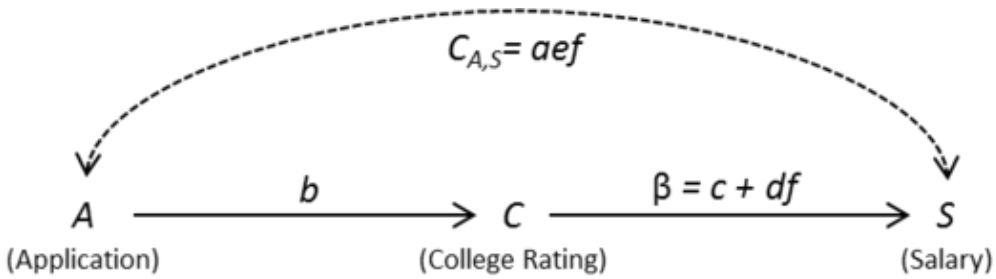


Figure 4: Identified model with latent variables summarized

Since U_S is correlated with U_C and, therefore, C , C is endogenous and β is not identified in Model 2. However, if we include the strength of an individual's college application, A , as shown in Figure 3, we obtain the following model:

Model 3

$$Q_1 = U_1$$
$$A = a \cdot Q_1 + U_2$$
$$C = b \cdot A + U_3$$
$$Q_2 = e \cdot Q_1 + d \cdot C + U_4$$
$$S = c \cdot C + f \cdot Q_2 + U_5,$$

By removing the latent variables from the model specification we obtain:

Model 4

$A = a \cdot Q_1 + U_A$
$C = b \cdot A + U_C$
$S = \beta \cdot C + U_S,$

with U_A correlated with U_S .

Now, β is identified and can be estimated using the regression of S on C and A . This can be verified using the *single-door criterion*,[1][23] a necessary and sufficient graphical condition for the identification of a structural coefficients, like β , using regression.

7.4 References

[1] Pearl, Judea (2000). *Causality*. Cambridge, MA: MIT Press.

[2] Tian, Jin; Pearl, Judea (2002). "A general identification condition for causal effects". *Proceedings of AAAI*.

[3] Shpitser, Ilya; Pearl, Judea (2008). "Complete Identification Methods for the Causal Hierarchy". *Journal of Machine Learning Research* **9**: 1941–1979.

[4] Huang, Y.; Valtorta, M. (2006). "Identifiability in causal bayesian networks: A sound and complete algorithm". *Proceedings of AAAI*.

[5] Bareinboim, Elias; Pearl, Judea (2012). "Causal Inference by Surrogate Experiments: z-Identifiability". *Proceedings of the UAI*.

[6] Tian, Jin; Pearl, Judea (2002). "On the Testable Implications of Causal Models with Hidden Variables". *Proceedings of UAI*: 519–27.

[7] Shpitser, Ilya; Pearl, Judea (2008). "Dormant Independence". *Proceedings of AAAI*.

[8] Chen, Bryant; Pearl, Judea (2014). "Testable Implications of Linear Structural Equation Models". *Proceedings of AAAI*.

[9] Bareinmboim, Elias; Pearl, Judea (2014). "External Validity: From do-calculus to Transportability across Populations". *Statistical Science*.

[10] Mohan, Karthika; Pearl, Judea; Tian, Jin (2013). "Graphical Models for Inference with Missing Data". *Advances in Neural Information Processing Systems*.

[11] Bareinboim, Elias; Tian, Jin; Pearl, Judea (2014). "Recovering from Selection Bias in Causal and Statistical Inference". *Proceedings of AAAI*.

[12] Wright, S. (1921). "Correlation and causation". *J. Agricultural Research* **20**: 557–585.

[13] Blalock, H. M. (1960). "Correlational analysis and causal inferences". *American Anthropologist* **62** (4): 624–631.

[14] Duncan, O. D. (1966). "Path analysis: Sociological examples.". *American Journal of Sociology*: 1–16.

[15] Duncan, O. D. (1976). "Introduction to structural equation models". *American Journal of Sociology* **82** (3): 731–733.

[16] Jöreskog, K. G. (1969). "A general approach to confirmatory maximum likelihood factor analysis". *Psychometrika*: 183–202.

[17] Goldberger, A. S.; Duncan, O. D. (1973). *Structural equation models in the social sciences*. New York: Seminar Press.

[18] Goldberger, A. S. (1972). "Structural equation models in the social sciences". *Econometrica*: 979–1001.

[19] White, Halbert; Chalak, Karim; Lu, Xun (2011). "Linking granger causality and the pearl causal model with settable systems". *Causality in Time Series Challenges in Machine Learning* **5**.

[20] Rothman, Kenneth J.; Greenland, Sander; Lash, Timothy (2008). *Modern epidemiology*. Lippincott Williams & Wilkins.

[21] Morgan, S. L.; Winship, C. (2007). *Counterfactuals and causal inference: Methods and principles for social research*. New York: Cambridge University Press.

[22] Geiger, Dan; Pearl, Judea (1993). "Logical and Algorithmic Properties of Conditional Independence". *Annals of Statistics* **21** (4): 2001–2021. doi:10.1214/aos/1176349407.

[23] Chen, B.; Pearl, J (2014). "Graphical Tools for Linear Structural Equation Modeling". *Technical Report*.

•

Chapter 8

Causal inference

Causal inference is the process of drawing a conclusion about a causal connection based on the conditions of the occurrence of an effect. The main difference between causal inference and inference of association is that the former analyzes the response of the effect variable when the cause is changed.[1][2] The science of why things occur is called etiology.

8.1 Definition

Inferring the cause of something has been described as

- "...reason[ing] to the conclusion that something is, or is likely to be, the cause of something else".[3]
- "Identification of the cause or causes of a phenomenon, by establishing covariation of cause and effect, a time-order relationship with the cause preceding the effect, and the elimination of plausible alternative causes."[4]

8.2 Methods

Epidemiological studies employ different epidemiological methods of collecting and measuring evidence of risk factors and effect and different ways of measuring association between the two. A hypothesis is formulated, and then tested with statistical methods (see Statistical hypothesis testing). It is statistical inference that helps decide if data are due to chance, also called random variation, or indeed correlated and if so how strongly.

Common frameworks for causal inference are structural equation modeling and the Rubin causal model.

8.3 In epidemiology

Epidemiology studies patterns of health and disease in defined populations of living beings, in order to infer causes and effects. An association between an exposure to a putative risk factor and a disease may be suggestive of, but is not equivalent to causality or correlation does not imply causation. Historically, Koch's postulates have been used since the 19th century to decide if a microorganism was the cause of a disease. In the 20th century the Bradford Hill criteria, described in 1965[5] have been used to assess causality of variables outside microbiology, although even these criteria are not exclusive ways to determine causality.

In molecular epidemiology the phenomena studied are on a molecular biology level, including genetics, where biomarkers are evidence of cause or effects.

A recent trend is to identify evidence for influence of the exposure on molecular pathology within diseased tissue or cells, in the emerging interdisciplinary field of molecular pathological epidemiology (MPE). Linking the exposure to molecular pathologic signatures of the disease can help to assess causality. Considering the inherent nature of heterogeneity of a given disease, the unique disease principle, disease phenotyping and subtyping are trends in biomedical and public health sciences, exemplified as personalized medicine and precision medicine.

8.4 In computer science

Determination of cause and effect from joint observational data for two time-independent variables, say X and Y, has been tackled using asymmetry between evidence for some model in the directions, X → Y and Y → X. One idea is to incorporate an independent noise term in the model to compare the evidences of the two directions.

Here are some of the noise models for the hypothesis Y → X with the noise E:

- Additive noise:[6] $Y = F(X) + E$
- Linear noise:[7] $Y = pX + qE$
- Post-non-linear:[8] $Y = G(F(X) + E)$
- Heteroskedastic noise: $Y = F(X) + E.G(X)$
- Functional noise:[9] $Y = F(X, E)$

The common assumption in these models are:

- There are no other causes of Y.
- X and E have no common causes.
- Distribution of cause is independent from causal mechanisms.

On an intuitive level, the idea is that the factorization of the joint distribution P(Cause,Effect) into P(Cause)*P(Effect | Cause) typically yields models of lower total complexity than the factorization into P(Effect)*P(Cause | Effect). Although the notion of "complexity" is intuitively appealing, it is not obvious how it should be precisely defined.[9]

8.5 Education

Graduate courses on causal inferences have been introduced to the curriculum of many schools.

- Karolinska Institutet, Department of Medical Epidemiology and Biostatistics
- University of Groningen, Department of Statistics & Measurement Theory
- Harvard University, School of Public Health
- McGill University, Department of Epidemiology, Biostatistics and Occupational Health
- The University of British Columbia, School of Population and Public Health

8.6 See also

- Epidemiological method
- Granger causality
- Molecular pathological epidemiology
- Multivariate statistics
- Partial least squares regression
- Pathogenesis
- Pathology
- Regression analysis
- Transfer entropy

8.7 References

[1] Pearl, Judea (1 January 2009). "Causal inference in statistics: An overview" (PDF). *Statistics Surveys* **3**: 96–146. doi:10.1214/09-SS057.

[2] Morgan, Stephen; Winship, Chris (2007). *Counterfactuals and Causal inference*. Cambridge University Press. ISBN 978-0-521-67193-4.

[3] "causal inference". Encyclopædia Britannica, Inc. Retrieved 24 August 2014.

[4] John Shaughnessy; Eugene Zechmeister; Jeanne Zechmeister (2000). *Research Methods in Psychology*. McGraw-Hill Humanities/Social Sciences/Languages. pp. Chapter 1 : Introduction. ISBN 0077825365. Retrieved 24 August 2014.

[5] Hill, Austin Bradford (1965). "The Environment and Disease: Association or Causation?". *Proceedings of the Royal Society of Medicine* **58** (5): 295–300. PMC 1898525. PMID 14283879.

[6] Hoyer, Patrik O., et al. "Nonlinear causal discovery with additive noise models." NIPS. Vol. 21. 2008.

[7] Shimizu, Shohei, et al. "DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model." The Journal of Machine Learning Research 12 (2011): 1225-1248.

[8] Zhang, Kun, and Aapo Hyvärinen. "On the identifiability of the post-nonlinear causal model." Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009.

[9] Mooij, Joris M., et al. "Probabilistic latent variable models for distinguishing between cause and effect." NIPS. 2010.

8.8 External links

- NIPS 2013 Workshop on Causality
- Causal inference at the Max-Planck-Institute for Intelligent Systems Tübingen

Chapter 9

Causal loop diagram

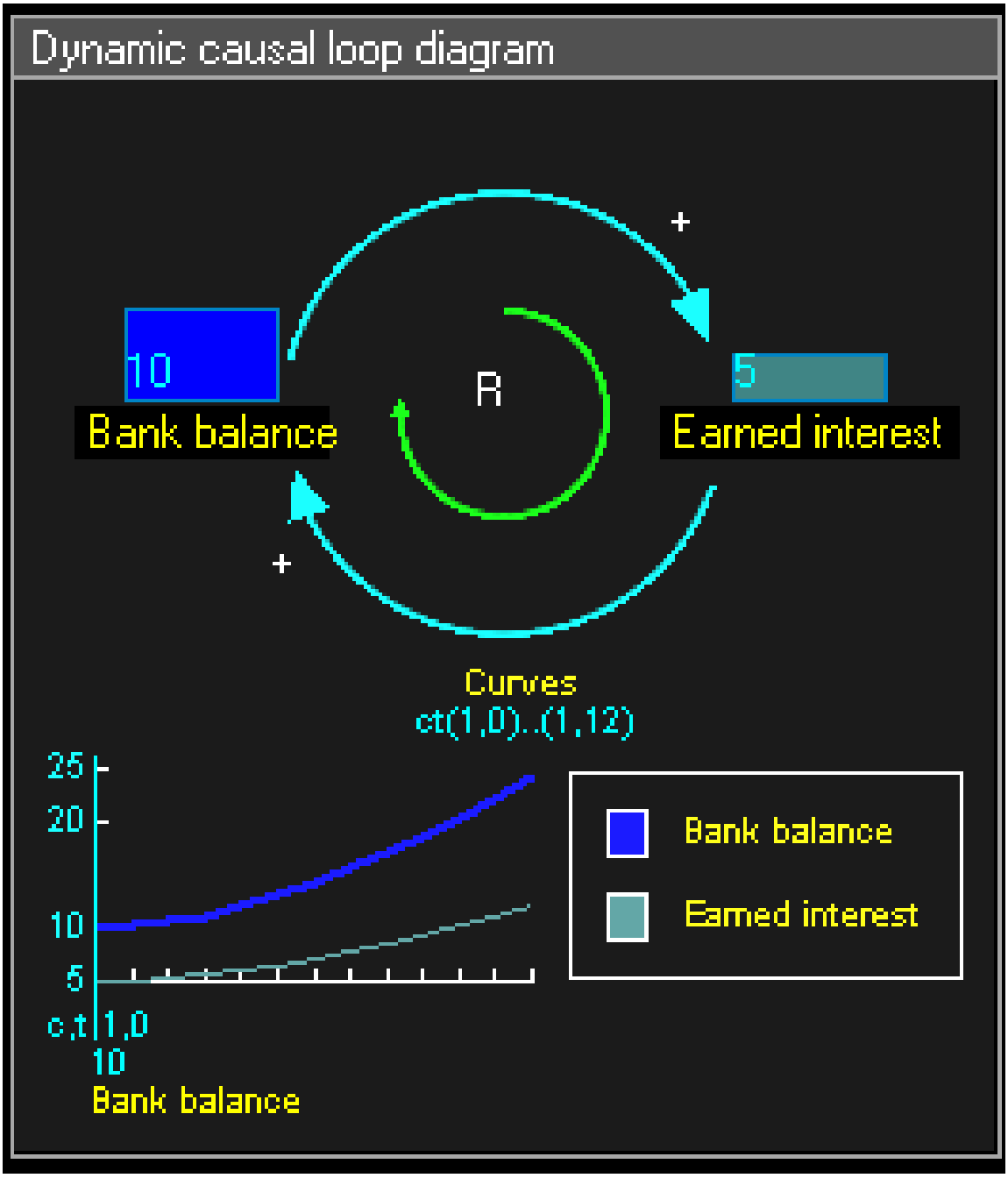
A **causal loop diagram** (CLD) is a causal diagram that aids in visualizing how different variables in a system are interrelated. The diagram consists of a set of nodes and edges. Nodes represent the variables and edges are the links that represent a connection or a relation between the two variables. A link marked **positive** indicates a positive relation and a link marked negative indicates a negative relation. A positive causal link means the two nodes change in the same direction, i.e. if the node in which the link starts decreases, the other node also decreases. Similarly, if the node in which the link starts increases, the other node increases as well. A **negative** causal link means the two nodes change in opposite directions, i.e. if the node in which the link starts increases, the other node decreases and vice versa.
Closed cycles in the diagram are very important features of the CLDs. A closed cycle is either defined as a **reinforcing** or **balancing** loop. A reinforcing loop is a cycle in which the effect of a variation in any variable propagates through the loop and returns to the variable reinforcing the initial deviation i.e. if a variable increases in a reinforcing loop the effect through the cycle will return an increase to the same variable and vice versa. A balancing loop is the cycle in which the effect of a variation in any variable propagates through the loop and returns to the variable a deviation opposite to the initial one i.e. if a variable increases in a balancing loop the effect through the cycle will return a decrease to the same variable and vice versa.
If a variable varies in a reinforcing loop the effect of the change reinforces the initial variation. The effect of the variation will then create another reinforcing effect. Without breaking the loop the system will be caught in a vicious cycles of circular chain reactions. For this reason, closed loops are critical features in the CLDs.
Example of positive reinforcing loop:
The amount of the *Bank Balance* will affect the amount of the *Earned Interest*, as represented by the top blue arrow, pointing from *Bank Balance* to *Earned Interest*.
Since an increase in *Bank balance* results in an increase in *Earned Interest*, this link is positive, which is denoted with a ""+"".
The *Earned interest* gets added to the *Bank balance*, also a positive link, represented by the bottom blue arrow.
The causal effect between these nodes forms a **positive** reinforcing loop, represented by the green arrow, which is denoted with an “R”.[1]

9.1 History

Main article: System Dynamics

The use of nodes and arrows to construct directed graph models of cause and effect dates back to the invention of path analysis by Sewall Wright in 1918, long before System Dynamics. Due to the limitations of genetic data, however, these early causal graphs contained no loops — they were directed acyclic graphs. The first formal use of Causal Loop Diagrams was explained by Dr. Dennis Meadows at a conference for educators (Systems Thinking & Dynamic Modeling Conference for K-12 Education in New Hampshire sponsored by Creative Learning Exchange [2]).

Meadows explained that when he and others were working on the World3 model (circa 1970–72), they realized they would not be able to use the computer output to explain how the feedback loops worked in their model when presenting their results to others. They decided to show feedback loops (without the stocks, flows and every variable), using arrows connecting the names of major model components in the feedback loops. This may have been the first



Example of positive reinforcing loop: Bank balance *and* Earned interest

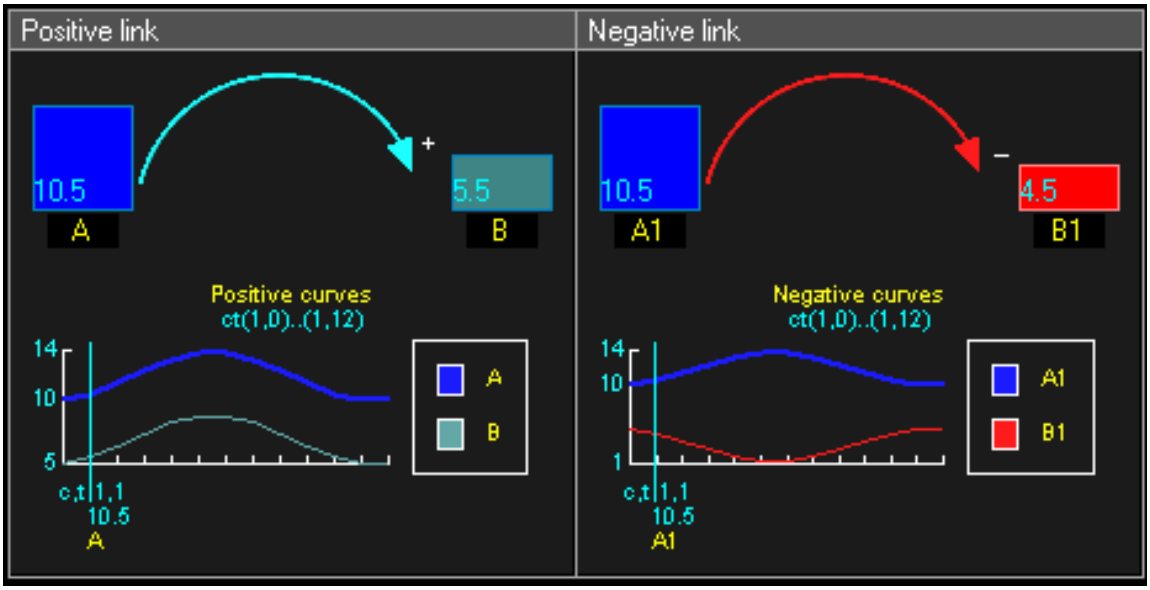
formal use of Causal Loop Diagrams.[3]

9.2 Positive and negative causal links

- **Positive causal link** means that the two nodes change in the same direction, i.e. if the node in which the link starts decreases, the other node also decreases. Similarly, if the node in which the link starts increases, the other node increases.

- **Negative causal link** means that the two nodes change in opposite directions, i.e. if the node in which the link starts increases, then the other node decreases, and vice versa.

9.2.1 Example



Dynamic causal loop diagram: positive and negative links

9.3 Reinforcing and balancing loops

To determine if a causal loop is reinforcing or balancing, one can start with an assumption, e.g. “Node 1 increases” and follow the loop around. The loop is:

- **reinforcing** if, after going around the loop, one ends up with the same result as the initial assumption.
- **balancing** if the result contradicts the initial assumption.

Or to put it in other words:

- reinforcing loops have an even number of negative links (zero also is even, see example below)
- balancing loops have an odd number of negative links.

Identifying reinforcing and balancing loops is an important step for identifying *Reference Behaviour Patterns*, i.e. possible dynamic behaviours of the system.

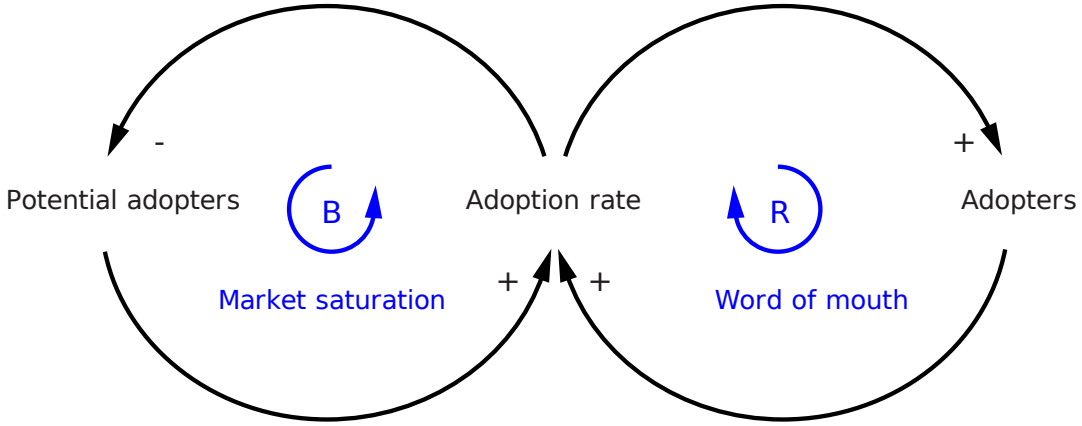
- Reinforcing loops are associated with exponential increases/decreases.
- Balancing loops are associated with reaching a plateau.

If the system has delays (often denoted by drawing a short line across the causal link), the system might fluctuate.

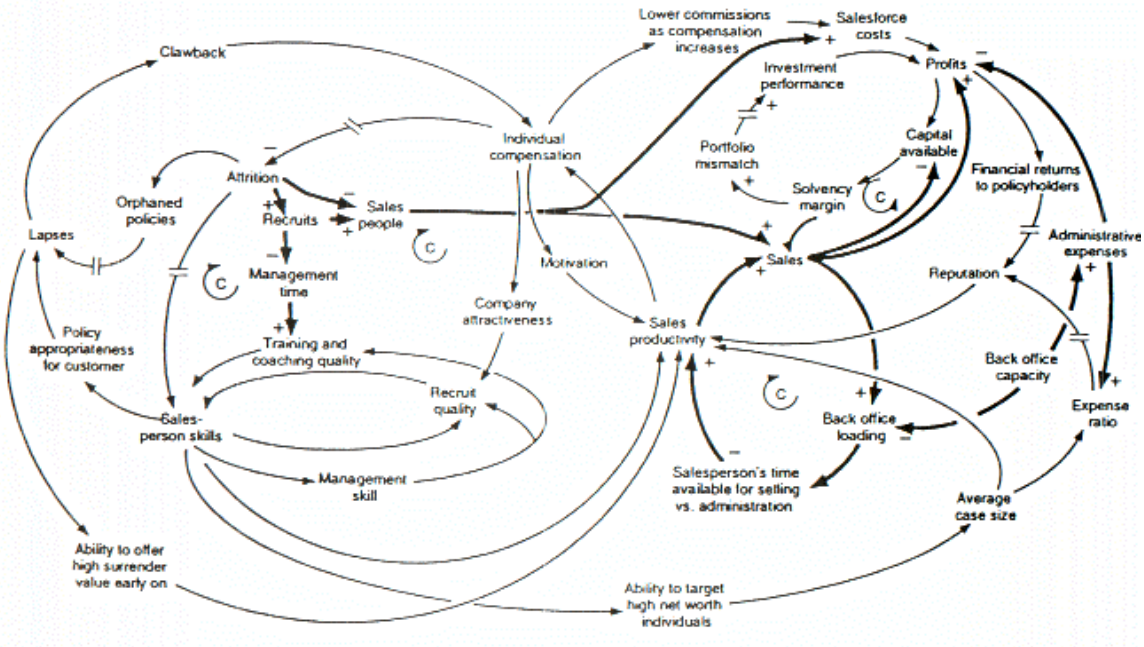
9.3.1 Example

9.4 See also

- Bayesian network
- Directed acyclic graph



Causal loop diagram of Adoption *model, used to demonstrate systems dynamics*



Causal loop diagram of a model examining the growth or decline of a life insurance company

- Negative feedback
- Path analysis (statistics)
- Positive feedback
- System dynamics

9.5 References

[1] John D.Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World.* McGraw Hill/Irwin, 2000. ISBN 9780072389159

[2] http://www.clexchange.org/

[3] Anecdote by Richard Turnock attending informal discussion where Dennis Meadows explained origin of CLD

9.6 External links

- WikiSD the System Dynamics Society Wiki
- Learn to Read Causal Loop Diagrams via SystemsAndUs

Chapter 10

Causal Markov condition

The **Markov condition** (sometimes called *Markov assumption*) for a Bayesian network states that any node in a Bayesian network is conditionally independent of its nondescendents, given its parents.

A node is conditionally independent of the entire network, given its Markov blanket.

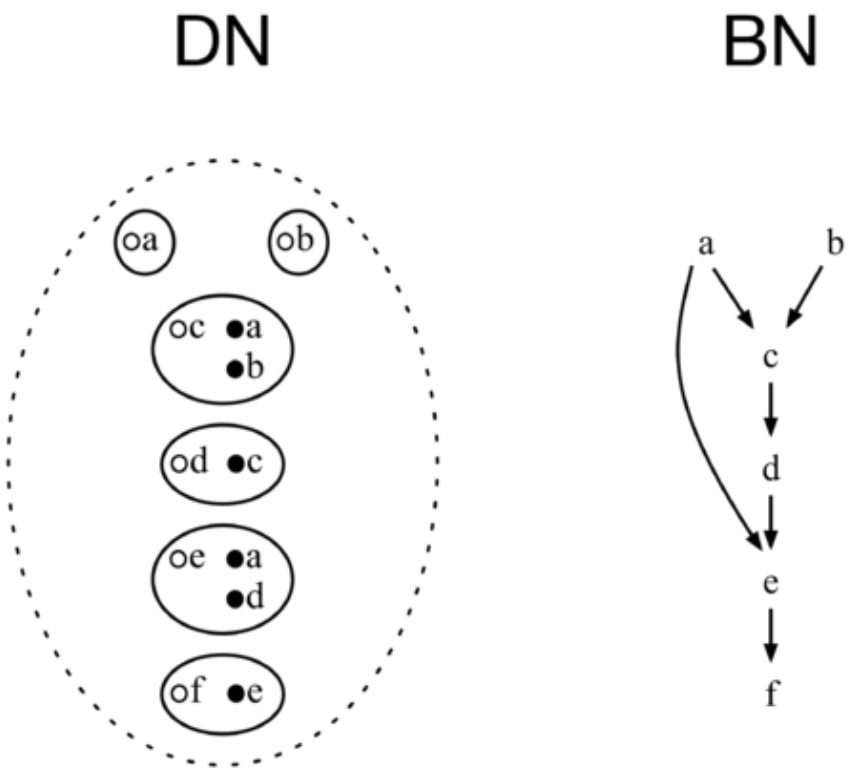
The related **causal Markov condition** is that a phenomenon is independent of its noneffects, given its direct causes.[1] In the event that the structure of a Bayesian network accurately depicts causality, the two conditions are equivalent. However, a network may accurately embody the Markov condition without depicting causality, in which case it should not be assumed to embody the causal Markov condition.

10.1 Notes

[1] Hausman, D.M.; Woodward, J. (December 1999). "Independence, Invariance, and the Causal Markov Condition" (PDF). *British Journal for the Philosophy of Science* **50** (4): 521–583. doi:10.1093/bjps/50.4.521.

Chapter 11

Darwinian network



A Darwinian network diagram that shows six populations, including $p(c, ab)$, short for $p(c, a, b)$, illustrated with a closed curve around the (clear) combative trait b and two (dark) docile traits a and g .

A **Darwinian network** (DN),[1] proposed in 2015 by,[2] is a probabilistic graphical model to simplify working with Bayesian networks.[3]

Rather than modelling the variables in a problem domain, DNs represent the probability tables in the model. The graphical manipulation of the tables then takes on a biological feel, where a CPT $P(X|Y)$ is viewed as the novel representation of a *population* $p(C, D)$ using both *combative* traits C (coloured clear) and *docile* traits D (coloured dark).

DNs can unify modeling and reasoning tasks into a single platform. DNs can represent exact inference using either *variable elimination*[4] or *arc-reversal*,[5] *lazy propagation*,[6] as well as how DNs can represent testing independencies. Adaptation and evolution are used to represent the testing of independencies and inference, respectively.

11.1 References

[1] http://www.darwiniannetworks.com/

[2] Butz, C. J.; Oliveira, J. S.; and dos Santos, A. E. 2015. Darwinian networks. In Proceedings of the Twenty-Eighth Canadian Artificial Intelligence Conference.

[3] Pearl, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann.

[4] Zhang, N.L., Poole, D.: A simple approach to bayesian network computations. In: Tenth Canadian Artificial Intelligence Conference. pp. 171–178 (1994)

[5] Olmsted, S.: On Representing and Solving Decision Problems. Ph.D. thesis, Stanford University (1983)

[6] Madsen, A. L., and Jensen, F. V. 1999. Lazy propagation: A junction tree inference algorithm based on lazy evaluation. Artificial Intelligence 113(1-2):203–245.

Chapter 12

Dempster–Shafer theory

Arthur P. Dempster at the Workshop on Theory of Belief Functions (Brest, 1 April 2010).

The theory of belief functions, also referred to as evidence theory or **Dempster–Shafer theory** (**DST**), is a general framework for reasoning with uncertainty, with understood connections to other frameworks such as probability, possibility and imprecise probability theories. First introduced by Arthur P. Dempster[1] in the context of statistical inference, the theory was later developed by Glenn Shafer into a general framework for modeling epistemic uncertainty - a mathematical theory of evidence.[2][3] The theory allows one to combine evidence from different sources and arrive at a degree of belief (represented by a mathematical object called *belief function*) that takes into account

all the available evidence.

In a narrow sense, the term Dempster–Shafer theory refers to the original conception of the theory by Dempster and Shafer. However, it is more common to use the term in the wider sense of the same general approach, as adapted to specific kinds of situations. In particular, many authors have proposed different rules for combining evidence, often with a view to handling conflicts in evidence better.[4] The early contributions have also been the starting points of many important developments, including the Transferable Belief Model and the Theory of Hints.[5]

12.1 Overview

Dempster–Shafer theory is a generalization of the Bayesian theory of subjective probability. Belief functions base degrees of belief (or confidence, or trust) for one question on the probabilities for a related question. The degrees of belief itself may or may not have the mathematical properties of probabilities; how much they differ depends on how closely the two questions are related.[6] Put another way, it is a way of representing epistemic plausibilities but it can yield answers that contradict those arrived at using probability theory.

Often used as a method of sensor fusion, Dempster–Shafer theory is based on two ideas: obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule[7] for combining such degrees of belief when they are based on independent items of evidence. In essence, the degree of belief in a proposition depends primarily upon the number of answers (to the related questions) containing the proposition, and the subjective probability of each answer. Also contributing are the rules of combination that reflect general assumptions about the data.

In this formalism a **degree of belief** (also referred to as a **mass**) is represented as a **belief function** rather than a Bayesian probability distribution. Probability values are assigned to *sets* of possibilities rather than single events: their appeal rests on the fact they naturally encode evidence in favor of propositions.

Dempster–Shafer theory assigns its masses to all of the non-empty subsets of the propositions that compose a system. (In set-theoretic terms, the Power set of the propositions.) For instance, assume a situation where there are two related questions, or propositions, in a system. In this system, any belief function assigns mass to the first proposition, the second, both or neither.

12.1.1 Belief and plausibility

Shafer's formalism starts from a set of *possibilities* under consideration, for instance numerical values of a variable, or pairs of linguistic variables like "date and place of origin of a relic" (asking whether it is antique or a recent fake). A hypothesis is represented by a subset of this *frame of discernment*, like "(Ming dynasty, China)", or "(19th century, Germany)".[2]:p.35f.

Shafer's framework allows for belief about such propositions to be represented as intervals, bounded by two values, *belief* (or *support*) and *plausibility*:

$$belief \leq plausibility.$$

In a first step, subjective probabilities (*masses*) are assigned to all subsets of the frame; usually, only a restricted number of sets will have non-zero mass (*focal elements*).[2]:39f. *Belief* in a hypothesis is constituted by the sum of the masses of all sets enclosed by it. It is the amount of belief that directly supports a given hypothesis or a more specific one, forming a lower bound. Belief (usually denoted *Bel*) measures the strength of the evidence in favor of a proposition p. It ranges from 0 (indicating no evidence) to 1 (denoting certainty). *Plausibility* is 1 minus the sum of the masses of all sets whose intersection with the hypothesis is empty. Or, it can be obtained as the sum of the masses of all sets whose intersection with the hypothesis is not empty. It is an upper bound on the possibility that the hypothesis could be true, *i.e.* it "could possibly be the true state of the system" up to that value, because there is only so much evidence that contradicts that hypothesis. Plausibility (denoted by *Pl*) is defined to be $Pl(p)=1-Bel(\sim p)$. It also ranges from 0 to 1 and measures the extent to which evidence in favor of $\sim p$ leaves room for belief in p.

For example, suppose we have a belief of 0.5 and a plausibility of 0.8 for a proposition, say "the cat in the box is dead." This means that we have evidence that allows us to state strongly that the proposition is true with a confidence of 0.5. However, the evidence contrary to that hypothesis (i.e. "the cat is alive") only has a confidence of 0.2. The remaining mass of 0.3 (the gap between the 0.5 supporting evidence on the one hand, and the 0.2 contrary evidence

on the other) is "indeterminate," meaning that the cat could either be dead or alive. This interval represents the level of uncertainty based on the evidence in your system.

The null hypothesis is set to zero by definition (it corresponds to "no solution"). The orthogonal hypotheses "Alive" and "Dead" have probabilities of 0.2 and 0.5, respectively. This could correspond to "Live/Dead Cat Detector" signals, which have respective reliabilities of 0.2 and 0.5. Finally, the all-encompassing "Either" hypothesis (which simply acknowledges there is a cat in the box) picks up the slack so that the sum of the masses is 1. The belief for the "Alive" and "Dead" hypotheses matches their corresponding masses because they have no subsets; belief for "Either" consists of the sum of all three masses (Either, Alive, and Dead) because "Alive" and "Dead" are each subsets of "Either". The "Alive" plausibility is $1 - m$ (Dead) and the "Dead" plausibility is $1 - m$ (Alive). In other way, the "Alive" plausibility is m(Alive) + m (Either) and the "Dead" plausibility is m(Dead) + m(Either). Finally, the "Either" plausibility sums m(Alive) + m(Dead) + m(Either). The universal hypothesis ("Either") will always have 100% belief and plausibility —it acts as a checksum of sorts.

Here is a somewhat more elaborate example where the behavior of belief and plausibility begins to emerge. We're looking through a variety of detector systems at a single faraway signal light, which can only be coloured in one of three colours (red, yellow, or green):

Events of this kind would not be modeled as disjoint sets in probability space as they are here in mass assignment space. Rather the event "Red or Yellow" would be considered as the union of the events "Red" and "Yellow", and (see probability axioms) P(Red or Yellow) ≥ P(Yellow), and P(Any)=1, where *Any* refers to *Red* or *Yellow* or *Green*. In DST the mass assigned to *Any* refers to the proportion of evidence that can't be assigned to any of the other states, which here means evidence that says there is a light but doesn't say anything about what color it is. In this example, the proportion of evidence that shows the light is either *Red* or *Green* is given a mass of 0.05. Such evidence might, for example, be obtained from a R/G color blind person. DST lets us extract the value of this sensor's evidence. Also, in DST the Null set is considered to have zero mass, meaning here that the signal light system exists and we are examining its possible states, not speculating as to whether it exists at all.

12.1.2 Combining beliefs

Beliefs from different sources can be combined with various fusion operators to model specific situations of belief fusion, e.g. with Dempster's rule of combination, which combines belief constraints[8] that are dictated by independent belief sources, such as in the case of combining hints[5] or combining preferences.[9] Note that the probability masses from propositions that contradict each other can be used to obtain a measure of conflict between the independent belief sources. Other situations can be modeled with different fusion operators, such as cumulative fusion of beliefs from independent sources which can be modeled with the cumulative fusion operator.[10]

Dempster's rule of combination is sometimes interpreted as an approximate generalisation of Bayes' rule. In this interpretation the priors and conditionals need not be specified, unlike traditional Bayesian methods, which often use a symmetry (minimax error) argument to assign prior probabilities to random variables (*e.g.* assigning 0.5 to binary values for which no information is available about which is more likely). However, any information contained in the missing priors and conditionals is not used in Dempster's rule of combination unless it can be obtained indirectly—and arguably is then available for calculation using Bayes equations.

Dempster–Shafer theory allows one to specify a degree of ignorance in this situation instead of being forced to supply prior probabilities that add to unity. This sort of situation, and whether there is a real distinction between *risk* and *ignorance*, has been extensively discussed by statisticians and economists. See, for example, the contrasting views of Daniel Ellsberg, Howard Raiffa, Kenneth Arrow and Frank Knight.

12.2 Formal definition

Let X be the *universe*: the set representing all possible states of a system under consideration. The power set

$$2^X$$

is the set of all subsets of X, including the empty set $\emptyset$. For example, if:

$$X = \{a, b\}$$

then

$$2^X = \{\emptyset, \{a\}, \{b\}, X\}.$$

The elements of the power set can be taken to represent propositions concerning the actual state of the system, by containing all and only the states in which the proposition is true.

The theory of evidence assigns a belief mass to each element of the power set. Formally, a function

$$m : 2^X \to [0, 1]$$

is called a *basic belief assignment* (BBA), when it has two properties. First, the mass of the empty set is zero:

$$m(\emptyset) = 0.$$

Second, the masses of the remaining members of the power set add up to a total of 1:

$$\sum_{A \in 2^X} m(A) = 1$$

The mass $m(A)$ of A, a given member of the power set, expresses the proportion of all relevant and available evidence that supports the claim that the actual state belongs to A but to no particular subset of A. The value of $m(A)$ pertains *only* to the set A and makes no additional claims about any subsets of A, each of which have, by definition, their own mass.

From the mass assignments, the upper and lower bounds of a probability interval can be defined. This interval contains the precise probability of a set of interest (in the classical sense), and is bounded by two non-additive continuous measures called **belief** (or **support**) and **plausibility**:

$$\text{bel}(A) \leq P(A) \leq \text{pl}(A).$$

The belief bel(A) for a set A is defined as the sum of all the masses of subsets of the set of interest:

$$\text{bel}(A) = \sum_{B | B \subseteq A} m(B).$$

The plausibility pl(A) is the sum of all the masses of the sets B that intersect the set of interest A:

$$\text{pl}(A) = \sum_{B | B \cap A \neq \emptyset} m(B).$$

The two measures are related to each other as follows:

$$\text{pl}(A) = 1 - \text{bel}(\overline{A}).$$

And conversely, for finite A, given the belief measure bel(B) for all subsets B of A, we can find the masses m(A) with the following inverse function:

$$m(A) = \sum_{B | B \subseteq A} (-1)^{|A-B|} \, \text{bel}(B)$$

where $|A - B|$ is the difference of the cardinalities of the two sets.[4]

It follows from the last two equations that, for a finite set X, you need know only one of the three (mass, belief, or plausibility) to deduce the other two; though you may need to know the values for many sets in order to calculate one of the other values for a particular set. In the case of an infinite X, there can be well-defined belief and plausibility functions but no well-defined mass function.[11]

12.3 Dempster's rule of combination

The problem we now face is how to combine two independent sets of probability mass assignments in specific situations. In case different sources express their beliefs over the frame in terms of belief constraints such as in case of giving hints or in case of expressing preferences, then Dempster's rule of combination is the appropriate fusion operator. This rule derives common shared belief between multiple sources and ignores *all* the conflicting (non-shared) belief through a normalization factor. Use of that rule in other situations than that of combining belief constraints has come under serious criticism, such as in case of fusing separate beliefs estimates from multiple sources that are to be integrated in a cumulative manner, and not as constraints. Cumulative fusion means that all probability masses from the different sources are reflected in the derived belief, so no probability mass is ignored.

Specifically, the combination (called the **joint mass**) is calculated from the two sets of masses m_1 and m_2 in the following manner:

$$m_{1,2}(\emptyset) = 0$$

$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B) m_2(C)$$

where

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C).$$

K is a measure of the amount of conflict between the two mass sets.

12.3.1 Effects of conflict

The normalization factor above, $1 - K$, has the effect of completely ignoring conflict and attributing *any* mass associated with conflict to the null set. This combination rule for evidence can therefore produce counterintuitive results, as we show next.

Example producing correct results in case of high conflict

The following example shows how Dempster's rule produces intuitive results when applied in a preference fusion situation, even when there is high conflict.

> Suppose that two friends, Alice and Bob, want to see a film at the cinema one evening, and that there are only three films showing: X, Y and Z. Alice expresses her preference for film X with probability 0.99, and her preference for film Y with a probability of only 0.01. Bob expresses his preference for film Z with probability 0.99, and his preference for film Y with a probability of only 0.01. When combining the preferences with Dempster's rule of combination it turns out that their combined preference results in probability 1.0 for film Y, because it is the only film that they both agree to see.

> Dempster's rule of combination produces intuitive results even in case of totally conflicting beliefs when interpreted in this way. Assume that Alice prefers film X with probability 1.0, and that Bob prefers film Z with probability 1.0. When trying to combine their preferences with Dempster's rule it turns out that it is undefined in this case, which means that there is no solution. This would mean that they can not agree on seeing any film together, so they don't go to the cinema together that evening. However, the semantics of interpreting preference as a probability is vague - if it is referring to the probability of seeing film X tonight, then we face the Fallacy of the excluded middle: the event that actually occurs, seeing none of the films tonight, has a probability mass of 0.

Example producing counter-intuitive results in case of high conflict

An example with exactly the same numerical values was introduced by Zadeh in 1979,[12][13][14] to point out counter-intuitive results generated by Dempster's rule when there is a high degree of conflict. The example goes as follows:

> Suppose that one has two equi-reliable doctors and one doctor believes a patient has either a brain tumor— with a probability (i.e. a basic belief assignment - bba's, or mass of belief) of 0.99 — or meningitis—with a probability of only 0.01. A second doctor believes the patient has a concussion — with a probability of 0.99 — and believes the patient suffers from meningitis — with a probability of only 0.01. Applying Dempster's rule to combine these two sets of masses of belief, one gets finally *m(meningitis)=1* (the meningitis is diagnosed with 100 percent of confidence).

Such result goes against the common sense since both doctors agree that there is a little chance that the patient has a meningitis. This example has been the starting point of many research works for trying to find a solid justification for Dempster's rule and for foundations of Dempster-Shafer Theory[15][16] or to show the inconsistencies of this theory.[17][18][19]

Example producing counter-intuitive results in case of low conflict

The following example shows where Dempster's rule produces a counter-intuitive result, even when there is low conflict.

> Suppose that one doctor believes a patient has either a brain tumor, with a probability of 0.99, or meningitis, with a probability of only 0.01. A second doctor also believes the patient has a brain tumor, with a probability of 0.99, and believes the patient suffers from concussion, with a probability of only 0.01. If we calculate m (brain tumor) with Dempster's rule, we obtain

$$m(\text{tumor brain}) = \text{Bel}(\text{tumor brain}) = 1.$$

This result implies *complete support* for the diagnosis of a brain tumor, which both doctors believed *very likely*. The agreement arises from the low degree of conflict between the two sets of evidence comprised by the two doctors' opinions.

In either case, it would be reasonable to expect that:

$$m(\text{tumor brain}) < 1 \text{ and } \text{Bel}(\text{tumor brain}) < 1,$$

since the existence of non-zero belief probabilities for other diagnoses implies *less than complete support* for the brain tumour diagnosis.

12.4 Bayesian theory as a special case

As in Dempster–Shafer theory, a Bayesian belief function $m : 2^X \to [0, 1]$ has the properties $\text{bel}(\emptyset) = 0$ and $\text{bel}(X) = 1$. The third condition, however, is subsumed by, but relaxed in DS theory:[2]:p. 19

$$\text{If} A \cap B = \emptyset, \text{then } \text{bel}(A \cup B) = \text{bel}(A) + \text{bel}(B).$$

Equivalently, each of the following conditions defines the Bayesian special case of the DS theory:[2]:p. 37,45

- $\text{bel}(A) + \text{bel}(\bar{A}) = 1$ all for $A \subseteq X$.
- For finite *X*, all focal elements of the belief function are singletons.

Bayes' conditional probability is a special case of Dempster's rule of combination.[2]:p. 19f.

12.5 Criticism

Judea Pearl (1988a, chapter 9;[20] 1988b[21] and 1990)[22] has argued that it is misleading to interpret belief functions as representing either "probabilities of an event," or "the confidence one has in the probabilities assigned to various outcomes," or "degrees of belief (or confidence, or trust) in a proposition," or "degree of ignorance in a situation." Instead, belief functions represent the probability that a given proposition is *provable* from a set of other propositions, to which probabilities are assigned. Confusing probabilities of *truth* with probabilities of *provability* may lead to counterintuitive results in reasoning tasks such as (1) representing incomplete knowledge, (2) belief-updating and (3) evidence pooling. He further demonstrated that, if partial knowledge is encoded and updated by belief function methods, the resulting beliefs cannot serve as a basis for rational decisions.

Kłopotek and Wierzchoń[23] proposed to interpret the Dempster–Shafer theory in terms of statistics of decision tables (of the rough set theory), whereby the operator of combining evidence should be seen as relational joining of decision tables. In another interpretation M.A. Kłopotek and S.T. Wierzchoń[24] propose to view this theory as describing destructive material processing (under loss of properties), *e.g.* like in some semiconductor production processes. Under both interpretations reasoning in DST gives correct results, contrary to the earlier probabilistic interpretations, criticized by Pearl in the cited papers and by other researchers.

Jøsang proved that Dempster's rule of combination actually is a method for fusing belief constraints.[8] It only represents an approximate fusion operator in other situations, such as cumulative fusion of beliefs, but generally produces incorrect results in such situations. The confusion around the validity of Dempster's rule therefore originates in the failure of correctly interpreting the nature of situations to be modeled. Dempster's rule of combination always produces correct and intuitive results in situation of fusing belief constraints from different sources.

12.6 See also

- Imprecise probability
- Upper and lower probabilities
- Possibility theory
- Probabilistic logic
- Bayes' theorem
- Bayesian network
- G. L. S. Shackle
- Transferable belief model
- Info-gap decision theory
- Subjective logic
- Doxastic logic
- Linear belief function

12.7 References

[1] Dempster, A. P. (1967). "Upper and lower probabilities induced by a multivalued mapping". *The Annals of Mathematical Statistics* **38** (2): 325–339. doi:10.1214/aoms/1177698950.

[2] Shafer, Glenn; *A Mathematical Theory of Evidence*, Princeton University Press, 1976, ISBN 0-608-02508-9

[3] Fine, Terrence L. (1977). "Review: Glenn Shafer, *A mathematical theory of evidence*". *Bull. Amer. Math. Soc.* **83** (4): 667–672. doi:10.1090/s0002-9904-1977-14338-3.

[4] Kari Sentz and Scott Ferson (2002); *Combination of Evidence in Dempster–Shafer Theory*, Sandia National Laboratories SAND 2002-0835

[5] Kohlas, J., and Monney, P.A., 1995. *A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence*. Vol. 425 in Lecture Notes in Economics and Mathematical Systems. Springer Verlag.

[6] Shafer, Glenn; *Dempster–Shafer theory*, 2002

[7] Dempster, Arthur P.; *A generalization of Bayesian inference*, Journal of the Royal Statistical Society, Series B, Vol. 30, pp. 205–247, 1968

[8] Jøsang, A., and Simon, P. (2012). "Dempster's Rule as Seen by Little Colored Balls". *Computational Intelligence* **28** (4): 453–474. doi:10.1111/j.1467-8640.2012.00421.x.

[9] Jøsang, A., and Hankin, R., 2012. *Interpretation and Fusion of Hyper Opinions in Subjective Logic*. 15th International Conference on Information Fusion (FUSION) 2012. E-ISBN 978-0-9824438-4-2, IEEE.|url=http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6289948

[10] Jøsang, A., Diaz, J., and Rifqi, M. (2010). "Cumulative and averaging fusion of beliefs". *Information Fusion* **11** (2): 192–200. doi:10.1016/j.inffus.2009.05.005.

[11] J.Y. Halpern (2003) *Reasoning about Uncertainty* MIT Press

[12] L. Zadeh, On the validity of Dempster's rule of combination, Memo M79/24, Univ. of California, Berkeley, USA, 1979

[13] L. Zadeh, Book review: A mathematical theory of evidence, The Al Magazine, Vol. 5, No. 3, pp. 81-83, 1984

[14] L. Zadeh, A simple view of the Dempster-Shafer Theory of Evidence and its implication for the rule of combination, The Al Magazine, Vol. 7, No. 2, pp. 85-90, Summer 1986.

[15] E. Ruspini, "The logical foundations of evidential reasoning", *SRI Technical Note* **408**, December 20, 1986 (revised April 27, 1987)

[16] N. Wilson, "The assumptions behind Dempster's rule", in *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 527-−534, Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993

[17] F. Voorbraak, "On the justification of Dempster's rule of combination", *Artificial Intelligence*, Vol. **48**, pp. 171-−197, 1991

[18] Pei Wang, "A Defect in Dempster-Shafer Theory", in *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 560-566, Morgan Kaufmann Publishers, San Mateo, CA, USA, 1994

[19] P. Walley, "Statistical Reasoning with Imprecise Probabilities", Chapman and Hall, London, pp. 278-281, 1991

[20] Pearl, J. (1988a), *Probabilistic Reasoning in Intelligent Systems,* (Revised Second Printing) San Mateo, CA: Morgan Kaufmann.

[21] Pearl, J. (1988b). "On Probability Intervals". *International Journal of Approximate Reasoning* **2** (3): 211–216. doi:10.1016/0888-613X(88)90117-X.

[22] Pearl, J. (1990). "Reasoning with Belief Functions: An Analysis of Compatibility". *The International Journal of Approximate Reasoning* **4** (5/6): 363–389. doi:10.1016/0888-613X(90)90013-R.

[23] M.A. Kłopotek, S.T. Wierzchoń': "A New Qualitative Rough-Set Approach to Modeling Belief Functions." [in:] L. Polkowski, A, Skowron eds: *Rough Sets And Current Trends In Computing. Proc. 1st International Conference RSCTC'98*, Warsaw, June 22–26, 1998, *Lecture Notes in Artificial Intelligence 1424*, Springer-Verlag, pp. 346–353.

[24] M.A. Kłopotek and S.T. Wierzchoń, "Empirical Models for the Dempster–Shafer Theory". in: Srivastava, R.P., Mock, T.J., (Eds.). *Belief Functions in Business Decisions*. Series: *Studies in Fuzziness and Soft Computing*. Vol. **88** Springer-Verlag. March 2002. ISBN 3-7908-1451-2, pp. 62–112

12.8 Further reading

- Yang, J. B. and Xu, D. L. *Evidential Reasoning Rule for Evidence Combination*, Artificial Intelligence, Vol.205, pp. 1–29, 2013.
- Yager, R. R., & Liu, L. (2008). *Classic works of the Dempster–Shafer theory of belief functions.* Studies in fuzziness and soft computing, v. 219. Berlin: Springer. ISBN 978-3-540-25381-5.
- more references
- Joseph C. Giarratano and Gary D. Riley (2005); *Expert Systems: principles and programming*, ed. Thomson Course Tech., ISBN 0-534-38447-1

12.9 External links

- BFAS: Belief Functions and Applications Society

Chapter 13

Dynamic Bayesian network

A **Dynamic Bayesian Network** (DBN) is a Bayesian Network which relates variables to each other over adjacent time steps. This is often called a *Two-Timeslice* BN (2TBN) because it says that at any point in time T, the value of a variable can be calculated from the internal regressors and the immediate prior value (time T-1). DBNs are common in robotics, and have shown potential for a wide range of data mining applications. For example, they have been used in speech recognition, digital forensics, protein sequencing, and bioinformatics. DBN is a generalization of hidden Markov models and Kalman filters.[1]

13.1 See also

- Recursive Bayesian estimation
- Generalized filtering

13.2 References

[1] Stuart Russell; Peter Norvig (2010). *Artificial Intelligence: A Modern Approach* (PDF) (Third ed.). Prentice Hall. p. 566. ISBN 978-0136042594. Retrieved 22 October 2014. **dynamic Bayesian networks** (which include hidden Markov models and Kalman filters as special cases)

- Murphy, Kevin (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. UC Berkeley, Computer Science Division.
- Ghahramani, Zoubin (1997). "Learning Dynamic Bayesian Networks". *Lecture Notes In Computer Science* **1387**: 168–197. CiteSeerX: 10.1.1.56.7874.
- Friedman, N.; Murphy, K.; Russell, S. (1998). *Learning the structure of dynamic probabilistic networks*. UAI'98. Morgan Kaufmann. pp. 139–147. CiteSeerX: 10.1.1.75.2969.

13.3 Software

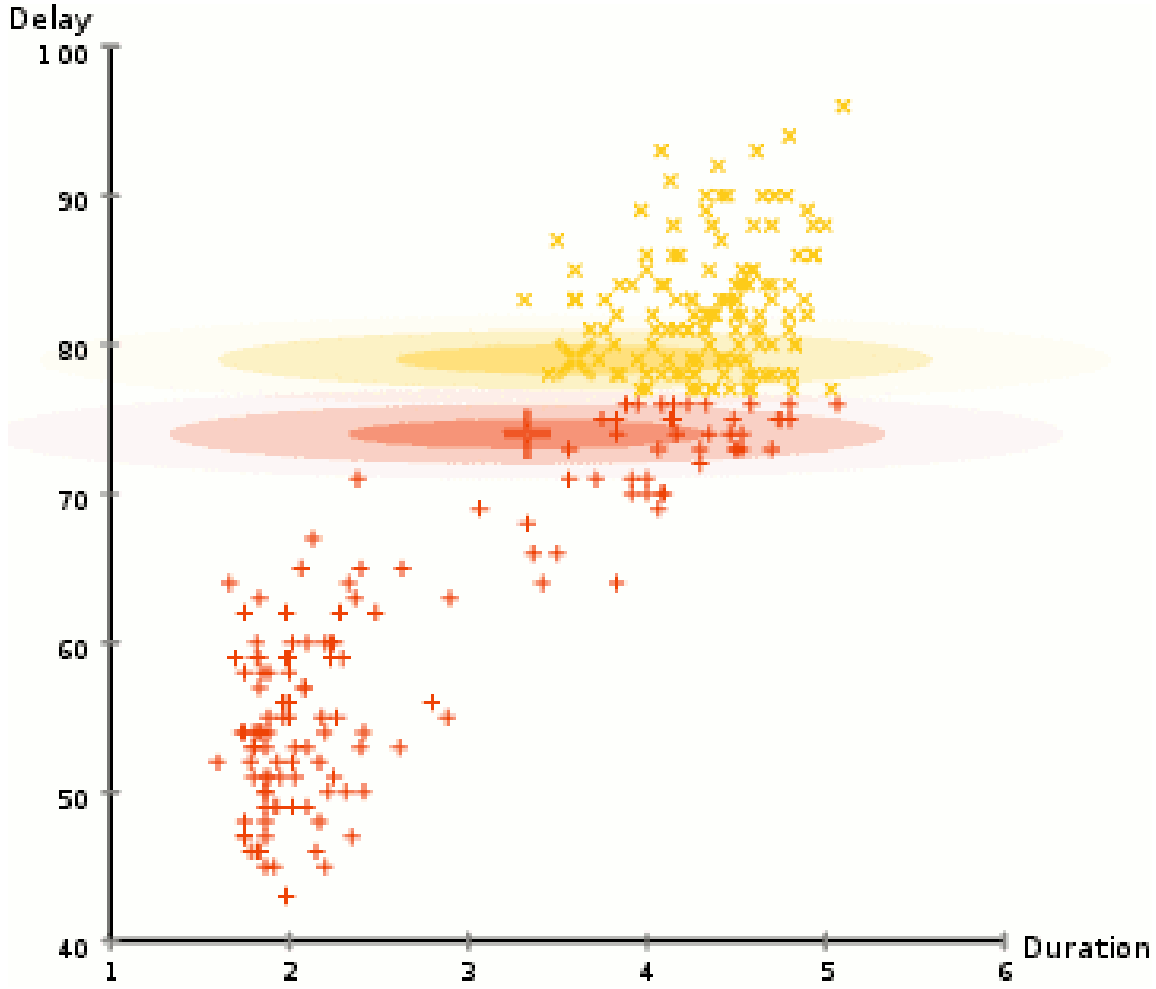
- Dynamic Bayesian network repository at GitHub: the Bayes Net Toolbox for Matlab, by Kevin Murphy, (released under a GPL license)
- Graphical Models Toolkit (GMTK): an open source, publicly available toolkit for rapidly prototyping statistical models using dynamic graphical models (DGMs) and dynamic Bayesian networks (DBNs). GMTK can be used for applications and research in speech and language processing, bioinformatics, activity recognition, and any time series application.
- DBmcmc : Inferring Dynamic Bayesian Networks with MCMC, for Matlab (free software)

- GlobalMIT Matlab toolbox at Google Code: Modeling gene regulatory network via global optimization of dynamic bayesian network (released under a GPL license)
- libDAI: C++ library that provides implementations of various (approximate) inference methods for discrete graphical models; supports arbitrary factor graphs with discrete variables, including discrete Markov Random Fields and Bayesian Networks (released under the FreeBSD license)

Chapter 14

Expectation–maximization algorithm

In statistics, an **expectation–maximization** (**EM**) **algorithm** is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the *E* step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.



EM clustering of Old Faithful eruption data. The random initial model (which, due to the different scales of the axes, appears to be two very flat and wide spheres) is fit to the observed data. In the first iterations, the model changes substantially, but then converges to the two modes of the geyser. Visualized using ELKI.

14.1 History

The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin.[1] They pointed out that the method had been "proposed many times in special circumstances" by earlier authors. In particular, a very detailed treatment of the EM method for exponential families was published by Rolf Sundberg in his thesis and several papers[2][3][4] following his collaboration with Per Martin-Löf and Anders Martin-Löf.[5][6][7][8][9][10][11] The Dempster-Laird-Rubin paper in 1977 generalized the method and sketched a convergence analysis for a wider class of problems. Regardless of earlier inventions, the innovative Dempster-Laird-Rubin paper in the *Journal of the Royal Statistical Society* received an enthusiastic discussion at the Royal Statistical Society meeting with Sundberg calling the paper "brilliant". The Dempster-Laird-Rubin paper established the EM method as an important tool of statistical analysis.

The convergence analysis of the Dempster-Laird-Rubin paper was flawed and a correct convergence analysis was published by C.F. Jeff Wu in 1983.[12] Wu's proof established the EM method's convergence outside of the exponential family, as claimed by Dempster-Laird-Rubin.[13]

14.2 Introduction

The EM algorithm is used to find (locally) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either there are missing values among the data, or the model can be formulated more simply by assuming the existence of additional unobserved data points. For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component that each data point belongs to.

Finding a maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values — viz. the parameters and the latent variables — and simultaneously solving the resulting equations. In statistical models with latent variables, this usually is not possible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice versa, but substituting one set of equations into the other produces an unsolvable equation.

The EM algorithm proceeds from the observation that the following is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work at all, but in fact it can be proven that in this particular context it does, and that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a maximum or a saddle point. In general there may be multiple maxima, and there is no guarantee that the global maximum will be found. Some likelihoods also have singularities in them, i.e. nonsensical maxima. For example, one of the "solutions" that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points.

14.3 Description

Given a statistical model which generates a set $\mathbf{X}$ of observed data, a set of unobserved latent data or missing values $\mathbf{Z}$, and a vector of unknown parameters $\boldsymbol{\theta}$, along with a likelihood function $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$, the maximum likelihood estimate (MLE) of the unknown parameters is determined by the marginal likelihood of the observed data

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

However, this quantity is often intractable (e.g. if $\mathbf{Z}$ is a sequence of events, so that the number of values grows exponentially with the sequence length, making the exact calculation of the sum extremely difficult).

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying the following two steps:

Expectation step (E step): Calculate the expected value of the log likelihood function, with respect to the conditional distribution of $\mathbf{Z}$ given $\mathbf{X}$ under the current estimate of the parameters $\boldsymbol{\theta}^{(t)}$:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}\left[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})\right]$$

Maximization step (M step): Find the parameter that maximizes this quantity:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

Note that in typical models to which EM is applied:

1. The observed data points $\mathbf{X}$ may be discrete (taking values in a finite or countably infinite set) or continuous (taking values in an uncountably infinite set). There may in fact be a vector of observations associated with each data point.
2. The missing values (aka latent variables) $\mathbf{Z}$ are discrete, drawn from a fixed number of values, and there is one latent variable per observed data point.
3. The parameters are continuous, and are of two kinds: Parameters that are associated with all data points, and parameters associated with a particular value of a latent variable (i.e. associated with all data points whose corresponding latent variable has a particular value).

However, it is possible to apply EM to other sorts of models.

The motivation is as follows. If we know the value of the parameters $\boldsymbol{\theta}$, we can usually find the value of the latent variables $\mathbf{Z}$ by maximizing the log-likelihood over all possible values of $\mathbf{Z}$, either simply by iterating over $\mathbf{Z}$ or through an algorithm such as the Viterbi algorithm for hidden Markov models. Conversely, if we know the value of the latent variables $\mathbf{Z}$, we can find an estimate of the parameters $\boldsymbol{\theta}$ fairly easily, typically by simply grouping the observed data points according to the value of the associated latent variable and averaging the values, or some function of the values, of the points in each group. This suggests an iterative algorithm, in the case where both $\boldsymbol{\theta}$ and $\mathbf{Z}$ are unknown:

1. First, initialize the parameters $\boldsymbol{\theta}$ to some random values.
2. Compute the best value for $\mathbf{Z}$ given these parameter values.
3. Then, use the just-computed values of $\mathbf{Z}$ to compute a better estimate for the parameters $\boldsymbol{\theta}$. Parameters associated with a particular value of $\mathbf{Z}$ will use only those data points whose associated latent variable has that value.
4. Iterate steps 2 and 3 until convergence.

The algorithm as just described monotonically approaches a local minimum of the cost function, and is commonly called *hard EM*. The *k*-means algorithm is an example of this class of algorithms.

However, one can do somewhat better: Rather than making a hard choice for $\mathbf{Z}$ given the current parameter values and averaging only over the set of data points associated with a particular value of $\mathbf{Z}$, one can instead determine the probability of each possible value of $\mathbf{Z}$ for each data point, and then use the probabilities associated with a particular value of $\mathbf{Z}$ to compute a weighted average over the entire set of data points. The resulting algorithm is commonly called *soft EM*, and is the type of algorithm normally associated with EM. The counts used to compute these weighted averages are called *soft counts* (as opposed to the *hard counts* used in a hard-EM-type algorithm such as *k*-means). The probabilities computed for $\mathbf{Z}$ are posterior probabilities and are what is computed in the E step. The soft counts used to compute new parameter values are what is computed in the M step.

14.4 Properties

Speaking of an expectation (E) step is a bit of a misnomer. What is calculated in the first step are the fixed, data-dependent parameters of the function Q. Once the parameters of Q are known, it is fully determined and is maximized in the second (M) step of an EM algorithm.

Although an EM iteration does increase the observed data (i.e. marginal) likelihood function there is no guarantee that the sequence converges to a maximum likelihood estimator. For multimodal distributions, this means that an EM algorithm may converge to a local maximum of the observed data likelihood function, depending on starting values. There are a variety of heuristic or metaheuristic approaches for escaping a local maximum such as random restart (starting with several different random initial estimates $\theta^{(t)}$), or applying simulated annealing methods.

EM is particularly useful when the likelihood is an exponential family: the E step becomes the sum of expectations of sufficient statistics, and the M step involves maximizing a linear function. In such a case, it is usually possible to derive closed form updates for each step, using the Sundberg formula (published by Rolf Sundberg using unpublished results of Per Martin-Löf and Anders Martin-Löf).[3][4][7][8][9][10][11]

The EM method was modified to compute maximum a posteriori (MAP) estimates for Bayesian inference in the original paper by Dempster, Laird, and Rubin.

There are other methods for finding maximum likelihood estimates, such as gradient descent, conjugate gradient or variations of the Gauss–Newton method. Unlike EM, such methods typically require the evaluation of first and/or second derivatives of the likelihood function.

14.5 Proof of correctness

Expectation-maximization works to improve $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ rather than directly improving $\log p(\mathbf{X}|\boldsymbol{\theta})$. Here we show that improvements to the former imply improvements to the latter.[14]

For any $\mathbf{Z}$ with non-zero probability $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$, we can write

$$\log p(\mathbf{X}|\boldsymbol{\theta}) = \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \,.$$

We take the expectation over values of $\mathbf{Z}$ by multiplying both sides by $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)})$ and summing (or integrating) over $\mathbf{Z}$. The left-hand side is the expectation of a constant, so we get:

$$\begin{aligned} \log p(\mathbf{X}|\boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \\ &= Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \,, \end{aligned}$$

where $H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ is defined by the negated sum it is replacing. This last equation holds for any value of $\boldsymbol{\theta}$ including $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$,

$$\log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \,,$$

and subtracting this last equation from the previous equation gives

$$\log p(\mathbf{X}|\boldsymbol{\theta}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \,,$$

However, Gibbs' inequality tells us that $H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \geq H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)})$, so we can conclude that

$$\log p(\mathbf{X}|\boldsymbol{\theta}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \,.$$

In words, choosing $\boldsymbol{\theta}$ to improve $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ beyond $Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)})$ will improve $\log p(\mathbf{X}|\boldsymbol{\theta})$ beyond $\log p(\mathbf{X}|\boldsymbol{\theta}^{(t)})$ at least as much.

14.6 Alternative description

Under some circumstances, it is convenient to view the EM algorithm as two alternating maximization steps.[15][16] Consider the function:

$$F(q,\theta) = \mathrm{E}_q[\log L(\theta; x, Z)] + H(q) = -D_{\mathrm{KL}}\big(q\big\|p_{Z|X}(\cdot|x;\theta)\big) + \log L(\theta; x)$$

where q is an arbitrary probability distribution over the unobserved data z, $pZ|X(\cdot\,|x;\theta)$ is the conditional distribution of the unobserved data given the observed data x, H is the entropy and DKL is the Kullback–Leibler divergence.

Then the steps in the EM algorithm may be viewed as:

Expectation step: Choose q to maximize F:

$q^{(t)} = *\,arg\,max_q\ F(q, \theta^{(t)})$

Maximization step: Choose θ to maximize F:

$\theta^{(t+1)} = *\,arg\,max_\theta\ F(q^{(t)}, \theta)$

14.7 Applications

EM is frequently used for data clustering in machine learning and computer vision. In natural language processing, two prominent instances of the algorithm are the Baum-Welch algorithm and the inside-outside algorithm for unsupervised induction of probabilistic context-free grammars.

In psychometrics, EM is almost indispensable for estimating item parameters and latent abilities of item response theory models.

With the ability to deal with missing data and observe unidentified variables, EM is becoming a useful tool to price and manage risk of a portfolio.[ref?]

The EM algorithm (and its faster variant Ordered subset expectation maximization) is also widely used in medical image reconstruction, especially in positron emission tomography and single photon emission computed tomography. See below for other faster variants of EM.

14.8 Filtering and smoothing EM algorithms

A Kalman filter is typically used for on-line state estimation and a minimum-variance smoother may be employed for off-line or batch state estimation. However, these minimum-variance solutions require estimates of the state-space model parameters. EM algorithms can be used for solving joint state and parameter estimation problems.

Filtering and smoothing EM algorithms arise by repeating the following two-step procedure:

E-step Operate a Kalman filter or a minimum-variance smoother designed with current parameter estimates to obtain updated state estimates.

M-step Use the filtered or smoothed state estimates within maximum-likelihood calculations to obtain updated parameter estimates.

Suppose that a Kalman filter or minimum-variance smoother operates on noisy measurements of a single-input-single-output system. An updated measurement noise variance estimate can be obtained from the maximum likelihood calculation

$$\hat{\sigma}_v^2 = \frac{1}{N}\sum_{k=1}^{N}(z_k - \hat{x}_k)^2$$

where $\hat{x}_k$ are scalar output estimates calculated by a filter or a smoother from N scalar measurements z_k . Similarly, for a first-order auto-regressive process, an updated process noise variance estimate can be calculated by

$$\hat{\sigma}_w^2 = \frac{1}{N}\sum_{k=1}^{N}\left(\hat{x}_{k+1} - \hat{F}\hat{x}_k\right)^2$$

where $\hat{x}_k$ and $\hat{x}_{k+1}$ are scalar state estimates calculated by a filter or a smoother. The updated model coefficient estimate is obtained via

$$\hat{F} = \frac{\sum_{k=1}^{N}(\hat{x}_{k+1} - \hat{F}\hat{x}_k)}{\sum_{k=1}^{N}\hat{x}_k^2}$$

The convergence of parameter estimates such as those above are well studied.[17][18][19]

14.9 Variants

A number of methods have been proposed to accelerate the sometimes slow convergence of the EM algorithm, such as those using conjugate gradient and modified Newton–Raphson techniques.[20] Additionally EM can be used with constrained estimation techniques.

Expectation conditional maximization (ECM) replaces each M step with a sequence of conditional maximization (CM) steps in which each parameter θi is maximized individually, conditionally on the other parameters remaining fixed.[21]

This idea is further extended in **generalized expectation maximization (GEM)** algorithm, in which one only seeks an increase in the objective function F for both the E step and M step under the alternative description.[15]

It is also possible to consider the EM algorithm as a subclass of the **MM** (Majorize/Minimize or Minorize/Maximize, depending on context) algorithm,[22] and therefore use any machinery developed in the more general case.

14.9.1 α-EM algorithm

The Q-function used in the EM algorithm is based on the log likelihood. Therefore, it is regarded as the log-EM algorithm. The use of the log likelihood can be generalized to that of the α-log likelihood ratio. Then, the α-log likelihood ratio of the observed data can be exactly expressed as equality by using the Q-function of the α-log likelihood ratio and the α-divergence. Obtaining this Q-function is a generalized E step. Its maximization is a generalized M step. This pair is called the α-EM algorithm [23] which contains the log-EM algorithm as its subclass. Thus, the α-EM algorithm by Yasuo Matsuyama is an exact generalization of the log-EM algorithm. No computation of gradient or Hessian matrix is needed. The α-EM shows faster convergence than the log-EM algorithm by choosing an appropriate α. The α-EM algorithm leads to a faster version of the Hidden Markov model estimation algorithm α-HMM. [24]

14.10 Relation to variational Bayes methods

EM is a partially non-Bayesian, maximum likelihood method. Its final result gives a probability distribution over the latent variables (in the Bayesian style) together with a point estimate for θ (either a maximum likelihood estimate or a posterior mode). We may want a fully Bayesian version of this, giving a probability distribution over θ as well as the latent variables. In fact the Bayesian approach to inference is simply to treat θ as another latent variable. In this paradigm, the distinction between the E and M steps disappears. If we use the factorized Q approximation as described above (variational Bayes), we may iterate over each latent variable (now including θ) and optimize them one at a time. There are now k steps per iteration, where k is the number of latent variables. For graphical models this is easy to do as each variable's new Q depends only on its Markov blanket, so local message passing can be used for efficient inference.

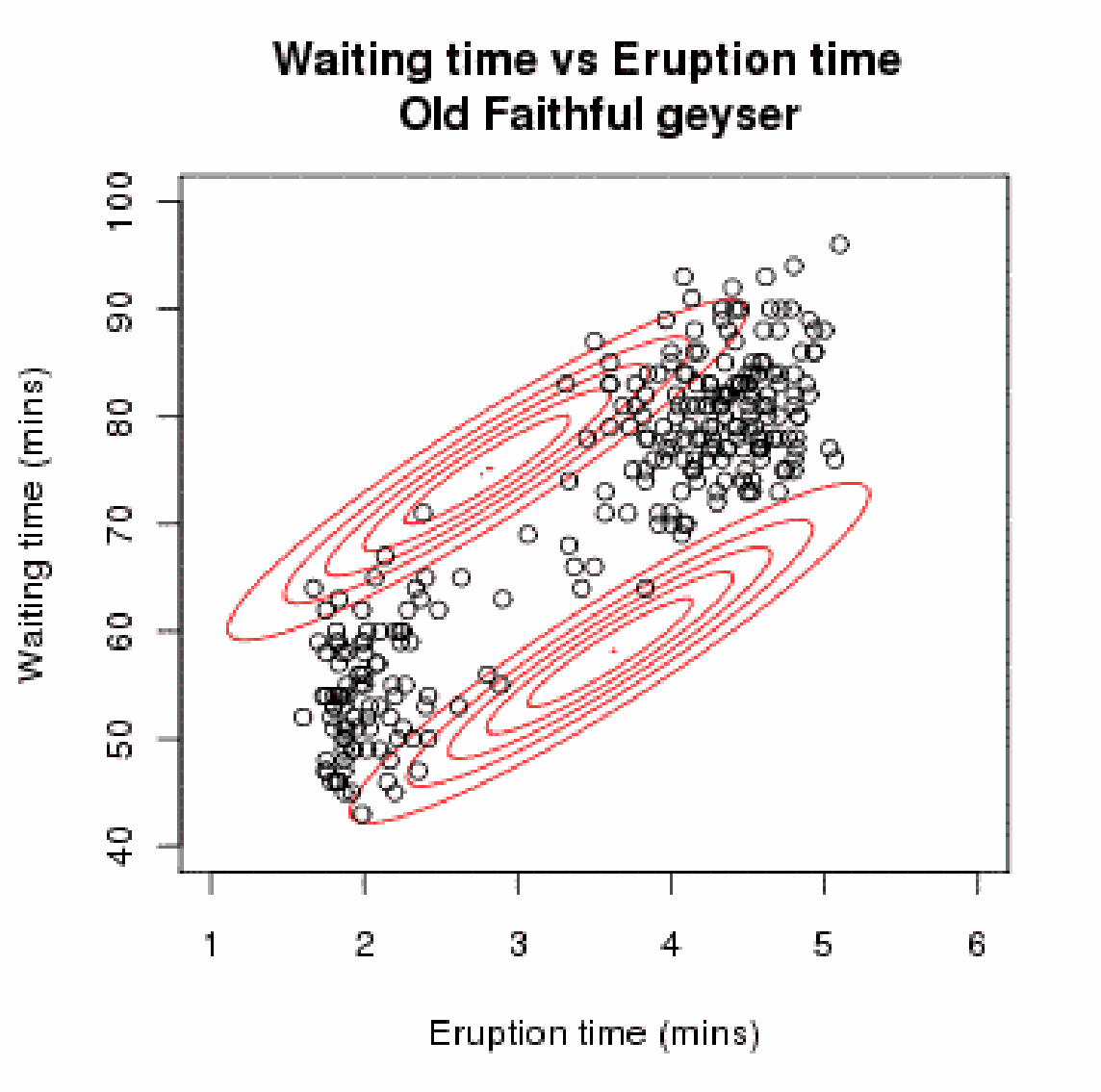
14.11 Geometric interpretation

For more details on this topic, see Information geometry.

In information geometry, the E step and the M step are interpreted as projections under dual affine connections, called the e-connection and the m-connection; the Kullback–Leibler divergence can also be understood in these terms.

14.12 Examples

14.12.1 Gaussian mixture



An animation demonstrating the EM algorithm fitting a two component Gaussian mixture model to the Old Faithful dataset. The algorithm steps through from a random initialization to convergence.

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be a sample of n independent observations from a mixture of two multivariate normal distributions of dimension d , and let $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ be the latent variables that determine the component from which the observation originates.[16]

$$X_i|(Z_i = 1) \sim \mathcal{N}_d(\boldsymbol{\mu}_1, \Sigma_1) \text{ and } X_i|(Z_i = 2) \sim \mathcal{N}_d(\boldsymbol{\mu}_2, \Sigma_2)$$

where

$$\mathrm{P}(Z_i = 1) = \tau_1 \text{ and } \mathrm{P}(Z_i = 2) = \tau_2 = 1 - \tau_1$$

The aim is to estimate the unknown parameters representing the "mixing" value between the Gaussians and the means and covariances of each:

$$\theta = \left(\boldsymbol{\tau}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2\right)$$

where the incomplete-data likelihood function is

$$L(\theta; \mathbf{x}) = \prod_{i=1}^{n} \sum_{j=1}^{2} \tau_j \, f(\mathbf{x}_i; \boldsymbol{\mu}_j, \Sigma_j)$$

and the complete-data likelihood function is

$$L(\theta; \mathbf{x}, \mathbf{z}) = P(\mathbf{x}, \mathbf{z}|\theta) = \prod_{i=1}^{n} \sum_{j=1}^{2} \mathbb{I}(z_i = j) \, f(\mathbf{x}_i; \boldsymbol{\mu}_j, \Sigma_j)\tau_j$$

or

$$L(\theta; \mathbf{x}, \mathbf{z}) = \exp\left\{ \sum_{i=1}^{n} \sum_{j=1}^{2} \mathbb{I}(z_i = j) \big[-\tfrac{1}{2} \log|\Sigma_j| - \tfrac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) - \tfrac{d}{2} \log(2\pi) \big] \right\}.$$

where $\mathbb{I}$ is an indicator function and f is the probability density function of a multivariate normal.

To see the last equality, note that for each *i* all indicators $\mathbb{I}(z_i = j)$ are equal to zero, except for one which is equal to one. The inner sum thus reduces to a single term.

E step

Given our current estimate of the parameters $\theta^{(t)}$, the conditional distribution of the *Zi* is determined by Bayes theorem to be the proportional height of the normal density weighted by τ:

$$T_{j,i}^{(t)} := \mathrm{P}(Z_i = j | X_i = \mathbf{x}_i; \theta^{(t)}) = \frac{\tau_j^{(t)} \, f(\mathbf{x}_i; \boldsymbol{\mu}_j^{(t)}, \Sigma_j^{(t)})}{\tau_1^{(t)} \, f(\mathbf{x}_i; \boldsymbol{\mu}_1^{(t)}, \Sigma_1^{(t)}) + \tau_2^{(t)} \, f(\mathbf{x}_i; \boldsymbol{\mu}_2^{(t)}, \Sigma_2^{(t)})}$$

These are called the "membership probabilities" which are normally considered the output of the E step (although this is not the Q function of below).

Note that this E step corresponds with the following function for Q:

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \mathrm{E}[\log L(\theta; \mathbf{x}, \mathbf{Z})] \\
&= \mathrm{E}[\log \prod_{i=1}^{n} L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \mathrm{E}[\sum_{i=1}^{n} \log L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \sum_{i=1}^{n} \mathrm{E}[\log L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{2} T_{j,i}^{(t)} \big[\log \tau_j - \tfrac{1}{2} \log|\Sigma_j| - \tfrac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) - \tfrac{d}{2} \log(2\pi) \big]
\end{aligned}$$

This does not need to be calculated, because in the M step we only require the terms depending on τ when we maximize for τ, or only the terms depending on $\boldsymbol{\mu}$ if we maximize for $\boldsymbol{\mu}$.

M step

The fact that $Q(\theta|\theta^{(t)})$ is quadratic in form means that determining the maximizing values of θ is relatively straightforward. Note that τ, ($\boldsymbol{\mu}_1$,Σ_1) and ($\boldsymbol{\mu}_2$,Σ_2) may all be maximized independently since they all appear in separate linear terms.

To begin, consider τ, which has the constraint τ_1 + τ_2=1:

$$\begin{aligned}\boldsymbol{\tau}^{(t+1)} &= \arg\max_{\boldsymbol{\tau}} Q(\theta|\theta^{(t)}) \\ &= \arg\max_{\boldsymbol{\tau}} \left\{ \left[\sum_{i=1}^{n} T_{1,i}^{(t)}\right] \log \tau_1 + \left[\sum_{i=1}^{n} T_{2,i}^{(t)}\right] \log \tau_2 \right\}\end{aligned}$$

This has the same form as the MLE for the binomial distribution, so

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{j,i}^{(t)}}{\sum_{i=1}^{n} (T_{1,i}^{(t)} + T_{2,i}^{(t)})} = \frac{1}{n}\sum_{i=1}^{n} T_{j,i}^{(t)}$$

For the next estimates of ($\boldsymbol{\mu}_1$,σ_1):

$$\begin{aligned}(\boldsymbol{\mu}_1^{(t+1)}, \Sigma_1^{(t+1)}) &= \arg\max_{\boldsymbol{\mu}_1,\Sigma_1} Q(\theta|\theta^{(t)}) \\ &= \arg\max_{\boldsymbol{\mu}_1,\Sigma_1} \sum_{i=1}^{n} T_{1,i}^{(t)} \left\{-\tfrac{1}{2}\log|\Sigma_1| - \tfrac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_1)^\top \Sigma_1^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_1)\right\}\end{aligned}$$

This has the same form as a weighted MLE for a normal distribution, so

$$\boldsymbol{\mu}_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{n} T_{1,i}^{(t)}} \text{ and } \Sigma_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})^\top}{\sum_{i=1}^{n} T_{1,i}^{(t)}}$$

and, by symmetry

$$\boldsymbol{\mu}_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{n} T_{2,i}^{(t)}} \text{ and } \Sigma_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})^\top}{\sum_{i=1}^{n} T_{2,i}^{(t)}} .$$

Termination

Conclude the iterative process if $\log L(\theta^t; \mathbf{x}, \mathbf{Z}) \leq \log L(\theta^{(t-1)}; \mathbf{x}, \mathbf{Z}) + \epsilon$ for ϵ below some preset threshold.

Generalization

The algorithm illustrated above can be generalized for mixtures of more than two multivariate normal distributions.

14.12.2 Truncated and censored regression

The EM algorithm has been implemented in the case where there is an underlying linear regression model explaining the variation of some quantity, but where the values actually observed are censored or truncated versions of those represented in the model.[25] Special cases of this model include censored or truncated observations from a single normal distribution.[25]

14.13 Alternatives to EM

EM typically converges to a local optimum--not necessarily the global optimum--and there is no bound on the convergence rate in general. It is possible that it can be arbitrarily poor in high dimensions and there can be an exponential number of local optima. Hence, there is a need for alternative techniques for guaranteed learning, especially in the high-dimensional setting. There are alternatives to EM with better guarantees in terms of consistency which are known as moment-based approaches or the so-called "spectral techniques". Moment-based approaches[26] to learning the parameters of a probabilistic model are of increasing interest recently since they enjoy guarantees such as global convergence under certain conditions unlike EM which is often plagued by the issue of getting stuck in local optima. Algorithms with guarantees for learning can be derived for a number of important models such as mixture models, Hidden Markov models [27] and community models.[28] For these spectral methods, there are no spurious local optima and the true parameters can be consistently estimated under some regularity conditions.

14.14 See also

- Density estimation
- Total absorption spectroscopy
- The EM algorithm can be viewed as a special case of the majorize-minimization (MM) algorithm.[29]

14.15 Further reading

- Robert Hogg, Joseph McKean and Allen Craig. *Introduction to Mathematical Statistics*. pp. 359–364. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.
- The on-line textbook: Information Theory, Inference, and Learning Algorithms, by David J.C. MacKay includes simple examples of the EM algorithm such as clustering using the soft *k*-means algorithm, and emphasizes the variational view of the EM algorithm, as described in Chapter 33.7 of version 7.2 (fourth edition).
- Dellaert, Frank. "The Expectation Maximization Algorithm". CiteSeerX: 10.1.1.9.9735, gives an easier explanation of EM algorithm in terms of lowerbound maximization.
- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer. ISBN 0-387-31073-8.
- M. R. Gupta and Y. Chen (2010). *Theory and Use of the EM Algorithm*. doi:10.1561/2000000034. A well-written short book on EM, including detailed derivation of EM for GMMs, HMMs, and Dirichlet.
- Bilmes, Jeff. "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models". CiteSeerX: 10.1.1.28.613, includes a simplified derivation of the EM equations for Gaussian Mixtures and Gaussian Mixture Hidden Markov Models.
- Variational Algorithms for Approximate Bayesian Inference, by M. J. Beal includes comparisons of EM to Variational Bayesian EM and derivations of several models including Variational Bayesian HMMs (chapters).
- The Expectation Maximization Algorithm: A short tutorial, A self-contained derivation of the EM Algorithm by Sean Borman.
- The EM Algorithm, by Xiaojin Zhu.
- EM algorithm and variants: an informal tutorial by Alexis Roche. A concise and very clear description of EM and many interesting variants.
- Einicke, G.A. (2012). *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*. Rijeka, Croatia: Intech. ISBN 978-953-307-752-9.

14.16 References

[1] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society, Series B* **39** (1): 1–38. JSTOR 2984875. MR 0501537.

[2] Sundberg, Rolf (1974). "Maximum likelihood theory for incomplete data from an exponential family". *Scandinavian Journal of Statistics* **1** (2): 49–58. JSTOR 4615553. MR 381110.

[3] Rolf Sundberg. 1971. *Maximum likelihood theory and applications for distributions generated when observing a function of an exponential family variable*. Dissertation, Institute for Mathematical Statistics, Stockholm University.

[4] Sundberg, Rolf (1976). "An iterative method for solution of the likelihood equations for incomplete data from exponential families". *Communications in Statistics – Simulation and Computation* **5** (1): 55–64. doi:10.1080/03610917608812007. MR 443190.

[5] See the acknowledgement by Dempster, Laird and Rubin on pages 3, 5 and 11.

[6] G. Kulldorff. 1961. *Contributions to the theory of estimation from grouped and partially grouped samples*. Almqvist & Wiksell.

[7] Anders Martin-Löf. 1963. "Utvärdering av livslängder i subnanosekundsområdet" ("Evaluation of sub-nanosecond lifetimes"). ("Sundberg formula")

[8] Per Martin-Löf. 1966. *Statistics from the point of view of statistical mechanics*. Lecture notes, Mathematical Institute, Aarhus University. ("Sundberg formula" credited to Anders Martin-Löf).

[9] Per Martin-Löf. 1970. *Statistika Modeller (Statistical Models): Anteckningar från seminarier läsåret 1969–1970 (Notes from seminars in the academic year 1969-1970), with the assistance of Rolf Sundberg*. Stockholm University. ("Sundberg formula")

[10] Martin-Löf, P. The notion of redundancy and its use as a quantitative measure of the deviation between a statistical hypothesis and a set of observational data. With a discussion by F. Abildgård, A. P. Dempster, D. Basu, D. R. Cox, A. W. F. Edwards, D. A. Sprott, G. A. Barnard, O. Barndorff-Nielsen, J. D. Kalbfleisch and G. Rasch and a reply by the author. *Proceedings of Conference on Foundational Questions in Statistical Inference* (Aarhus, 1973), pp. 1–42. Memoirs, No. 1, Dept. Theoret. Statist., Inst. Math., Univ. Aarhus, Aarhus, 1974.

[11] Martin-Löf, Per The notion of redundancy and its use as a quantitative measure of the discrepancy between a statistical hypothesis and a set of observational data. *Scand. J. Statist.* 1 (1974), no. 1, 3–18.

[12] Wu, C. F. Jeff (1983). "On the Convergence Properties of the EM Algorithm". *The Annals of Statistics* (Institute of Mathematical Statistics) **11** (1): 95–103. Retrieved 11 December 2014.

[13] Wu, C. F. Jeff (Mar 1983). "On the Convergence Properties of the EM Algorithm". *Annals of Statistics* **11** (1): 95–103. doi:10.1214/aos/1176346060. JSTOR 2240463. MR 684867.

[14] Little, Roderick J.A.; Rubin, Donald B. (1987). *Statistical Analysis with Missing Data*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons. pp. 134–136. ISBN 0-471-80254-9.

[15] Neal, Radford; Hinton, Geoffrey (1999). Michael I. Jordan, ed. "A view of the EM algorithm that justifies incremental, sparse, and other variants" (PDF). *Learning in Graphical Models* (Cambridge, MA: MIT Press): 355–368. ISBN 0-262-60032-3. Retrieved 2009-03-22.

[16] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2001). "8.5 The EM algorithm". *The Elements of Statistical Learning*. New York: Springer. pp. 236–243. ISBN 0-387-95284-5.

[17] Einicke, G.A.; Malos, J.T.; Reid, D.C.; Hainsworth, D.W. (January 2009). "Riccati Equation and EM Algorithm Convergence for Inertial Navigation Alignment". *IEEE Trans. Signal Processing* **57** (1): 370–375. doi:10.1109/TSP.2008.2007090.

[18] Einicke, G.A.; Falco, G.; Malos, J.T. (May 2010). "EM Algorithm State Matrix Estimation for Navigation". *IEEE Signal Processing Letters* **17** (5): 437–440. Bibcode:2010ISPL...17..437E. doi:10.1109/LSP.2010.2043151.

[19] Einicke, G.A.; Falco, G.; Dunn, M.T.; Reid, D.C. (May 2012). "Iterative Smoother-Based Variance Estimation". *IEEE Signal Processing Letters* **19** (5): 275–278. Bibcode:2012ISPL...19..275E. doi:10.1109/LSP.2012.2190278.

[20] Jamshidian, Mortaza; Jennrich, Robert I. (1997). "Acceleration of the EM Algorithm by using Quasi-Newton Methods". *Journal of the Royal Statistical Society, Series B* **59** (2): 569–587. doi:10.1111/1467-9868.00083. MR 1452026.

[21] Meng, Xiao-Li; Rubin, Donald B. (1993). "Maximum likelihood estimation via the ECM algorithm: A general framework". *Biometrika* **80** (2): 267–278. doi:10.1093/biomet/80.2.267. MR 1243503.

[22] Hunter DR and Lange K (2004), A Tutorial on MM Algorithms, The American Statistician, 58: 30-37

[23] Matsuyama, Yasuo (2003). "The α-EM algorithm: Surrogate likelihood maximization using α-logarithmic information measures". *IEEE Transactions on Information Theory* **49** (3): 692–706. doi:10.1109/TIT.2002.808105.

[24] Matsuyama, Yasuo (2011). "Hidden Markov model estimation based on alpha-EM algorithm: Discrete and continuous alpha-HMMs". *International Joint Conference on Neural Networks*: 808–816.

[25] Wolynetz, M.S. (1979). "Maximum likelihood estimation in a linear model from confined and censored normal data". *Journal of the Royal Statistical Society, Series C* **28** (2): 195–206.

[26] Anandkumar, Animashree; Ge, Rong; Hsu, Daniel; Kakade, Sham M; Telgarsky, Matus (2014). "Tensor decompositions for learning latent variable models". *The Journal of Machine Learning Research* **15** (1): 2773–2832.

[27] Anandkumar, Animashree; Hsu, Daniel; Kakade, Sham M (2012). "A method of moments for mixture models and hidden Markov models". *arXiv preprint arXiv:1203.0683*.

[28] Anandkumar, Animashree; Ge, Rong; Hsu, Daniel; Kakade, Sham M (2014). "A tensor approach to learning mixed membership community models". *The Journal of Machine Learning Research* **15** (1): 2239–2312.

[29] Lange, Kenneth. "The MM Algorithm" (PDF).

14.17 External links

- Various 1D, 2D and 3D demonstrations of EM together with Mixture Modeling are provided as part of the paired SOCR activities and applets. These applets and activities show empirically the properties of the EM algorithm for parameter estimation in diverse settings.
- k-MLE: A fast algorithm for learning statistical mixture models
- Class hierarchy in C++ (GPL) including Gaussian Mixtures
- Fast and clean C implementation of the Expectation Maximization (EM) algorithm for estimating Gaussian Mixture Models (GMMs).

Chapter 15

Factor graph

Not to be confused with Graph factorization.

A **factor graph** is a bipartite graph representing the factorization of a function. In probability theory and its applications, factor graphs are used to represent factorization of a probability distribution function, enabling efficient computations, such as the computation of marginal distributions through the sum-product algorithm. One of the important success stories of factor graphs and the sum-product algorithm is the decoding of capacity-approaching error-correcting codes, such as LDPC and turbo codes.

Factor graphs generalize constraint graphs. A factor whose value is either 0 or 1 is called a constraint. A constraint graph is a factor graph where all factors are constraints. The max-product algorithm for factor graphs can be viewed as a generalization of the arc-consistency algorithm for constraint processing.

15.1 Definition

A factor graph is a bipartite graph representing the factorization of a function. Given a factorization of a function $g(X_1, X_2, \ldots, X_n)$,

$$g(X_1, X_2, \ldots, X_n) = \prod_{j=1}^{m} f_j(S_j),$$

where $S_j \subseteq \{X_1, X_2, \ldots, X_n\}$, the corresponding factor graph $G = (X, F, E)$ consists of variable vertices $X = \{X_1, X_2, \ldots, X_n\}$, factor vertices $F = \{f_1, f_2, \ldots, f_m\}$, and edges E . The edges depend on the factorization as follows: there is an undirected edge between factor vertex f_j and variable vertex X_k iff $X_k \in S_j$. The function is tacitly assumed to be real-valued: $g(X_1, X_2, \ldots, X_n) \in \mathbb{R}$.

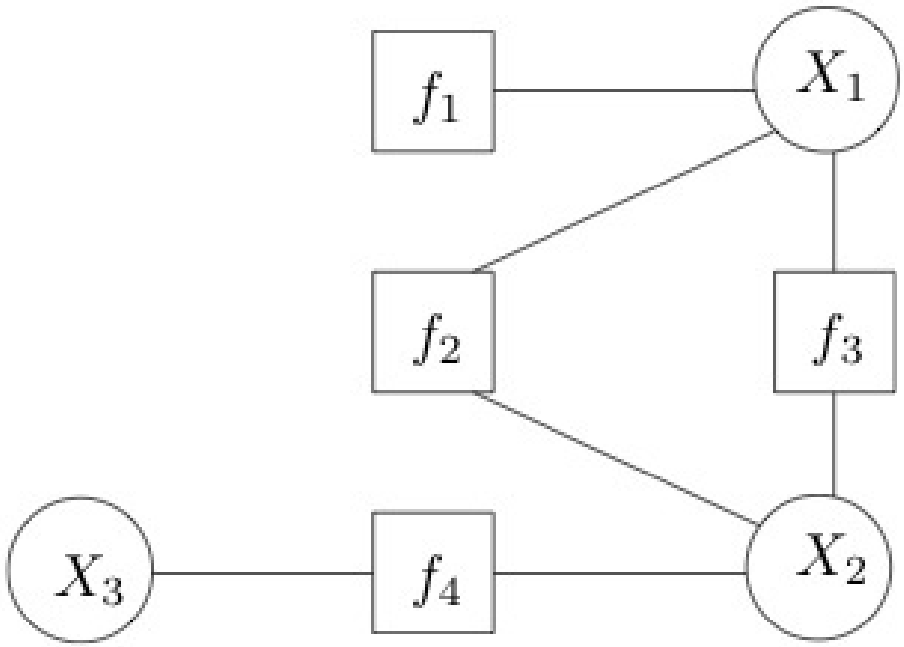
Factor graphs can be combined with message passing algorithms to efficiently compute certain characteristics of the function $g(X_1, X_2, \ldots, X_n)$, such as the marginal distributions.

15.2 Examples

Consider a function that factorizes as follows:

$$g(X_1, X_2, X_3) = f_1(X_1) f_2(X_1, X_2) f_3(X_1, X_2) f_4(X_2, X_3)$$

with a corresponding factor graph shown on the right. Observe that the factor graph has a cycle. If we merge $f_2(X_1, X_2) f_3(X_1, X_2)$ into a single factor, the resulting factor graph will be a tree. This is an important distinction, as message passing algorithms are usually exact for trees, but only approximate for graphs with cycles.



An example factor graph

15.3 Message passing on factor graphs

A popular message passing algorithm on factor graphs is the sum-product algorithm, which efficiently computes all the marginals of the individual variables of the function. In particular, the marginal of variable X_k is defined as

$$g_k(X_k) = \sum_{X_{\bar{k}}} g(X_1, X_2, \ldots, X_n)$$

where the notation $X_{\bar{k}}$ means that the summation goes over all the variables, *except* X_k . The messages of the sum-product algorithm are conceptually computed in the vertices and passed along the edges. A message from or to a variable vertex is always a function of that particular variable. For instance, when a variable is binary, the messages over the edges incident to the corresponding vertex can be represented as vectors of length 2: the first entry is the message evaluated in 0, the second entry is the message evaluated in 1. When a variable belongs to the field of real numbers, messages can be arbitrary functions, and special care needs to be taken in their representation.

In practice, the sum-product algorithm is used for statistical inference, whereby $g(X_1, X_2, \ldots, X_n)$ is a joint distribution or a joint likelihood function, and the factorization depends on the conditional independencies among the variables.

The Hammersley–Clifford theorem shows that other probabilistic models such as Markov networks and Bayesian networks can be represented as factor graphs; the latter representation is frequently used when performing inference over such networks using belief propagation. On the other hand, Bayesian networks are more naturally suited for generative models, as they can directly represent the causalities of the model.

15.4 See also

- Belief propagation
- Bayesian inference
- Bayesian programming
- Conditional probability

- Markov network
- Bayesian network
- Hammersley–Clifford theorem

15.5 External links

- A tutorial-style dissertation by Volker Koch
- An Introduction to Factor Graphs by Hans-Andrea Loeliger, *IEEE Signal Processing Magazine,* January 2004, pp. 28–41.
- dimple an open-source tool for building and solving factor graphs in MATLAB.
- An introduction to Factor Graph. Presentation from the ETH by Prof. Dr. Hans Loeliger

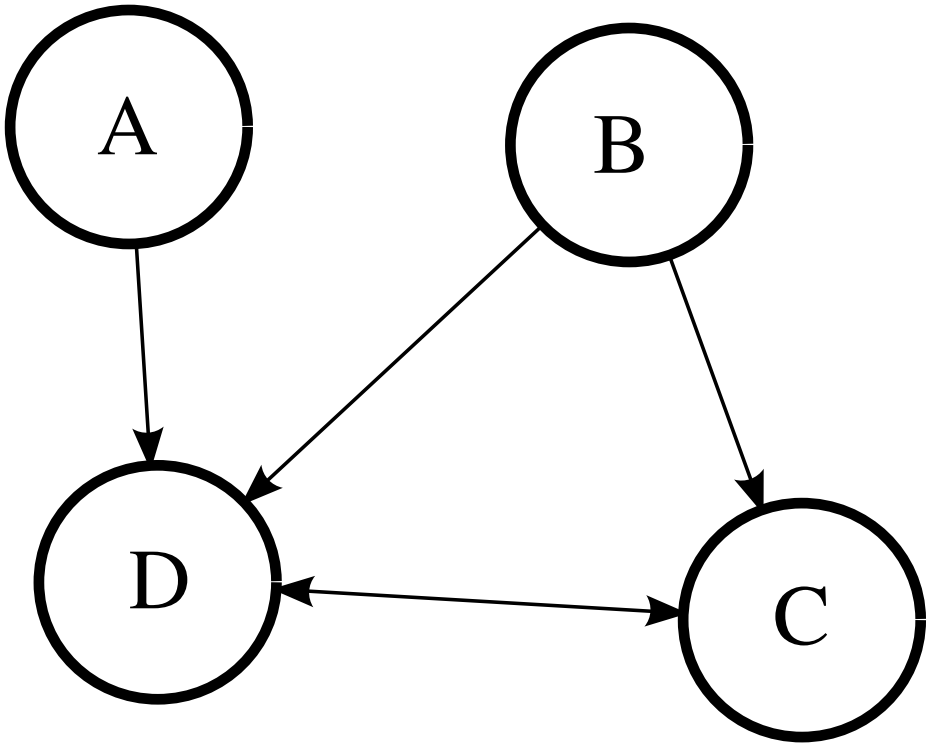
15.6 References

- Clifford (1990), "Markov random fields in statistics", in Grimmett, G.R.; Welsh, D.J.A., *Disorder in Physical Systems, J.M. Hammersley Festschrift*, Oxford University Press, pp. 19–32
- Frey, Brendan J. (2003), "Extending Factor Graphs so as to Unify Directed and Undirected Graphical Models", in jain, Nitin, *UAI'03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, August 7–10, Acapulco, Mexico*, Morgan Kaufmann, pp. 257–264
- Kschischang, Frank R.; Frey, Brendan J.; Loeliger, Hans-Andrea (2001), "Factor Graphs and the Sum-Product Algorithm", *IEEE Transactions on Information Theory* **47** (2): 498–519, doi:10.1109/18.910572, retrieved 2008-02-06.
- Wymeersch, Henk (2007), *Iterative Receiver Design*, Cambridge University Press, ISBN 0-521-87315-0

Chapter 16

Graphical model

A **graphical model** or **probabilistic graphical model** (**PGM**) is a probabilistic model for which a graph expresses the conditional dependence structure between random variables. They are commonly used in probability theory, statistics—particularly Bayesian statistics—and machine learning.



An example of a graphical model. Each arrow indicates a dependency. In this example: D depends on A, D depends on B, D depends on C, C depends on B, and C depends on D.

16.1 Types of graphical models

Generally, probabilistic graphical models use a graph-based representation as the foundation for encoding a complete distribution over a multi-dimensional space and a graph that is a compact or factorized representation of a set of independences that hold in the specific distribution. Two branches of graphical representations of distributions

are commonly used, namely, Bayesian networks and Markov networks. Both families encompass the properties of factorization and independences, but they differ in the set of independences they can encode and the factorization of the distribution that they induce.[1]

16.1.1 Bayesian network

Main article: Bayesian network

If the network structure of the model is a directed acyclic graph, the model represents a factorization of the joint probability of all random variables. More precisely, if the events are $X_1, \ldots, X_n$ then the joint probability satisfies

$$P[X_1, \ldots, X_n] = \prod_{i=1}^{n} P[X_i | pa_i]$$

where pa_i is the set of parents of node X_i . In other words, the joint distribution factors into a product of conditional distributions. For example, the graphical model in the Figure shown above (which is actually not a directed acyclic graph, but an ancestral graph) consists of the random variables A, B, C, D with a joint probability density that factors as

$$P[A, B, C, D] = P[A] \cdot P[B] \cdot P[C|B, D] \cdot P[D|A, B, C].$$

Any two nodes are conditionally independent given the values of their parents. In general, any two sets of nodes are conditionally independent given a third set if a criterion called *d*-separation holds in the graph. Local independences and global independences are equivalent in Bayesian networks.

This type of graphical model is known as a directed graphical model, Bayesian network, or belief network. Classic machine learning models like hidden Markov models, neural networks and newer models such as variable-order Markov models can be considered special cases of Bayesian networks.

16.1.2 Markov random field

Main article: Markov random field

A Markov random field, also known as a Markov network, is a model over an undirected graph. A graphical model with many repeated subunits can be represented with plate notation.

16.1.3 Other types

- A factor graph is an undirected bipartite graph connecting variables and factors. Each factor represents a function over the variables it is connected to. This is a helpful representation for understanding and implementing belief propagation.
- A clique tree or junction tree is a tree of cliques, used in the junction tree algorithm.
- A chain graph is a graph which may have both directed and undirected edges, but without any directed cycles (i.e. if we start at any vertex and move along the graph respecting the directions of any arrows, we cannot return to the vertex we started from if we have passed an arrow). Both directed acyclic graphs and undirected graphs are special cases of chain graphs, which can therefore provide a way of unifying and generalizing Bayesian and Markov networks.[2]
- An ancestral graph is a further extension, having directed, bidirected and undirected edges.[3]
- A conditional random field is a discriminative model specified over an undirected graph.
- A restricted Boltzmann machine is a generative model specified over an undirected graph.

16.2 Applications

The framework of the models, which provides algorithms for discovering and analyzing structure in complex distributions to describe them succinctly and extract the unstructured information, allows them to be constructed and utilized effectively.[1] Applications of graphical models include information extraction, speech recognition, computer vision, decoding of low-density parity-check codes, modeling of gene regulatory networks, gene finding and diagnosis of diseases, and graphical models for protein structure.

16.3 See also

- Belief propagation
- Structural equation model

16.4 Notes

[1] Koller; Friedman (2009). Probabilistic Graphical Models. Massachusetts: MIT Press. ISBN 0-262-01319-3.

[2] Frydenberg, Morten (1990). "The Chain Graph Markov Property". *Scandinavian Journal of Statistics* **17** (4): 333–353. JSTOR 4616181. MR 1096723.

[3] Richardson, Thomas; Spirtes, Peter (2002). "Ancestral graph Markov models". *Annals of Statistics* **30** (4): 962–1030. doi:10.1214/aos/1031689015. MR 1926166. Zbl 1033.60008.

16.5 Tutorial

- Graphical models and Conditional Random Fields
- Probabilistic Graphical Models taught by Eric Xing at CMU

16.6 References and further reading

16.6.1 Books and book chapters

- Bishop, Christopher M. (2006). "Chapter 8. Graphical Models" (PDF). *Pattern Recognition and Machine Learning*. Springer. pp. 359–422. ISBN 0-387-31073-8. MR 2247587.
- Cowell, Robert G.; Dawid, A. Philip; Lauritzen, Steffen L.; Spiegelhalter, David J. (1999). *Probabilistic networks and expert systems*. Berlin: Springer. ISBN 0-387-98767-3. MR 1697175. A more advanced and statistically oriented book
- Jensen, Finn (1996). *An introduction to Bayesian networks*. Berlin: Springer. ISBN 0-387-91502-8.
- Koller, D.; Friedman, N. (2009). *Probabilistic Graphical Models*. Massachusetts: MIT Press. p. 1208. ISBN 0-262-01319-3.
- Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems* (2nd revised ed.). San Mateo, CA: Morgan Kaufmann. ISBN 1-55860-479-0. MR 0965765. A computational reasoning approach, where the relationships between graphs and probabilities were formally introduced.

16.6.2 Journal articles

- Edoardo M. Airoldi (2007). "Getting Started in Probabilistic Graphical Models". *PLoS Computational Biology* **3** (12): e252. doi:10.1371/journal.pcbi.0030252. PMC 2134967. PMID 18069887.
- Jordan, M. I. (2004). "Graphical Models". *Statistical Science* **19**: 140–155. doi:10.1214/088342304000000026.

16.6.3 Other

- Heckerman's Bayes Net Learning Tutorial
- A Brief Introduction to Graphical Models and Bayesian Networks
- Sargur Srihari's lecture slides on probabilistic graphical models

Chapter 17

Influence diagram

An **influence diagram (ID)** (also called a **relevance diagram**, **decision diagram** or a **decision network**) is a compact graphical and mathematical representation of a decision situation. It is a generalization of a Bayesian network, in which not only probabilistic inference problems but also decision making problems (following maximum expected utility criterion) can be modeled and solved.

ID was first developed in mid-1970s within the decision analysis community with an intuitive semantic that is easy to understand. It is now adopted widely and becoming an alternative to decision tree which typically suffers from exponential growth in number of branches with each variable modeled. ID is directly applicable in team decision analysis, since it allows incomplete sharing of information among team members to be modeled and solved explicitly. Extension of ID also find its use in game theory as an alternative representation of game tree.

17.1 Semantics

An ID is a directed acyclic graph with three types (plus one subtype) of node and three types of arc (or arrow) between nodes.

Nodes;

- *Decision node* (corresponding to each decision to be made) is drawn as a rectangle.
- *Uncertainty node* (corresponding to each uncertainty to be modeled) is drawn as an oval.
    - *Deterministic node* (corresponding to special kind of uncertainty that its outcome is deterministically known whenever the outcome of some other uncertainties are also known) is drawn as a double oval.
- *Value node* (corresponding to each component of additively separable Von Neumann-Morgenstern utility function) is drawn as an octagon (or diamond).

Arcs;

- *Functional arcs* (ending in value node) indicate that one of the components of additively separable utility function is a function of all the nodes at their tails.
- *Conditional arcs* (ending in uncertainty node) indicate that the uncertainty at their heads is probabilistically conditioned on all the nodes at their tails.
    - *Conditional arcs* (ending in deterministic node) indicate that the uncertainty at their heads is deterministically conditioned on all the nodes at their tails.
- *Informational arcs* (ending in decision node) indicate that the decision at their heads is made with the outcome of all the nodes at their tails known beforehand.

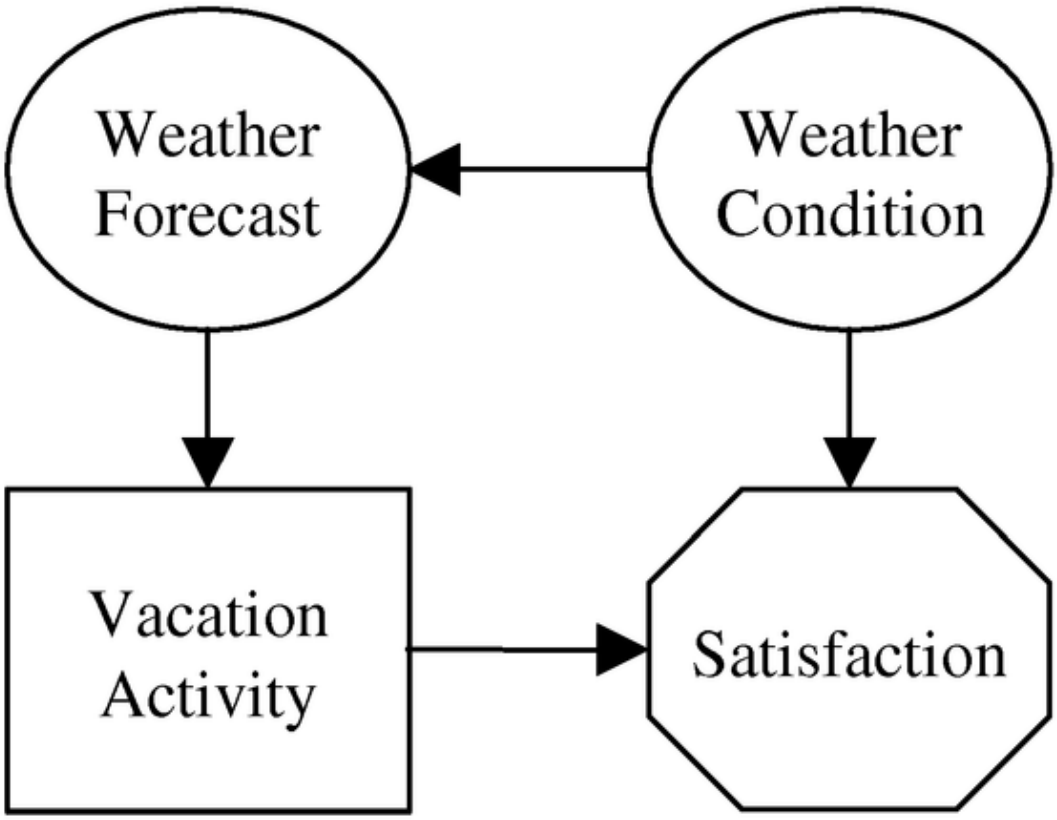
Given a properly structured ID;

- Decision nodes and incoming information arcs collectively state the *alternatives* (what can be done when the outcome of certain decisions and/or uncertainties are known beforehand)
- Uncertainty/deterministic nodes and incoming conditional arcs collectively model the *information* (what are known and their probabilistic/deterministic relationships)
- Value nodes and incoming functional arcs collectively quantify the *preference* (how things are preferred over one another).

Alternative, information, and preference are termed *decision basis* in decision analysis, they represent three required components of any valid decision situation.

Formally, the semantic of influence diagram is based on sequential construction of nodes and arcs, which implies a specification of all conditional independencies in the diagram. The specification is defined by the d -separation criterion of Bayesian network. According to this semantic, every node is probabilistically independent on its non-successor nodes given the outcome of its immediate predecessor nodes. Likewise, a missing arc between non-value node X and non-value node Y implies that there exists a set of non-value nodes Z , e.g., the parents of Y , that renders Y independent of X given the outcome of the nodes in Z .

17.2 Example



Simple influence diagram for making decision about vacation activity

Consider the simple influence diagram representing a situation where a decision-maker is planning her vacation.

- There is 1 decision node (*Vacation Activity*), 2 uncertainty nodes (*Weather Condition, Weather Forecast*), and 1 value node (*Satisfaction*).
- There are 2 functional arcs (ending in *Satisfaction*), 1 conditional arc (ending in *Weather Forecast*), and 1 informational arc (ending in *Vacation Activity*).

- Functional arcs ending in *Satisfaction* indicate that *Satisfaction* is a utility function of *Weather Condition* and *Vacation Activity*. In other words, her satisfaction can be quantified if she knows what the weather is like and what her choice of activity is. (Note that she does not value *Weather Forecast* directly)
- Conditional arc ending in *Weather Forecast* indicates her belief that *Weather Forecast* and *Weather Condition* can be dependent.
- Informational arc ending in *Vacation Activity* indicates that she will only know *Weather Forecast*, not *Weather Condition*, when making her choice. In other words, actual weather will be known after she makes her choice, and only forecast is what she can count on at this stage.

- It also follows semantically, for example, that *Vacation Activity* is independent on (irrelevant to) *Weather Condition* given *Weather Forecast* is known.

17.3 Applicability in value of information

The above example highlights the power of influence diagram in representing an extremely important concept in decision analysis known as value of information. Consider the following three scenarios;

- Scenario 1: The decision-maker could make her *Vacation Activity* decision while knowing what *Weather Condition* will be like. This corresponds to adding extra informational arc from *Weather Condition* to *Vacation Activity* in the above influence diagram.
- Scenario 2: The original influence diagram as shown above.
- Scenario 3: The decision-maker makes her decision without even knowing the *Weather Forecast*. This corresponds to removing informational arc from *Weather Forecast* to *Vacation Activity* in the above influence diagram.

Scenario 1 is the best possible scenario for this decision situation since there is no longer any uncertainty on what she cares about (*Weather Condition*) when making her decision. Scenario 3, however, is the worst possible scenario for this decision situation since she needs to make her decision without any hint (*Weather Forecast*) on what she cares about (*Weather Condition*) will turn out to be.

The decision-maker is usually better off (definitely no worse off) to move from scenario 3 to scenario 2 through the acquisition of new information. The most she should be willing to pay for such move is called value of information on *Weather Forecast*, which is essentially value of imperfect information on *Weather Condition*.

Likewise, it is the best for the decision-maker to move from scenario 3 to scenario 1. The most she should be willing to pay for such move is called value of perfect information on *Weather Condition*.

The applicability of this simple ID and the value of information concept is tremendous, especially in medical decision making when most decisions have to be made with imperfect information about patients, diseases, etc.

17.4 Notes

Influence diagrams are hierarchical and can be defined either in terms of their structure or in greater detail in terms of the functional and numerical relation between diagram elements. An ID that is consistently defined at all levels—structure, function, and number—is a well-defined mathematical representation and is referred to as a *well-formed influence diagram* (WFID). WFIDs can be evaluated using reversal and removal operations to yield answers to a large class of probabilistic, inferential, and decision questions. More recent techniques have been developed by artificial intelligence community with their works around Bayesian network inference (Belief propagation).

Influence diagram having only uncertainty nodes (i.e., Bayesian network) is also called **relevance diagram**. This is perhaps a better use of language than *influence diagram*. An arc connecting node A to B implies not only that "A is relevant to B", but also that "B is relevant to A" (i.e., relevance is a symmetric relationship). The word *influence* implies more of a one-way relationship, which is reinforced by the arc having a defined direction. Since some arcs are easily reversed, this “one-way” thinking that somehow "A influences B" is incorrect (the causality could be the other way around). However, the term *relevance diagram* is never adopted in larger community, and the world continues to refer to *influence diagram*.

17.5 Bibliography

- Cardenas, I. et al. (April 2015). "Modeling the Influence of Unknown Factors in Risk Analysis using Bayesian Networks" (PDF). *Under review by a refereed journal*.
- Detwarasiti, A.; Shachter, R.D. (December 2005). "Influence diagrams for team decision analysis" (PDF). *Decision Analysis* **2** (4): 207–228. doi:10.1287/deca.1050.0047.
- Holtzman, Samuel (1988). *Intelligent decision systems*. Addison-Wesley. ISBN 978-0-201-11602-1.
- Howard, R.A. and J.E. Matheson, "Influence diagrams" (1981), in *Readings on the Principles and Applications of Decision Analysis*, eds. R.A. Howard and J.E. Matheson, Vol. II (1984), Menlo Park CA: Strategic Decisions Group.
- Koller, D.; Milch, B. (October 2003). "Multi-agent influence diagrams for representing and solving games" (PDF). *Games and Economic Behavior* **45**: 181–221. doi:10.1016/S0899-8256(02)00544-4.
- Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Representation and Reasoning Series. San Mateo CA: Morgan Kaufmann. ISBN 0-934613-73-7.
- Shachter, R.D. (November–December 1986). "Evaluating influence diagrams" (PDF). *Operations Research* **34** (6): 871–882. doi:10.1287/opre.34.6.871.
- Shachter, R.D. (July–August 1988). "Probabilistic inference and influence diagrams" (PDF). *Operations Research* **36** (4): 589–604. doi:10.1287/opre.36.4.589.
- Virine, Lev; Trumper, Michael (2008). *Project Decisions: The Art and Science*. Vienna VA: Management Concepts. ISBN 978-1-56726-217-9.
- Pearl, J. (1985). *Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning* (UCLA TECHNICAL REPORT CSD-850017). Proceedings of the Seventh Annual Conference of the Cognitive Science Society 15–17 April 1985. http://ftp.cs.ucla.edu/tech-report/198_-reports/850017.pdf., University of California, Irvine, CA. pp. 329–334. Retrieved 2010-05-01.

17.6 See also

- Bayesian network
- Decision making software
- Decision tree
- Morphological analysis
- Node removal
- Node reversal

17.7 External links

- What are influence diagrams?
- Pearl, J. (December 2005). "Influence Diagrams — Historical and Personal Perspectives" (PDF). *Decision Analysis* **2** (4): 232–4. doi:10.1287/deca.1050.0055.

Chapter 18

Junction tree algorithm

The **junction tree algorithm** (also known as 'Clique Tree') is a method used in machine learning to extract marginalization in general graphs. In essence, it entails performing belief propagation on a modified graph called a junction tree. The basic premise is to eliminate cycles by clustering them into single nodes.

18.1 Junction tree algorithm

18.1.1 Hugin algorithm

- If the graph is directed then moralize it to make it undirected
- Introduce the evidence
- Triangulate the graph to make it chordal
- Construct a junction tree from the triangulated graph (we will call the vertices of the junction tree "supernodes")
- Propagate the probabilities along the junction tree (via belief propagation)

Note that this last step is inefficient for graphs of large treewidth. Computing the messages to pass between supernodes involves doing exact marginalization over the variables in both supernodes. Performing this algorithm for a graph with treewidth k will thus have at least one computation which takes time exponential in k.

18.1.2 Shafer-Shenoy algorithm

18.2 References

- Lauritzen, Steffen L.; Spiegelhalter, David J. (1988). "Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems". *Journal of the Royal Statistical Society. Series B (Methodological)* (Blackwell Publishing) **50** (2): 157–224. JSTOR 2345762. MR 0964177.
- Dawid, A. P. (1992). "Applications of a general propagation algorithm for probabilistic expert systems". *Statistics and Computing* (Springer) **2** (1): 25–26. doi:10.1007/BF01890546.
- Huang, Cecil; Darwiche, Adnan (1996). "Inference in Belief Networks: A Procedural Guide". *International Journal of Approximate Reasoning* (Elsevier) **15** (3): 225–263. doi:10.1016/S0888-613X(96)00069-2.
- Paskin, Mark A. Missing or empty |title= (help); |chapter= ignored (help)

Chapter 19

Latent variable

In statistics, **latent variables** or hidden variables (from Latin: present participle of *lateo* ("lie hidden"),[1] as opposed to observable variables), are variables that are not directly observed but are rather inferred (through a mathematical model) from other variables that are observed (directly measured). Mathematical models that aim to explain observed variables in terms of latent variables are called latent variable models. Latent variable models are used in many disciplines, including psychology, economics, medicine, physics, machine learning/artificial intelligence, bioinformatics, natural language processing, econometrics, management and the social sciences.

Sometimes latent variables correspond to aspects of physical reality, which could in principle be measured, but may not be for practical reasons. In this situation, the term **hidden variables** is commonly used (reflecting the fact that the variables are "really there", but hidden). Other times, latent variables correspond to abstract concepts, like categories, behavioral or mental states, or data structures. The terms **hypothetical variables** or **hypothetical constructs** may be used in these situations.

One advantage of using latent variables is that it reduces the dimensionality of data. A large number of observable variables can be aggregated in a model to represent an underlying concept, making it easier to understand the data. In this sense, they serve a function similar to that of scientific theories. At the same time, latent variables link observable ("sub-symbolic") data in the real world to symbolic data in the modeled world.

Latent variables, as created by factor analytic methods, generally represent "shared" variance, or the degree to which variables "move" together. Variables that have no correlation cannot result in a latent construct based on the common factor model.[2]

19.1 Examples of latent variables

19.1.1 Economics

Examples of latent variables from the field of economics include quality of life, business confidence, morale, happiness and conservatism: these are all variables which cannot be measured directly. But linking these latent variables to other, observable variables, the values of the latent variables can be inferred from measurements of the observable variables. Quality of life is a latent variable which can not be measured directly so observable variables are used to infer quality of life. Observable variables to measure quality of life include wealth, employment, environment, physical and mental health, education, recreation and leisure time, and social belonging.

19.1.2 Psychology

- The "Big Five personality traits" have been inferred using factor analysis.
- extraversion[3]
- spatial ability[3]
- wisdom "Two of the more predominant means of assessing wisdom include wisdom-related performance and latent variable measures."[4]

- Spearman's g, or the general intelligence factor in psychometrics[5]

19.2 Common methods for inferring latent variables

- Hidden Markov models
- Factor analysis
- Principal component analysis
- Latent semantic analysis and Probabilistic latent semantic analysis
- EM algorithms

19.2.1 Bayesian algorithms and methods

Bayesian statistics is often used for inferring latent variables.

- Latent Dirichlet Allocation
- The Chinese Restaurant Process is often used to provide a prior distribution over assignments of objects to latent categories.
- The Indian buffet process is often used to provide a prior distribution over assignments of latent binary features to objects.

19.3 See also

- Latent variable model
- Item response theory
- Rasch model
- Proxy (statistics)
- Partial least squares path modeling
- Partial least squares regression
- Structural equation modeling

19.4 References

[1] "Wiktionary". *http://en.wiktionary.org/wiki/latent". Retrieved 19 November 2014.*

[2] Tabachnick, B.G.; Fidell, L.S. (2001). *Using Multivariate Analysis*. Boston: Allyn and Bacon. ISBN 0-321-05677-9.

[3] Borsboom, D.; Mellenbergh, G.J.; van Heerden, J. (2003). "The Theoretical Status of Latent Variables" (PDF). *Psychological Review* **110** (2): 203–219. doi:10.1037/0033-295X.110.2.203.

[4] Greene, Jeffrey A.; Brown, Scott C. (2009). "The Wisdom Development Scale: Further Validity Investigations". *International Journal of Aging And Human Development* **68** (4): 289–320 (at p. 291). PMID 19711618.

[5] Spearman, C. (1904). ""General Intelligence," Objectively Determined and Measured". *The American Journal of Psychology* **15** (2): 201–292. doi:10.2307/1412107. JSTOR 1412107.

Chapter 20

M-separation

In statistics, ***m*-separation** is a measure of disconnectedness in ancestral graphs and a generalization of d-separation for directed acyclic graphs. It is the opposite of ***m*-connectedness**.

Suppose G is an ancestral graph. For given source and target nodes s and t and a set Z of nodes in $G\backslash\{s, t\}$, m-connectedness can be defined as follows. Consider a path from s to t. An intermediate node on the path is called a *collider* if both edges on the path touching it are directed toward the node. The path is said to *m-connect* the nodes s and t, given Z, if and only if:

- every non-collider on the path is outside Z, and
- for each collider c on the path, either c is in Z or there is a directed path from c to an element of Z.

If s and t cannot be m-connected by any path satisfying the above conditions, then the nodes are said to be *m-separated*.

The definition can be extended to node sets S and T. Specifically, S and T are m-connected if each node in S can be m-connected to any node in T, and are m-separated otherwise.

20.1 References

- Drton, Mathias and Thomas Richardson. *Iterative Conditional Fitting for Gaussian Ancestral Graph Models*. Technical Report 437, December 2003.

20.2 See also

- d-separation

Chapter 21

Markov blanket

In machine learning, the **Markov blanket** for a node A in a Bayesian network is the set of nodes ∂A composed of A 's parents, its children, and its children's other parents. In a Markov network, the Markov blanket of a node is its set of neighboring nodes. A Markov blanket may also be denoted by $MB(A)$.

Every set of nodes in the network is conditionally independent of A when conditioned on the set ∂A , that is, when conditioned on the Markov blanket of the node A . The probability has the Markov property; formally, for distinct nodes A and B :

$$\Pr(A \mid \partial A, B) = \Pr(A \mid \partial A).$$

The Markov blanket of a node contains all the variables that shield the node from the rest of the network. This means that the Markov blanket of a node is the only knowledge needed to predict the behavior of that node. The term was coined by Pearl in 1988.[1]

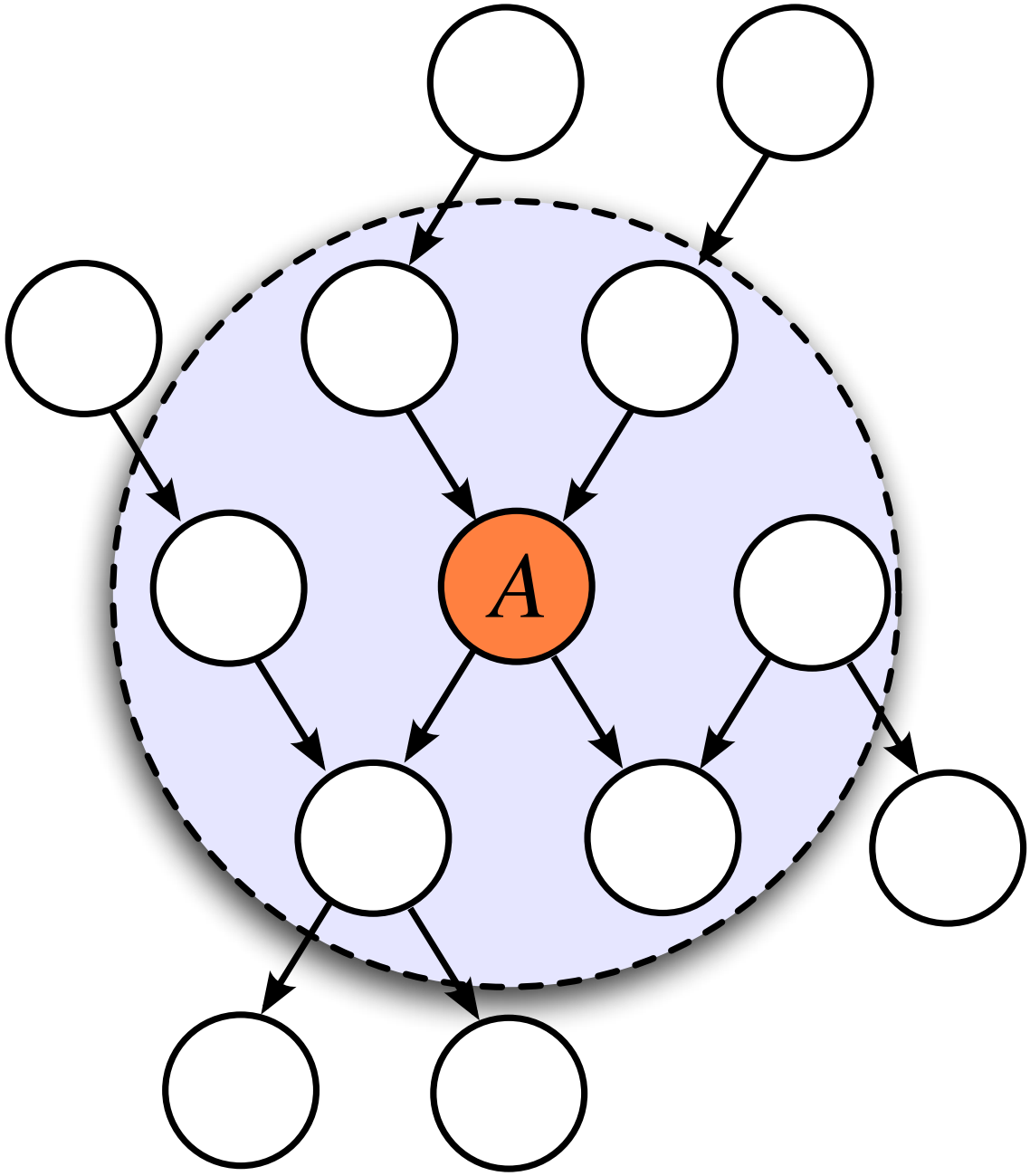
In a Bayesian network, the values of the parents and children of a node evidently give information about that node; however, its children's parents also have to be included, because they can be used to explain away the node in question. In a Markov random field, the Markov blanket for a node is simply its adjacent nodes.

21.1 See also

- Moral graph

21.2 Notes

[1] Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Representation and Reasoning Series. San Mateo CA: Morgan Kaufmann. ISBN 0-934613-73-7.



In a Bayesian network, the Markov blanket of node A *includes its parents, children and the other parents of all of its children.*

Chapter 22

Markov logic network

A **Markov logic network** (or MLN) is a probabilistic logic which applies the ideas of a Markov network to first-order logic, enabling uncertain inference. Markov logic networks generalize first-order logic, in the sense that, in a certain limit, all unsatisfiable statements have a probability of zero, and all tautologies have probability one.

22.1 Description

Briefly, it is a collection of formulas from first order logic, to each of which is assigned a real number, the weight. Taken as a Markov network, the vertices of the network graph are atomic formulas, and the edges are the logical connectives used to construct the formula. Each formula is considered to be a clique, and the Markov blanket is the set of formulas in which a given atom appears. A potential function is associated to each formula, and takes the value of one when the formula is true, and zero when it is false. The potential function is combined with the weight to form the Gibbs measure and partition function for the Markov network.

The above definition glosses over a subtle point: atomic formulas do not have a truth value unless they are grounded and given an interpretation; that is, until they are ground atoms with a Herbrand interpretation. Thus, a Markov logic network becomes a Markov network only with respect to a specific grounding and interpretation; the resulting Markov network is called the **ground Markov network**. The vertices of the graph of the ground Markov network are the ground atoms. The size of the resulting Markov network thus depends strongly (exponentially) on the number of constants in the domain of discourse.

22.2 Inference

The goal of inference in a Markov logic network is to find the stationary distribution of the system, or one that is close to it; that this may be difficult or not always possible is illustrated by the richness of behaviour seen in the Ising model. As in a Markov network, the stationary distribution finds the most likely assignment of probabilities to the vertices of the graph; in this case, the vertices are the ground atoms of an interpretation. That is, the distribution indicates the probability of the truth or falsehood of each ground atom. Given the stationary distribution, one can then perform inference in the traditional statistical sense of conditional probability: obtain the probability $P(A|B)$ that formula A holds, given that formula B is true.

Inference in MLNs can be performed using standard Markov network inference techniques over the minimal subset of the relevant Markov network required for answering the query. These techniques include Gibbs sampling, which is effective but may be excessively slow for large networks, belief propagation, or approximation via pseudolikelihood.

22.3 Resources

- Richardson, Matthew; Domingos, Pedro (2006). "Markov Logic Networks" (PDF). *Machine Learning* **62** (1-2): 107–136. doi:10.1007/s10994-006-5833-1.

22.4 See also

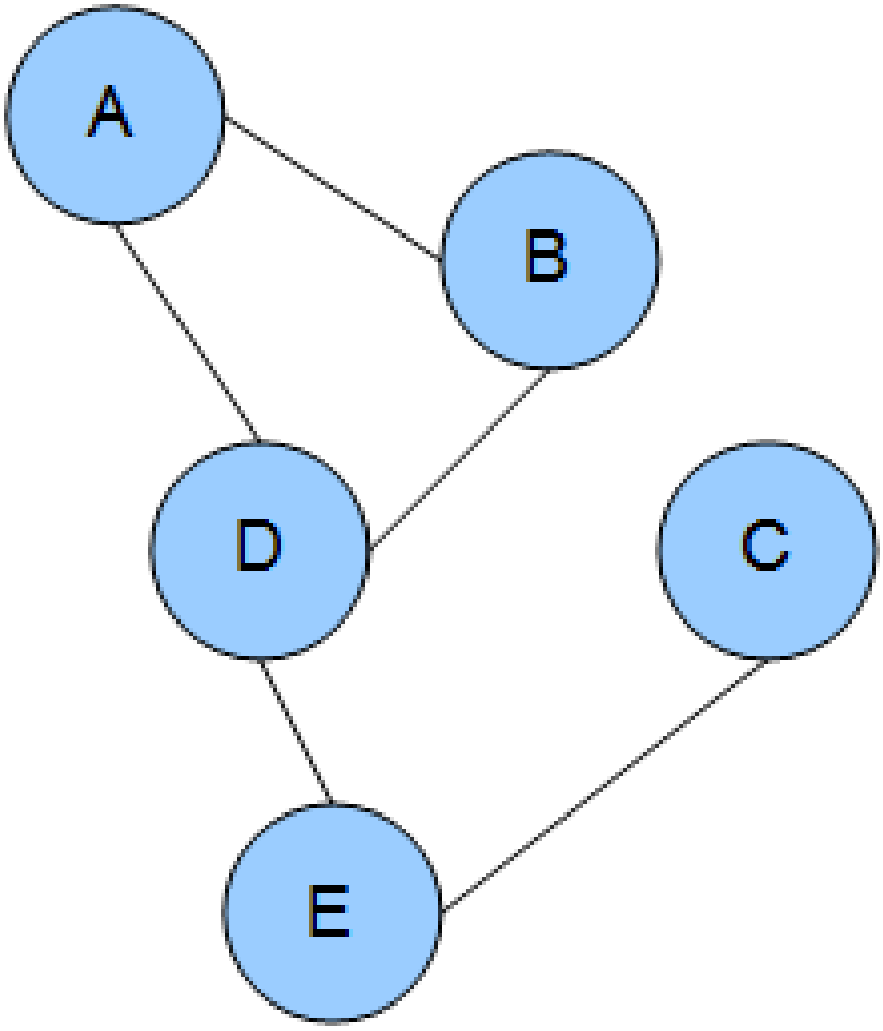
- Statistical relational learning
- Probabilistic logic network

22.5 External links

- University of Washington Statistical Relational Learning group
- Alchemy 2.0: Markov logic networks in C++
- ProbCog: Markov logic networks in Python and Java that can use its own inference engine or Alchemy's
- markov thebeast: Markov logic networks in Java
- RockIt: Markov logic networks in Java (with web interface/REST API)
- Tuffy: A Learning and Inference Engine with strong RDBMs-based optimization for scalability
- Felix: A successor to Tuffy, with prebuilt submodules to speed up common subtasks
- Factorie: Scala based probabilistic inference language, with prebuilt submodules for natural language processing etc
- Figaro: Scala based MLN language

Chapter 23

Markov random field



An example of a Markov random field. Each edge represents dependency. In this example: A depends on B and D. B depends on A and D. D depends on A, B, and E. E depends on D and C. C depends on E.

In the domain of physics and probability, a **Markov random field** (often abbreviated as **MRF**), **Markov network** or **undirected graphical model** is a set of random variables having a Markov property described by an undirected graph.

A Markov random field is similar to a Bayesian network in its representation of dependencies; the differences being that Bayesian networks are directed and acyclic, whereas Markov networks are undirected and may be cyclic. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies); on the other hand, it can't represent certain dependencies that a Bayesian network can (such as induced dependencies). The underlying graph of a Markov random field may be finite or infinite.

When the joint probability distribution of the random variables is strictly positive, it is also referred to as a **Gibbs random field**, because, according to the Hammersley–Clifford theorem, it can then be represented by a Gibbs measure for an appropriate (locally defined) energy function. The prototypical Markov random field is the Ising model; indeed, the Markov random field was introduced as the general setting for the Ising model.[1] In the domain of artificial intelligence, a Markov random field is used to model various low- to mid-level tasks in image processing and computer vision.[2] For example, MRFs are used for image restoration, image completion, segmentation, image registration, texture synthesis, super-resolution, stereo matching and information retrieval.

23.1 Definition

Given an undirected graph $G = (V, E)$, a set of random variables $X = (Xv)v \in V$ indexed by V form a Markov random field with respect to G if they satisfy the local Markov properties:

> **Pairwise Markov property**: Any two non-adjacent variables are conditionally independent given all other variables:
>
> $$X_u \perp\!\!\!\perp X_v \mid X_{V\setminus\{u,v\}} \quad \text{if} \{u, v\} \notin E$$
>
> **Local Markov property**: A variable is conditionally independent of all other variables given its neighbors:
>
> $$X_v \perp\!\!\!\perp X_{V\setminus \mathrm{cl}(v)} \mid X_{\mathrm{ne}(v)}$$
>
> where $\mathrm{ne}(v)$ is the set of neighbors of v, and $\mathrm{cl}(v) = \{v\} \cup \mathrm{ne}(v)$ is the closed neighbourhood of v.
>
> **Global Markov property**: Any two subsets of variables are conditionally independent given a separating subset:
>
> $$X_A \perp\!\!\!\perp X_B \mid X_S$$
>
> where every path from a node in A to a node in B passes through S.

The above three Markov properties are not equivalent: The Local Markov property is stronger than the Pairwise one, while weaker than the Global one.

23.2 Clique factorization

As the Markov properties of an arbitrary probability distribution can be difficult to establish, a commonly used class of Markov random fields are those that can be factorized according to the cliques of the graph.

Given a set of random variables $X = (Xv)v \in V$, let $P(X = x)$ be the probability of a particular field configuration x in X. That is, $P(X = x)$ is the probability of finding that the random variables X take on the particular value x. Because X is a set, the probability of x should be understood to be taken with respect to a *joint distribution* of the $\mathrm{X_v}$.

If this joint density can be factorized over the cliques of G:

$$P(X = x) = \prod_{C \in \mathrm{cl}(G)} \phi_C(x_C)$$

then *X* forms a Markov random field with respect to *G*. Here, cl(*G*) is the set of cliques of *G*. The definition is equivalent if only maximal cliques are used. The functions φC are sometimes referred to as *factor potentials* or *clique potentials*. Note, however, conflicting terminology is in use: the word *potential* is often applied to the logarithm of φC. This is because, in statistical mechanics, $\log(\varphi C)$ has a direct interpretation as the potential energy of a configuration *xC*.

Although some MRFs do not factorize (a simple example can be constructed on a cycle of 4 nodes[3]), in certain cases they can be shown to be equivalent conditions:

- if the density is positive (by the Hammersley–Clifford theorem),
- if the graph is chordal (by equivalence to a Bayesian network).

When such a factorization does exist, it is possible to construct a factor graph for the network.

23.3 Logistic model

Any Markov random field (with a strictly positive density) can be written as log-linear model with feature functions f_k such that the full-joint distribution can be written as

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_k w_k^\top f_k(x_{\{k\}})\right)$$

where the notation

$$w_k^\top f_k(x_{\{k\}}) = \sum_{i=1}^{N_k} w_{k,i} \cdot f_{k,i}(x_{\{k\}})$$

is simply a dot product over field configurations, and *Z* is the partition function:

$$Z = \sum_{x \in \mathcal{X}} \exp\left(\sum_k w_k^\top f_k(x_{\{k\}})\right).$$

Here, $\mathcal{X}$ denotes the set of all possible assignments of values to all the network's random variables. Usually, the feature functions $f_{k,i}$ are defined such that they are indicators of the clique's configuration, *i.e.* $f_{k,i}(x_{\{k\}}) = 1$ if $x_{\{k\}}$ corresponds to the *i*-th possible configuration of the *k*-th clique and 0 otherwise. This model is equivalent to the clique factorization model given above, if $N_k = |\operatorname{dom}(C_k)|$ is the cardinality of the clique, and the weight of a feature $f_{k,i}$ corresponds to the logarithm of the corresponding clique factor, *i.e.* $w_{k,i} = \log \phi(c_{k,i})$, where $c_{k,i}$ is the *i*-th possible configuration of the *k*-th clique, *i.e.* the *i*-th value in the domain of the clique C_k .

The probability *P* is often called the Gibbs measure. This expression of a Markov field as a logistic model is only possible if all clique factors are non-zero, *i.e.* if none of the elements of $\mathcal{X}$ are assigned a probability of 0. This allows techniques from matrix algebra to be applied, *e.g.* that the trace of a matrix is log of the determinant, with the matrix representation of a graph arising from the graph's incidence matrix.

The importance of the partition function *Z* is that many concepts from statistical mechanics, such as entropy, directly generalize to the case of Markov networks, and an *intuitive* understanding can thereby be gained. In addition, the partition function allows variational methods to be applied to the solution of the problem: one can attach a driving force to one or more of the random variables, and explore the reaction of the network in response to this perturbation. Thus, for example, one may add a driving term *Jv*, for each vertex *v* of the graph, to the partition function to get:

$$Z[J] = \sum_{x \in \mathcal{X}} \exp\left(\sum_k w_k^\top f_k(x_{\{k\}}) + \sum_v J_v x_v\right)$$

Formally differentiating with respect to Jv gives the expectation value of the random variable Xv associated with the vertex v:

$$E[X_v] = \frac{1}{Z} \left.\frac{\partial Z[J]}{\partial J_v}\right|_{J_v=0}.$$

Correlation functions are computed likewise; the two-point correlation is:

$$C[X_u, X_v] = \frac{1}{Z} \left.\frac{\partial^2 Z[J]}{\partial J_u \partial J_v}\right|_{J_u=0, J_v=0}.$$

Log-linear models are especially convenient for their interpretation. A log-linear model can provide a much more compact representation for many distributions, especially when variables have large domains. They are convenient too because their negative log likelihoods are convex. Unfortunately, though the likelihood of a logistic Markov network is convex, evaluating the likelihood or gradient of the likelihood of a model requires inference in the model, which is in general computationally infeasible.

23.4 Examples

23.4.1 Gaussian Markov random field

A multivariate normal distribution forms a Markov random field with respect to a graph $G = (V, E)$ if the missing edges correspond to zeros on the precision matrix (the inverse covariance matrix):

$$X = (X_v)_{v \in V} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

such that

$$(\Sigma^{-1})_{uv} = 0 \quad \text{if} \quad \{u, v\} \notin E.$$ [4]

23.5 Inference

As in a Bayesian network, one may calculate the conditional distribution of a set of nodes $V' = \{v_1, \ldots, v_i\}$ given values to another set of nodes $W' = \{w_1, \ldots, w_j\}$ in the Markov random field by summing over all possible assignments to $u \notin V', W'$; this is called exact inference. However, exact inference is a #P-complete problem, and thus computationally intractable in the general case. Approximation techniques such as Markov chain Monte Carlo and loopy belief propagation are often more feasible in practice. Some particular subclasses of MRFs, such as trees (see Chow–Liu tree), have polynomial-time inference algorithms; discovering such subclasses is an active research topic. There are also subclasses of MRFs that permit efficient MAP, or most likely assignment, inference; examples of these include associative networks.[5][6] Another interesting sub-class is the one of decomposable models (when the graph is chordal): having a closed-form for the MLE, it is possible to discover a consistent structure for hundreds of variables.[7]

23.6 Conditional random fields

Main article: Conditional random field

One notable variant of a Markov random field is a **conditional random field**, in which each random variable may also be conditioned upon a set of global observations o . In this model, each function ϕ_k is a mapping from all assignments to both the clique k and the observations o to the nonnegative real numbers. This form of the Markov network may be more appropriate for producing discriminative classifiers, which do not model the distribution over the observations. CRFs were proposed by John D. Lafferty, Andrew McCallum and Fernando C.N. Pereira in 2001.[8]

23.7 See also

- Maximum entropy method
- Hopfield network
- Graphical model
- Markov chain
- Markov logic network
- Hammersley–Clifford theorem
- Interacting particle system
- Probabilistic cellular automata
- Log-linear analysis

23.8 References

[1] Kindermann, Ross; Snell, J. Laurie (1980). *Markov Random Fields and Their Applications* (PDF). American Mathematical Society. ISBN 0-8218-5001-6. MR 0620955.

[2] Li, S. Z. (2009). *Markov Random Field Modeling in Image Analysis*. Springer.

[3] Moussouris, John (1974). "Gibbs and Markov random systems with constraints". *Journal of Statistical Physics* **10** (1): 11–33. doi:10.1007/BF01011714. MR 0432132.

[4] Rue, Håvard; Held, Leonhard (2005). *Gaussian Markov random fields: theory and applications*. CRC Press. ISBN 1-58488-432-0.

[5] Taskar, Benjamin; Chatalbashev, Vassil; Koller, Daphne (2004), "Learning associative Markov networks", in Brodley, Carla E., *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, ACM International Conference Proceeding Series **69**, Association for Computing Machinery, doi:10.1145/1015330.1015444.

[6] Duchi, John C.; Tarlow, Daniel; Elidan, Gal; Koller, Daphne (2006), "Using Combinatorial Optimization within Max-Product Belief Propagation", in Schölkopf, Bernhard; Platt, John C.; Hoffman, Thomas, *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, Advances in Neural Information Processing Systems **19**, MIT Press, pp. 369–376.

[7] Petitjean, F.; Webb, G.I.; Nicholson, A.E. (2013). *Scaling log-linear analysis to high-dimensional data* (PDF). International Conference on Data Mining. Dallas, TX, USA: IEEE.

[8] "Two classic paper prizes for papers that appeared at ICML 2013". *ICML*. 2013. Retrieved 15 December 2014.

23.9 External links

- MRF implementation in C++ for regular 2D lattices

Chapter 24

Mixture distribution

See also: Mixture model

In probability and statistics, a **mixture distribution** is the probability distribution of a random variable that is derived from a collection of other random variables as follows: first, a random variable is selected by chance from the collection according to given probabilities of selection, and then the value of the selected random variable is realized. The underlying random variables may be random real numbers, or they may be random vectors (each having the same dimension), in which case the mixture distribution is a multivariate distribution.

In cases where each of the underlying random variables is continuous, the outcome variable will also be continuous and its probability density function is sometimes referred to as a **mixture density**. The cumulative distribution function (and the probability density function if it exists) can be expressed as a convex combination (i.e. a weighted sum, with non-negative weights that sum to 1) of other distribution functions and density functions. The individual distributions that are combined to form the mixture distribution are called the **mixture components**, and the probabilities (or weights) associated with each component are called the **mixture weights**. The number of components in mixture distribution is often restricted to being finite, although in some cases the components may be countably infinite. More general cases (i.e. an uncountable set of component distributions), as well as the countable case, are treated under the title of compound distributions.

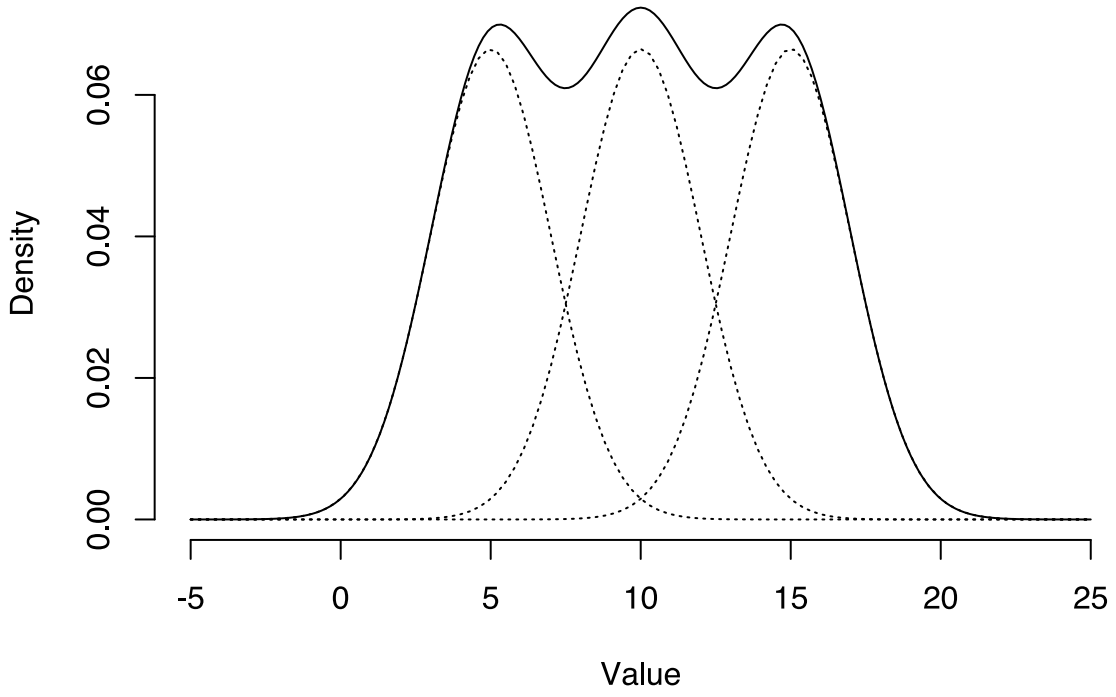
A distinction needs to be made between a random variable whose distribution function or density is the sum of a set of components (i.e. a mixture distribution) and a random variable whose value is the sum of the values of two or more underlying random variables, in which case the distribution is given by the convolution operator. As an example, the sum of two jointly normally distributed random variables, each with different means, will still have a normal distribution. On the other hand, a mixture density created as a mixture of two normal distributions with different means will have two peaks provided that the two means are far enough apart, showing that this distribution is radically different from a normal distribution.

Mixture distributions arise in many contexts in the literature and arise naturally where a statistical population contains two or more subpopulations. They are also sometimes used as a means of representing non-normal distributions. Data analysis concerning statistical models involving mixture distributions is discussed under the title of mixture models, while the present article concentrates on simple probabilistic and statistical properties of mixture distributions and how these relate to properties of the underlying distributions.

24.1 Finite and countable mixtures

Given a finite set of probability density functions $p_1(x)$, …, $pn(x)$, or corresponding cumulative distribution functions $P_1(x)$, …, $Pn(x)$ and **weights** w_1, …, wn such that $wi \geq 0$ and $\sum wi = 1$, the mixture distribution can be represented by writing either the density, f, or the distribution function, F, as a sum (which in both cases is a convex combination):

$$F(x) = \sum_{i=1}^{n} w_i \, P_i(x),$$



Density of a mixture of three normal distributions (μ = 5, 10, 15, σ = 2) with equal weights. Each component is shown as a weighted density (each integrating to 1/3)

$$f(x) = \sum_{i=1}^{n} w_i \, p_i(x).$$

This type of mixture, being a finite sum, is called a **finite mixture,** and in applications, an unqualified reference to a "mixture density" usually means a finite mixture. The case of a countably infinite set of components is covered formally by allowing $n = \infty$.

24.2 Uncountable mixtures

Main article: compound distribution

Where the set of component distributions is uncountable, the result is often called a compound probability distribution. The construction of such distributions has a formal similarity to that of mixture distributions, with either infinite summations or integrals replacing the finite summations used for finite mixtures.

Consider a probability density function $p(x;a)$ for a variable x, parameterized by a. That is, for each value of a in some set A, $p(x;a)$ is a probability density function with respect to x. Given a probability density function w (meaning that w is nonnegative and integrates to 1), the function

$$f(x) = \int_A w(a) \, p(x;a) \, da$$

is again a probability density function for x. A similar integral can be written for the cumulative distribution function. Note that the formulae here reduce to the case of a finite or infinite mixture if the density w is allowed to be a generalized function representing the "derivative" of the cumulative distribution function of a discrete distribution.

24.3 Mixtures of parametric families

The mixture components are often not arbitrary probability distributions, but instead are members of a parametric family (such as normal distributions), with different values for a parameter or parameters. In such cases, assuming that it exists, the density can be written in the form of a sum as:

$$f(x; a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i \, p(x; a_i)$$

for one parameter, or

$$f(x; a_1, \ldots, a_n, b_1, \ldots, b_n) = \sum_{i=1}^{n} w_i \, p(x; a_i, b_i)$$

for two parameters, and so forth.

24.4 Properties

24.4.1 Convexity

A general linear combination of probability density functions is not necessarily a probability density, since it may be negative or it may integrate to something other than 1. However, a convex combination of probability density functions preserves both of these properties (non-negativity and integrating to 1), and thus mixture densities are themselves probability density functions.

24.4.2 Moments

Let X_1, ..., Xn denote random variables from the n component distributions, and let X denote a random variable from the mixture distribution. Then, for any function $H(\cdot)$ for which $\mathrm{E}[H(X_i)]$ exists, and assuming that the component densities $pi(x)$ exist,

$$\begin{aligned}\mathrm{E}[H(X)] &= \int_{-\infty}^{\infty} H(x) \sum_{i=1}^{n} w_i p_i(x) \, dx \\ &= \sum_{i=1}^{n} w_i \int_{-\infty}^{\infty} p_i(x) H(x) \, dx = \sum_{i=1}^{n} w_i \, \mathrm{E}[H(X_i)].\end{aligned}$$

The relation,

$$\mathrm{E}[H(X)] = \sum_{i=1}^{n} w_i \, \mathrm{E}[H(X_i)],$$

holds more generally.

It is a trivial matter to note that the j^{th} moment about zero (i.e. choosing $H(x) = x^j$) is simply a weighted average of the j^{th} moments of the components. Moments about the mean $H(x) = (x - \mu)^j$ involve a binomial expansion:[1]

$$\begin{aligned}\mathrm{E}[(X - \mu)^j] &= \sum_{i=1}^{n} w_i \, \mathrm{E}[(X_i - \mu_i + \mu_i - \mu)^j] \\ &= \sum_{i=1}^{n} \sum_{k=0}^{j} \left(\begin{array}{c} j \\ k \end{array} \right) (\mu_i - \mu)^{j-k} w_i \, \mathrm{E}[(X_i - \mu_i)^k],\end{aligned}$$

where μi denotes the mean of the i^{th} component.

In case of a mixture of one-dimensional normal distributions with weights wi, means μi and variances σi^2, the total mean and variance will be:

$$\mathrm{E}[X] = \mu = \sum_{i=1}^{n} w_i \mu_i,$$

$$\mathrm{E}[(X-\mu)^2] = \sigma^2 = \sum_{i=1}^{n} w_i((\mu_i - \mu)^2 + \sigma_i^2).$$

These relations highlight the potential of mixture distributions to display non-trivial higher-order moments such as skewness and kurtosis (**fat tails**) and multi-modality, even in the absence of such features within the components themselves. Marron and Wand (1992) give an illustrative account of the flexibility of this framework.[2]

24.4.3 Modes

The question of multimodality is simple for some cases, such as mixtures of exponential distributions: all such mixtures are unimodal.[3] However, for the case of mixtures of normal distributions, it is a complex one. Conditions for the number of modes in a multivariate normal mixture are explored by Ray and Lindsay[4] extending the earlier work on univariate [5][6] and multivariate distributions (Carreira-Perpinan and Williams, 2003[7]).

Here the problem of evaluation of the modes of a n component mixture in a D dimensional space is reduced to identification of critical points (local minima, maxima and saddle points) on a manifold referred to as the ridgeline surface, which is the image of the ridgeline function

$$x^*(\alpha) = \left[\sum_{i=1}^{n} \alpha_i \Sigma_i^{-1}\right]^{-1} \times \left[\sum_{i=1}^{n} \alpha_i \Sigma_i^{-1} \mu_i\right],$$

where α belongs to the $n-1$ dimensional **unit simplex** $\mathcal{S}_n = \{\alpha \in \mathbb{R}^n : \alpha_i \in [0,1], \sum_{i=1}^{n} \alpha_i = 1\}$ and $\Sigma i \in \mathbf{R}^{D\times D}$, $\mu i \in \mathbf{R}^D$ correspond to the covariance and mean of the i^{th} component. Ray and Lindsay consider the case in which $n - 1 < D$ showing a one-to-one correspondence of modes of the mixture and those on the **elevation function** $h(\alpha) = q(x^*(\alpha))$ thus one may identify the modes by solving $\frac{dh(\alpha)}{d\alpha} = 0$ with respect to α and determining the value $x^*(\alpha)$.

Using graphical tools, the potential multi-modality of $n = \{2, 3\}$ mixtures is demonstrated; in particular it is shown that the number of modes may exceed n and that the modes may not be coincident with the component means. For two components they develop a graphical tool for analysis by instead solving the aforementioned differential with respect to w_1 and expressing the solutions as a function $\Pi(\alpha)$, $\alpha \in [0, 1]$ so that the number and location of modes for a given value of w_1 corresponds to the number of intersections of the graph on the line $\Pi(\alpha) = w_1$. This in turn can be related to the number of oscillations of the graph and therefore to solutions of $\frac{d\Pi(\alpha)}{d\alpha} = 0$ leading to an explicit solution for a two component homoscedastic mixture given by

$$1 - \alpha(1-\alpha) d_M(\mu_1, \mu_2, \Sigma)^2$$

where $dM(\mu_1, \mu_2, \Sigma) = (\mu_2 - \mu_1)^T \Sigma^{-1} (\mu_2 - \mu_1)$ is the Mahalanobis distance.

Since the above is quadratic it follows that in this instance there are at most two modes irrespective of the dimension or the weights.

24.5 Examples

Simple examples can be given by a mixture of two normal distributions.

Given an equal (50/50) mixture of two normal distributions with the same standard deviation and different means (homoscedastic), the overall distribution will exhibit low kurtosis relative to a single normal distribution – the means

of the subpopulations fall on the shoulders of the overall distribution. If sufficiently separated, namely by twice the (common) standard deviation, so $|\mu_1 - \mu_2| > 2\sigma$, these form a bimodal distribution, otherwise it simply has a wide peak.[8] The variation of the overall population will also be greater than the variation of the two subpopulations (due to spread from different means), and thus exhibits overdispersion relative to a normal distribution with fixed variation σ, though it will not be overdispersed relative to a normal distribution with variation equal to variation of the overall population.

Alternatively, given two subpopulations with the same mean and different standard deviations, the overall population will exhibit high kurtosis, with a sharper peak and heavier tails (and correspondingly shallower shoulders) than a single distribution.

- Univariate mixture distribution, showing bimodal distribution
- Multivariate mixture distribution, showing four modes

24.6 Applications

For more details on this topic, see Mixture model.

Mixture densities are complicated densities expressible in terms of simpler densities (the mixture components), and are used both because they provide a good model for certain data sets (where different subsets of the data exhibit different characteristics and can best be modeled separately), and because they can be more mathematically tractable, because the individual mixture components can be more easily studied than the overall mixture density.

Mixture densities can be used to model a statistical population with subpopulations, where the mixture components are the densities on the subpopulations, and the weights are the proportions of each subpopulation in the overall population.

Mixture densities can also be used to model experimental error or contamination – one assumes that most of the samples measure the desired phenomenon,

Parametric statistics that assume no error often fail on such mixture densities – for example, statistics that assume normality often fail disastrously in the presence of even a few outliers – and instead one uses robust statistics.

In meta-analysis of separate studies, study heterogeneity causes distribution of results to be a mixture distribution, and leads to overdispersion of results relative to predicted error. For example, in a statistical survey, the margin of error (determined by sample size) predicts the sampling error and hence dispersion of results on repeated surveys. The presence of study heterogeneity (studies have different sampling bias) increases the dispersion relative to the margin of error.

24.7 See also

- Convex combination
- Expectation-maximization algorithm
- Fat tail
- Not to be confused with: List_of_convolutions_of_probability_distributions

24.7.1 Mixture

- Mixture (probability)
- Mixture model

24.7.2 Hierarchical models

- Graphical model
- Hierarchical Bayes model

24.8 Notes

[1] Frühwirth-Schnatter (2006, Ch.1.2.4)

[2] Marron, J. S.; Wand, M. P. (1992). "Exact Mean Integrated Squared Error". *The Annals of Statistics* **20** (2): 712–736., http://projecteuclid.org/euclid.aos/1176348653

[3] Frühwirth-Schnatter (2006, Ch.1)

[4] Ray, R.; Lindsay, B. (2005), "The topography of multivariate normal mixtures", *The Annals of Statistics* **33** (5): 2042–2065

[5] Robertson CA, Fryer JG (1969) Some descriptive properties of normal mixtures. Skand Aktuarietidskr 137–146

[6] Behboodian J (1970) On the modes of a mixture of two normal distributions. Technometrics 12: 131–139

[7] http://faculty2.ucmerced.edu/mcarreira-perpinan/papers/EDI-INF-RR-0159.pdf

[8] Schilling, Mark F.; Watkins, Ann E.; Watkins, William (2002). "Is Human Height Bimodal?". *The American Statistician* **56** (3): 223–229. doi:10.1198/00031300265.

24.9 References

- Frühwirth-Schnatter, Sylvia (2006), *Finite Mixture and Markov Switching Models*, Springer, ISBN 978-1-4419-2194-9

Chapter 25

Mixture model

Not to be confused with mixed model.
See also: Mixture distribution

In statistics, a **mixture model** is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population. However, while problems associated with "mixture distributions" relate to deriving the properties of the overall population from those of the sub-populations, "mixture models" are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population identity information.

Some ways of implementing mixture models involve steps that attribute postulated sub-population-identities to individual observations (or weights towards such sub-populations), in which case these can be regarded as types of unsupervised learning or clustering procedures. However not all inference procedures involve such steps.

Mixture models should not be confused with models for compositional data, i.e., data whose components are constrained to sum to a constant value (1, 100%, etc.). However, compositional models can be thought of as mixture models, where members of the population are sampled at random. Conversely, mixture models can be thought of as compositional models, where the total size of the population has been normalized to 1.

25.1 Structure of a mixture model

25.1.1 General mixture model

A typical finite-dimensional mixture model is a hierarchical model consisting of the following components:

- N random variables corresponding to observations, each assumed to be distributed according to a mixture of K components, with each component belonging to the same parametric family of distributions (e.g., all normal, all Zipfian, etc.) but with different parameters
- N corresponding random latent variables specifying the identity of the mixture component of each observation, each distributed according to a K-dimensional categorical distribution
- A set of K mixture weights, each of which is a probability (a real number between 0 and 1 inclusive), all of which sum to 1
- A set of K parameters, each specifying the parameter of the corresponding mixture component. In many cases, each "parameter" is actually a set of parameters. For example, observations distributed according to a mixture of one-dimensional Gaussian distributions will have a mean and variance for each component. Observations distributed according to a mixture of V-dimensional categorical distributions (e.g., when each observation is a word from a vocabulary of size V) will have a vector of V probabilities, collectively summing to 1.

In addition, in a Bayesian setting, the mixture weights and parameters will themselves be random variables, and prior distributions will be placed over the variables. In such a case, the weights are typically viewed as a K-dimensional random vector drawn from a Dirichlet distribution (the conjugate prior of the categorical distribution), and the parameters will be distributed according to their respective conjugate priors.

Mathematically, a basic parametric mixture model can be described as follows:

$$
\begin{array}{lcl}
K & = & \text{components mixture of number} \\
N & = & \text{observations of number} \\
\theta_{i=1\ldots K} & = & \text{component with associated observation of distribution of parameter} i \\
\phi_{i=1\ldots K} & = & \text{component particular a of probability prior i.e., weight, mixture} i \\
\boldsymbol{\phi} & = & K \text{ individual the all of composed vector -dimensional} \phi_{1\ldots K} 1 \text{ to sum must ;} \\
z_{i=1\ldots N} & = & \text{observation of component} i \\
x_{i=1\ldots N} & = & \text{observation} i \\
F(x|\theta) & = & \text{on parametrized observation, an of distribution probability} \theta \\
z_{i=1\ldots N} & \sim & \text{Categorical}(\boldsymbol{\phi}) \\
x_{i=1\ldots N} & \sim & F(\theta_{z_i})
\end{array}
$$

In a Bayesian setting, all parameters are associated with random variables, as follows:

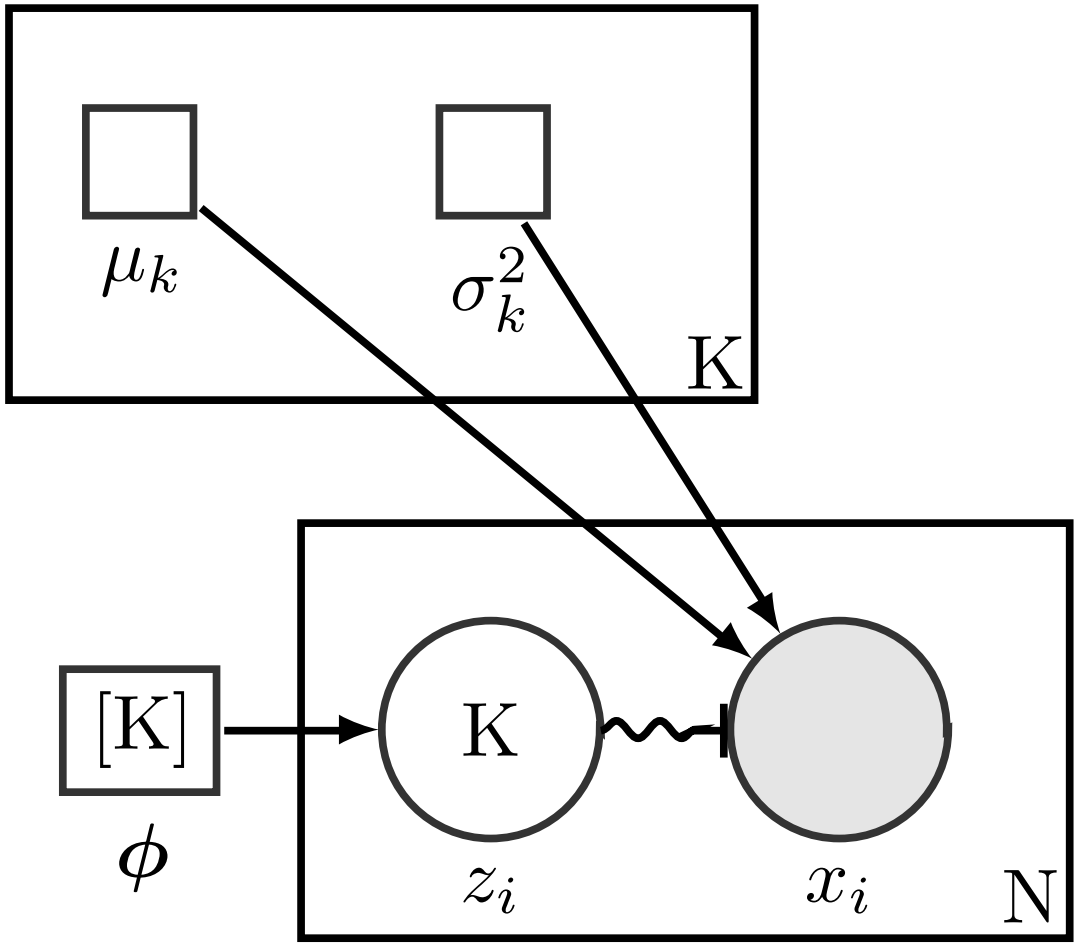
$$
\begin{array}{lcl}
K, N & = & \text{above as} \\
\theta_{i=1\ldots K}, \phi_{i=1\ldots K}, \boldsymbol{\phi} & = & \text{above as} \\
z_{i=1\ldots N}, x_{i=1\ldots N}, F(x|\theta) & = & \text{above as} \\
\alpha & = & \text{parameters component for hyperparameter shared} \\
\beta & = & \text{weights mixture for hyperparameter shared} \\
H(\theta|\alpha) & = & \text{on parametrized parameters, component of distribution probability prior} \alpha \\
\theta_{i=1\ldots K} & \sim & H(\theta|\alpha) \\
\boldsymbol{\phi} & \sim & \text{Symmetric-Dirichlet}_K(\beta) \\
z_{i=1\ldots N} & \sim & \text{Categorical}(\boldsymbol{\phi}) \\
x_{i=1\ldots N} & \sim & F(\theta_{z_i})
\end{array}
$$

This characterization uses F and H to describe arbitrary distributions over observations and parameters, respectively. Typically H will be the conjugate prior of F. The two most common choices of F are Gaussian aka "normal" (for real-valued observations) and categorical (for discrete observations). Other common possibilities for the distribution of the mixture components are:

- Binomial distribution, for the number of “positive occurrences” (e.g., successes, yes votes, etc.) given a fixed number of total occurrences
- Multinomial distribution, similar to the binomial distribution, but for counts of multi-way occurrences (e.g., yes/no/maybe in a survey)
- Negative binomial distribution, for binomial-type observations but where the quantity of interest is the number of failures before a given number of successes occurs
- Poisson distribution, for the number of occurrences of an event in a given period of time, for an event that is characterized by a fixed rate of occurrence
- Exponential distribution, for the time before the next event occurs, for an event that is characterized by a fixed rate of occurrence
- Log-normal distribution, for positive real numbers that are assumed to grow exponentially, such as incomes or prices
- Multivariate normal distribution (aka multivariate Gaussian distribution), for vectors of correlated outcomes that are individually Gaussian-distributed
- A vector of Bernoulli-distributed values, corresponding, e.g., to a black-and-white image, with each value representing a pixel; see the handwriting-recognition example below

25.1.2 Specific examples

Gaussian mixture model

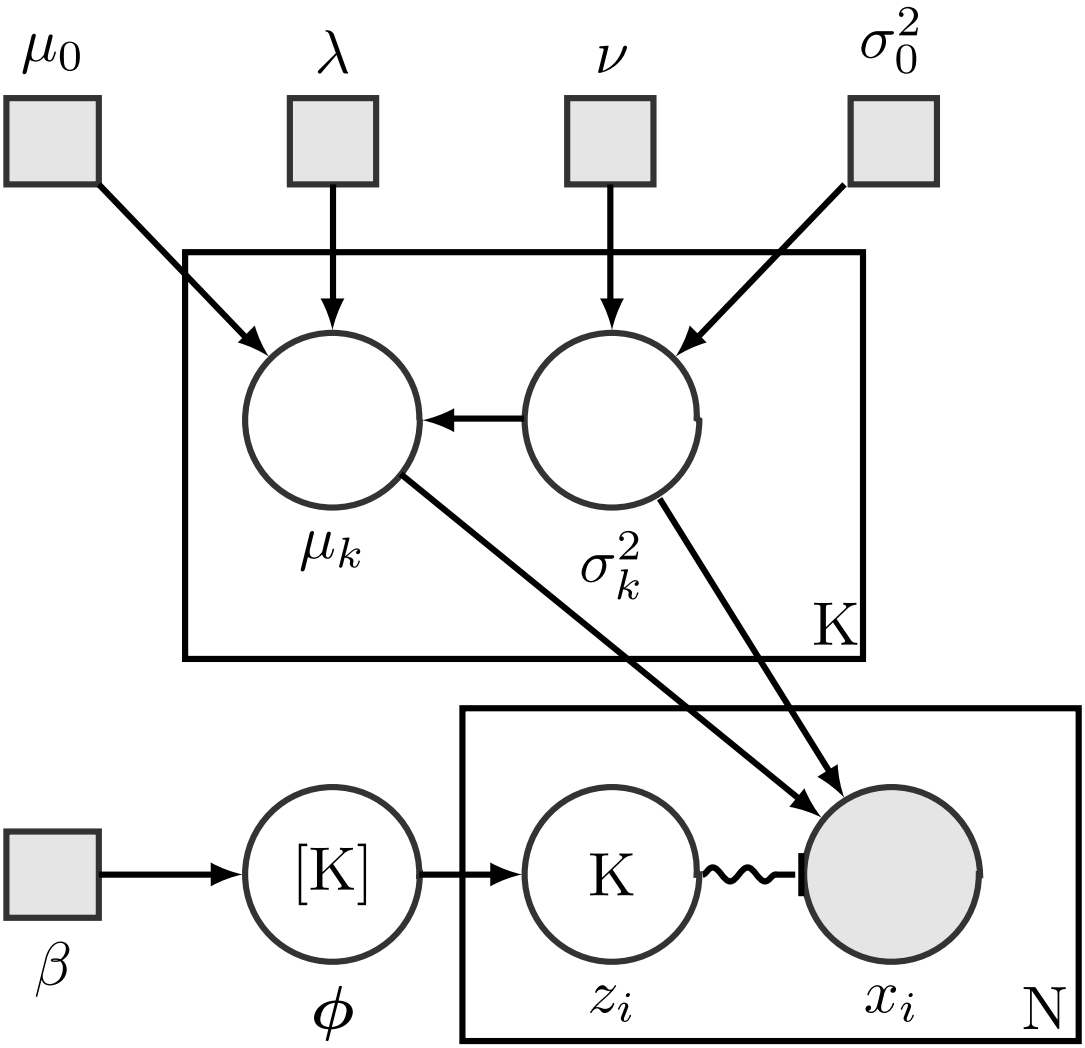


Non-Bayesian Gaussian mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size K.

A typical non-Bayesian Gaussian mixture model looks like this:

K, N	$=$	above as
$\phi_{i=1\ldots K}, \boldsymbol{\phi}$	$=$	above as
$z_{i=1\ldots N}, x_{i=1\ldots N}$	$=$	above as
$\mu_{i=1\ldots K}$	$=$	component of mean$i$
$\sigma^2_{i=1\ldots K}$	$=$	component of variance$i$
$z_{i=1\ldots N}$	$\sim$	Categorical$(\boldsymbol{\phi})$
$x_{i=1\ldots N}$	$\sim$	$\mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})$

A Bayesian version of a Gaussian mixture model is as follows:



Bayesian Gaussian mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size K.

K, N	$=$	above as
$\phi_{i=1\ldots K}, \boldsymbol{\phi}$	$=$	above as
$z_{i=1\ldots N}, x_{i=1\ldots N}$	$=$	above as
$\mu_{i=1\ldots K}$	$=$	component of mean$i$
$\sigma^2_{i=1\ldots K}$	$=$	component of variance$i$
$\mu_0, \lambda, \nu, \sigma_0^2$	$=$	hyperparameters shared
$\mu_{i=1\ldots K}$	$\sim$	$\mathcal{N}(\mu_0, \lambda\sigma_i^2)$
$\sigma^2_{i=1\ldots K}$	$\sim$	Inverse-Gamma(ν, σ_0^2)
$\boldsymbol{\phi}$	$\sim$	Symmetric-Dirichlet$_K(\beta)$
$z_{i=1\ldots N}$	$\sim$	Categorical$(\boldsymbol{\phi})$
$x_{i=1\ldots N}$	$\sim$	$\mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})$

Multivariate Gaussian mixture model

A Bayesian Gaussian mixture model is commonly extended to fit a vector of unknown parameters (denoted in bold), or multivariate normal distributions. In a multivariate distribution (i.e. one modelling a vector $\boldsymbol{x}$ with N random variables) one may model a vector of parameters (such as several observations of a signal or patches within an image) using a Gaussian mixture model prior distribution on the vector of estimates given by

$$p(\boldsymbol{\theta}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$$

where the i^{th} vector component is characterized by normal distributions with weights ϕ_i , means $\boldsymbol{\mu_i}$ and covariance matrices $\boldsymbol{\Sigma_i}$. To incorporate this prior into a Bayesian estimation, the prior is multiplied with the known distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$ of the data $\boldsymbol{x}$ conditioned on the parameters $\boldsymbol{\theta}$ to be estimated. With this formulation, the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ is "also" a Gaussian mixture model of the form

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{i=1}^{K} \tilde{\phi}_i \mathcal{N}(\tilde{\boldsymbol{\mu_i}}, \tilde{\boldsymbol{\Sigma}_i})$$

with new parameters $\tilde{\phi}_i, \tilde{\boldsymbol{\mu_i}}$ and $\tilde{\boldsymbol{\Sigma}_i}$ that are updated using the EM algorithm. [1] Although EM-based parameter updates are well-established, providing the initial estimates for these parameters is currently an area of active research. Note that this formulation yields a closed-form solution to the complete posterior distribution. Estimations of the random variable $\boldsymbol{\theta}$ may be obtained via one of several estimators, such as the mean or maximum of the posterior distribution.

Such distributions are useful for assuming patch-wise shapes of images and clusters, for example. In the case of image representation, each Gaussian may be tilted, expanded, and warped according to the covariance matrices $\boldsymbol{\Sigma_i}$. One Gaussian distribution of the set is fit to each patch (usually of size 8x8 pixels) in the image. Notably, any distribution of points around a cluster (see *k*-means) may be accurately given enough Gaussian components, but scarcely over *K*=20 components are needed to accurately model a given image distribution or cluster of data.

Categorical mixture model

A typical non-Bayesian mixture model with categorical observations looks like this:

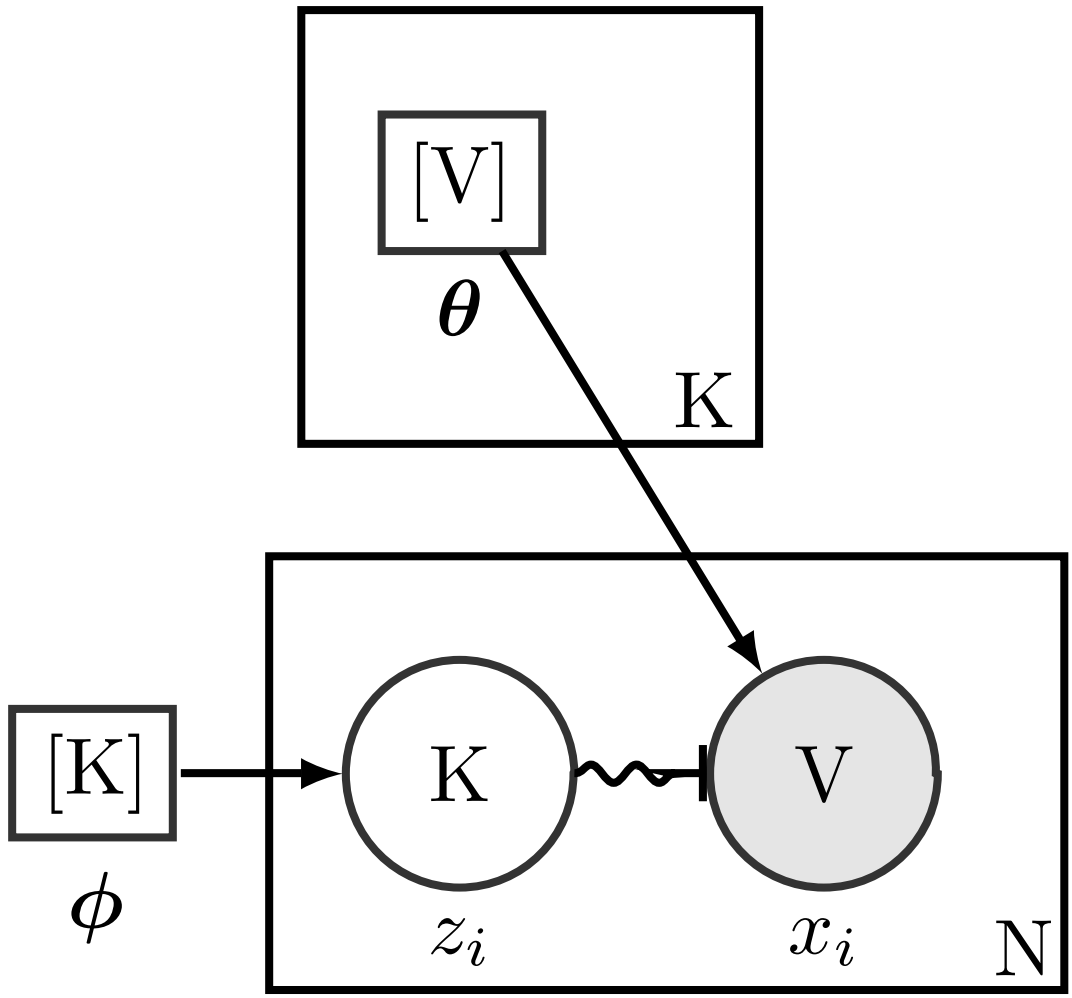
- K, N : as above
- $\phi_{i=1\ldots K}, \boldsymbol{\phi}$: as above
- $z_{i=1\ldots N}, x_{i=1\ldots N}$: as above
- V : dimension of categorical observations, e.g., size of word vocabulary
- $\theta_{i=1\ldots K, j=1\ldots V}$: probability for component i of observing item j
- $\boldsymbol{\theta}_{i=1\ldots K}$: vector of dimension V, composed of $\theta_{i,1\ldots V}$; must sum to 1

The random variables:

$$\begin{array}{lcl} z_{i=1\ldots N} & \sim & \text{Categorical}(\boldsymbol{\phi}) \\ x_{i=1\ldots N} & \sim & \text{Categorical}(\boldsymbol{\theta}_{z_i}) \end{array}$$

A typical Bayesian mixture model with categorical observations looks like this:

- K, N : as above
- $\phi_{i=1\ldots K}, \boldsymbol{\phi}$: as above
- $z_{i=1\ldots N}, x_{i=1\ldots N}$: as above
- V : dimension of categorical observations, e.g., size of word vocabulary
- $\theta_{i=1\ldots K, j=1\ldots V}$: probability for component i of observing item j
- $\boldsymbol{\theta}_{i=1\ldots K}$: vector of dimension V, composed of $\theta_{i,1\ldots V}$; must sum to 1



Non-Bayesian categorical mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size K*; likewise for [V].*

- α : shared concentration hyperparameter of $\boldsymbol{\theta}$ for each component
- β : concentration hyperparameter of $\boldsymbol{\phi}$

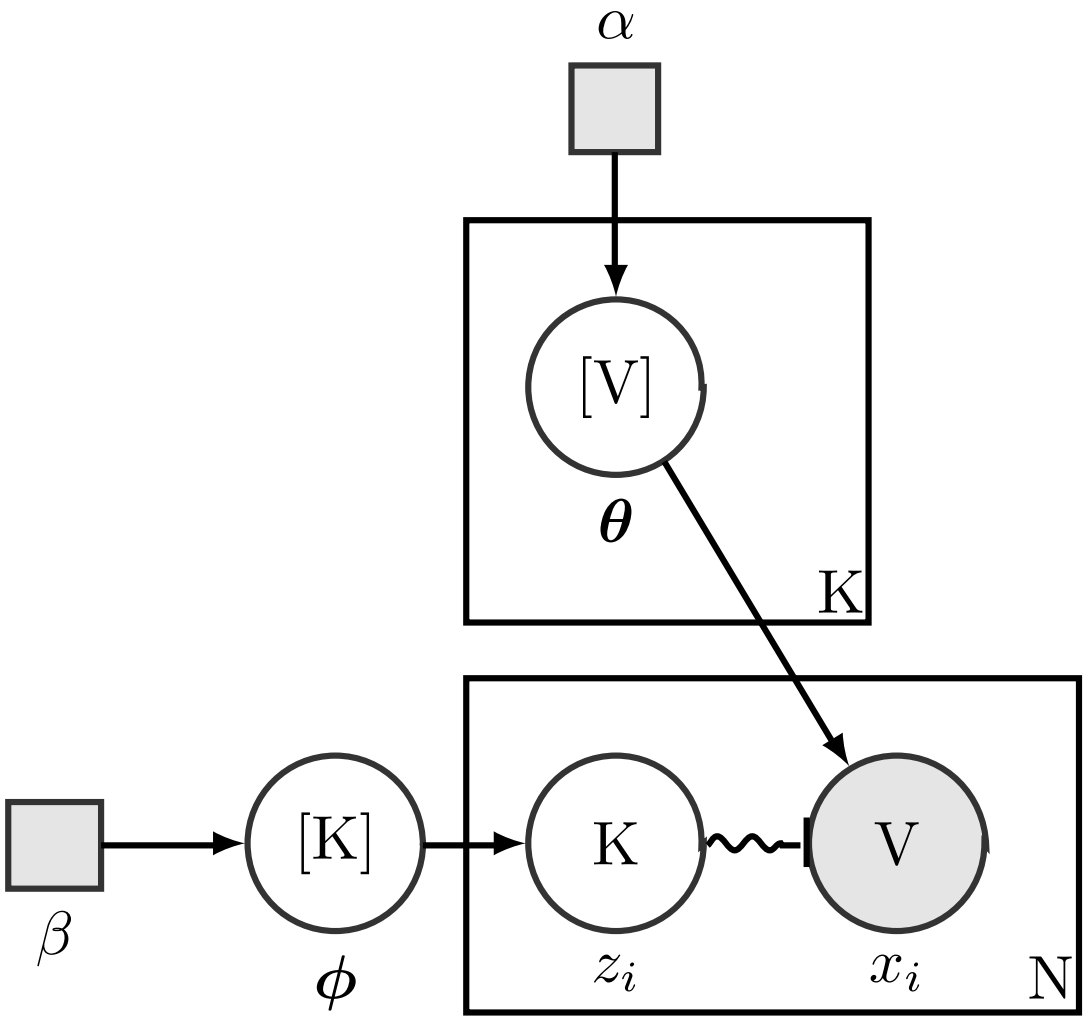
The random variables:

$$
\begin{array}{lcl}
\boldsymbol{\phi} & \sim & \text{Symmetric-Dirichlet}_K(\beta) \\
\boldsymbol{\theta}_{i=1\ldots K} & \sim & \text{Symmetric-Dirichlet}_V(\alpha) \\
z_{i=1\ldots N} & \sim & \text{Categorical}(\boldsymbol{\phi}) \\
x_{i=1\ldots N} & \sim & \text{Categorical}(\boldsymbol{\theta}_{z_i})
\end{array}
$$

25.2 Examples

25.2.1 A financial model

Financial returns often behave differently in normal situations and during crisis times. A mixture model [2] for return data seems reasonable. Sometimes the model used is a jump-diffusion model, or as a mixture of two normal distributions. See Financial economics#Challenges and criticism for further context.



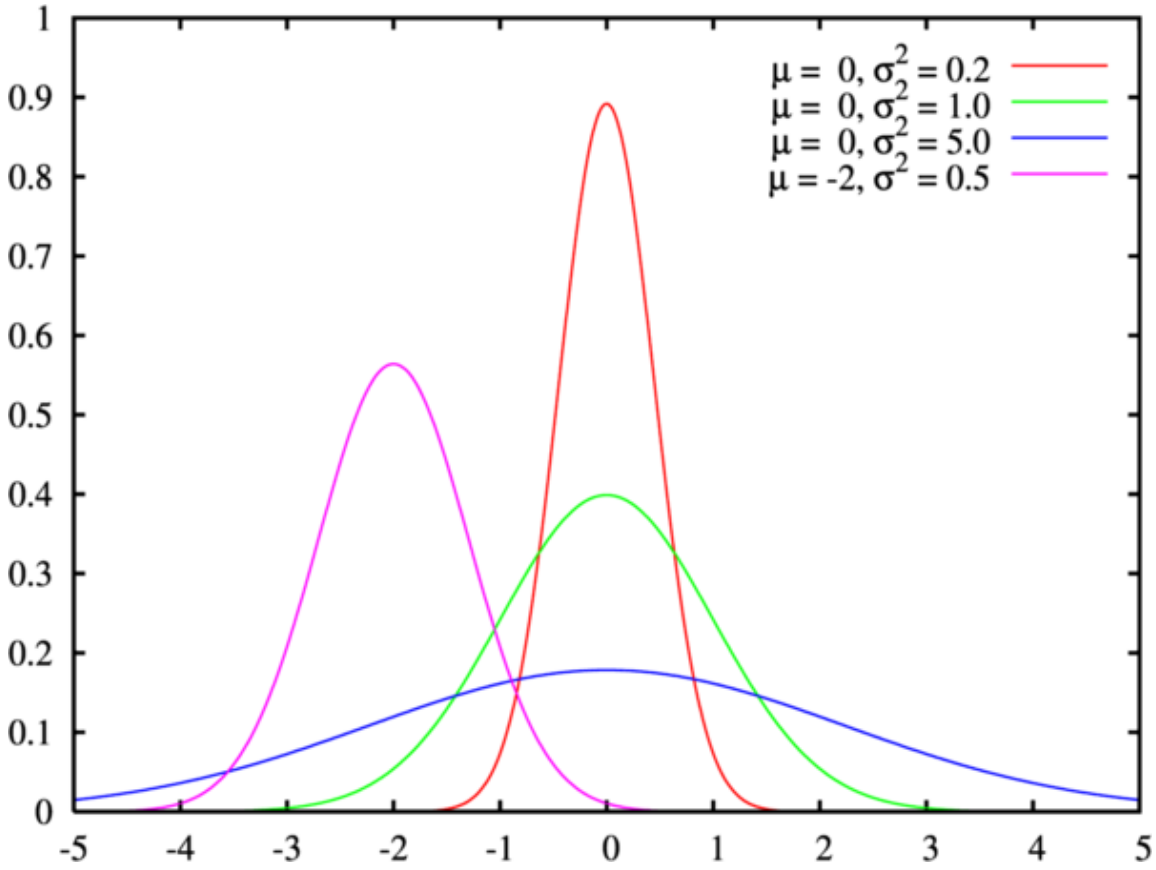
Bayesian categorical mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size K*; likewise for [V].*

25.2.2 House prices

Assume that we observe the prices of N different houses. Different types of houses in different neighborhoods will have vastly different prices, but the price of a particular type of house in a particular neighborhood (e.g., three-bedroom house in moderately upscale neighborhood) will tend to cluster fairly closely around the mean. One possible model of such prices would be to assume that the prices are accurately described by a mixture model with K different components, each distributed as a normal distribution with unknown mean and variance, with each component specifying a particular combination of house type/neighborhood. Fitting this model to observed prices, e.g., using the expectation-maximization algorithm, would tend to cluster the prices according to house type/neighborhood and reveal the spread of prices in each type/neighborhood. (Note that for values such as prices or incomes that are guaranteed to be positive and which tend to grow exponentially, a log-normal distribution might actually be a better model than a normal distribution.)

25.2.3 Topics in a document

Assume that a document is composed of N different words from a total vocabulary of size V, where each word corresponds to one of K possible topics. The distribution of such words could be modelled as a mixture of K different V-dimensional categorical distributions. A model of this sort is commonly termed a topic model. Note that expectation maximization applied to such a model will typically fail to produce realistic results, due (among other things) to the excessive number of parameters. Some sorts of additional assumptions are typically necessary to get



The normal distribution is plotted using different means and variances

good results. Typically two sorts of additional components are added to the model:

1. A prior distribution is placed over the parameters describing the topic distributions, using a Dirichlet distribution with a concentration parameter that is set significantly below 1, so as to encourage sparse distributions (where only a small number of words have significantly non-zero probabilities).
2. Some sort of additional constraint is placed over the topic identities of words, to take advantage of natural clustering.
    - For example, a Markov chain could be placed on the topic identities (i.e., the latent variables specifying the mixture component of each observation), corresponding to the fact that nearby words belong to similar topics. (This results in a hidden Markov model, specifically one where a prior distribution is placed over state transitions that favors transitions that stay in the same state.)
    - Another possibility is the latent Dirichlet allocation model, which divides up the words into D different documents and assumes that in each document only a small number of topics occur with any frequency.

25.2.4 Handwriting recognition

The following example is based on an example in Christopher M. Bishop, *Pattern Recognition and Machine Learning*.[3]

Imagine that we are given an $N \times N$ black-and-white image that is known to be a scan of a hand-written digit between 0 and 9, but we don't know which digit is written. We can create a mixture model with $K = 10$ different components, where each component is a vector of size N^2 of Bernoulli distributions (one per pixel). Such a model can be trained with the expectation-maximization algorithm on an unlabeled set of hand-written digits, and will effectively cluster the images according to the digit being written. The same model could then be used to recognize the digit of another

image simply by holding the parameters constant, computing the probability of the new image for each possible digit (a trivial calculation), and returning the digit that generated the highest probability.

25.2.5 Direct and indirect applications

The financial example above is one direct application of the mixture model, a situation in which we assume an underlying mechanism so that each observation belongs to one of some number of different sources or categories. This underlying mechanism may or may not, however, be observable. In this form of mixture, each of the sources is described by a component probability density function, and its mixture weight is the probability that an observation comes from this component.

In an indirect application of the mixture model we do not assume such a mechanism. The mixture model is simply used for its mathematical flexibilities. For example, a mixture of two normal distributions with different means may result in a density with two modes, which is not modeled by standard parametric distributions. Another example is given by the possibility of mixture distributions to model fatter tails than the basic Gaussian ones, so as to be a candidate for modeling more extreme events. When combined with dynamical consistency, this approach has been applied to financial derivatives valuation in presence of the volatility smile in the context of local volatility models. This defines our application.

25.2.6 Fuzzy image segmentation

In image processing and computer vision, traditional image segmentation models often assign to one pixel only one exclusive pattern. In fuzzy or soft segmentation, any pattern can have certain "ownership" over any single pixel. If the patterns are Gaussian, fuzzy segmentation naturally results in Gaussian mixtures. Combined with other analytic or geometric tools (e.g., phase transitions over diffusive boundaries), such spatially regularized mixture models could lead to more realistic and computationally efficient segmentation methods.[4]

25.3 Identifiability

Identifiability refers to the existence of a unique characterization for any one of the models in the class (family) being considered. Estimation procedure may not be well-defined and asymptotic theory may not hold if a model is not identifiable.

25.3.1 Example

Let J be the class of all binomial distributions with $n = 2$. Then a mixture of two members of J would have

$$p_0 = \pi(1-\theta_1)^2 + (1-\pi)(1-\theta_2)^2$$

$$p_1 = 2\pi\theta_1(1-\theta_1) + 2(1-\pi)\theta_2(1-\theta_2)$$

and $p_2 = 1 - p_0 - p_1$. Clearly, given p_0 and p_1, it is not possible to determine the above mixture model uniquely, as there are three parameters (π, θ_1, θ_2) to be determined.

25.3.2 Definition

Consider a mixture of parametric distributions of the same class. Let

$$J = \{f(\cdot;\theta) : \theta \in \Omega\}$$

be the class of all component distributions. Then the convex hull K of J defines the class of all finite mixture of distributions in J:

$$K = \left\{ p(\cdot) : p(\cdot) = \sum_{i=1}^{n} a_i f_i(\cdot;\theta_i), a_i > 0, \sum_{i=1}^{n} a_i = 1, f_i(\cdot;\theta_i) \in J \; \forall i, n \right\}$$

K is said to be identifiable if all its members are unique, that is, given two members *p* and *p′* in *K*, being mixtures of *k* distributions and *k′* distributions respectively in *J*, we have *p* = *p′* if and only if, first of all, *k* = *k′* and secondly we can reorder the summations such that *ai* = *ai′* and *fi* = *fi′* for all *i*.

25.4 Parameter estimation and system identification

Parametric mixture models are often used when we know the distribution *Y* and we can sample from *X*, but we would like to determine the *ai* and *θi* values. Such situations can arise in studies in which we sample from a population that is composed of several distinct subpopulations.

It is common to think of probability mixture modeling as a missing data problem. One way to understand this is to assume that the data points under consideration have "membership" in one of the distributions we are using to model the data. When we start, this membership is unknown, or missing. The job of estimation is to devise appropriate parameters for the model functions we choose, with the connection to the data points being represented as their membership in the individual model distributions.

A variety of approaches to the problem of mixture decomposition have been proposed, many of which focus on maximum likelihood methods such as expectation maximization (EM) or maximum *a posteriori* estimation (MAP). Generally these methods consider separately the question of parameter estimation and system identification, that is to say a distinction is made between the determination of the number and functional form of components within a mixture and the estimation of the corresponding parameter values. Some notable departures are the graphical methods as outlined in Tarter and Lock [5] and more recently minimum message length (MML) techniques such as Figueiredo and Jain [6] and to some extent the moment matching pattern analysis routines suggested by McWilliam and Loh (2009).[7]

25.4.1 Expectation maximization (EM)

Expectation maximization (EM) is seemingly the most popular technique used to determine the parameters of a mixture with an *a priori* given number of components. This is a particular way of implementing maximum likelihood estimation for this problem. EM is of particular appeal for finite normal mixtures where closed-form expressions are possible such as in the following iterative algorithm by Dempster *et al.* (1977)[8]

$$w_s^{(j+1)} = \frac{1}{N} \sum_{t=1}^{N} h_s^{(j)}(t)$$

$$\mu_s^{(j+1)} = \frac{\sum_{t=1}^{N} h_s^{(j)}(t) x^{(t)}}{\sum_{t=1}^{N} h_s^{(j)}(t)}$$

$$\Sigma_s^{(j+1)} = \frac{\sum_{t=1}^{N} h_s^{(j)}(t) [x^{(t)} - \mu_s^{(j+1)}][x^{(t)} - \mu_s^{(j+1)}]^\top}{\sum_{t=1}^{N} h_s^{(j)}(t)}$$

with the posterior probabilities

$$h_s^{(j)}(t) = \frac{w_s^{(j)} p_s(x^{(t)}; \mu_s^{(j)}, \Sigma_s^{(j)})}{\sum_{i=1}^{n} w_i^{(j)} p_i(x^{(t)}; \mu_i^{(j)}, \Sigma_i^{(j)})}.$$

Thus on the basis of the current estimate for the parameters, the conditional probability for a given observation $x^{(t)}$ being generated from state *s* is determined for each *t* = 1, …, *N* ; *N* being the sample size. The parameters are then updated such that the new component weights correspond to the average conditional probability and each component mean and covariance is the component specific weighted average of the mean and covariance of the entire sample.

Dempster[8] also showed that each successive EM iteration will not decrease the likelihood, a property not shared by other gradient based maximization techniques. Moreover EM naturally embeds within it constraints on the probability vector, and for sufficiently large sample sizes positive definiteness of the covariance iterates. This is a key advantage since explicitly constrained methods incur extra computational costs to check and maintain appropriate values. Theoretically EM is a first-order algorithm and as such converges slowly to a fixed-point solution. Redner and Walker (1984) make this point arguing in favour of superlinear and second order Newton and quasi-Newton methods and reporting slow convergence in EM on the basis of their empirical tests. They do concede that convergence in likelihood was rapid even if convergence in the parameter values themselves was not. The relative merits of EM and other algorithms vis-à-vis convergence have been discussed in other literature.[9]

Other common objections to the use of EM are that it has a propensity to spuriously identify local maxima, as well as displaying sensitivity to initial values.[10] One may address these problems by evaluating EM at several initial points in the parameter space but this is computationally costly and other approaches, such as the annealing EM method of Udea and Nakano (1998) (in which the initial components are essentially forced to overlap, providing a less heterogeneous basis for initial guesses), may be preferable.

Figueiredo and Jain [6] note that convergence to 'meaningless' parameter values obtained at the boundary (where regularity conditions breakdown, e.g., Ghosh and Sen (1985)) is frequently observed when the number of model components exceeds the optimal/true one. On this basis they suggest a unified approach to estimation and identification in which the initial n is chosen to greatly exceed the expected optimal value. Their optimization routine is constructed via a minimum message length (MML) criterion that effectively eliminates a candidate component if there is insufficient information to support it. In this way it is possible to systematize reductions in n and consider estimation and identification jointly.

The Expectation-maximization algorithm can be used to compute the parameters of a parametric mixture model distribution (the *ai* and *θi*). It is an iterative algorithm with two steps: an *expectation step* and a *maximization step*. Practical examples of EM and Mixture Modeling are included in the SOCR demonstrations.

The expectation step

With initial guesses for the parameters of our mixture model, "partial membership" of each data point in each constituent distribution is computed by calculating expectation values for the membership variables of each data point. That is, for each data point *xj* and distribution *Yi*, the membership value *yi, j* is:

$$y_{i,j} = \frac{a_i f_Y(x_j;\theta_i)}{f_X(x_j)}.$$

The maximization step

With expectation values in hand for group membership, plug-in estimates are recomputed for the distribution parameters.

The mixing coefficients *ai* are the means of the membership values over the *N* data points.

$$a_i = \frac{1}{N}\sum_{j=1}^{N} y_{i,j}$$

The component model parameters *θi* are also calculated by expectation maximization using data points *xj* that have been weighted using the membership values. For example, if θ is a mean μ

$$\mu_i = \frac{\sum_j y_{i,j} x_j}{\sum_j y_{i,j}}.$$

With new estimates for *ai* and the *θi*'s, the expectation step is repeated to recompute new membership values. The entire procedure is repeated until model parameters converge.

25.4.2 Markov chain Monte Carlo

As an alternative to the EM algorithm, the mixture model parameters can be deduced using posterior sampling as indicated by Bayes' theorem. This is still regarded as an incomplete data problem whereby membership of data points is the missing data. A two-step iterative procedure known as Gibbs sampling can be used.

The previous example of a mixture of two Gaussian distributions can demonstrate how the method works. As before, initial guesses of the parameters for the mixture model are made. Instead of computing partial memberships for each elemental distribution, a membership value for each data point is drawn from a Bernoulli distribution (that is, it will be assigned to either the first or the second Gaussian). The Bernoulli parameter θ is determined for each data point on the basis of one of the constituent distributions. Draws from the distribution generate membership associations for each data point. Plug-in estimators can then be used as in the M step of EM to generate a new set of mixture model parameters, and the binomial draw step repeated.

25.4.3 Moment matching

The method of moment matching is one of the oldest techniques for determining the mixture parameters dating back to Karl Pearson's seminal work of 1894. In this approach the parameters of the mixture are determined such that the composite distribution has moments matching some given value. In many instances extraction of solutions to the moment equations may present non-trivial algebraic or computational problems. Moreover numerical analysis by Day [11] has indicated that such methods may be inefficient compared to EM. Nonetheless there has been renewed interest in this method, e.g., Craigmile and Titterington (1998) and Wang.[12]

McWilliam and Loh (2009) consider the characterisation of a hyper-cuboid normal mixture copula in large dimensional systems for which EM would be computationally prohibitive. Here a pattern analysis routine is used to generate multivariate tail-dependencies consistent with a set of univariate and (in some sense) bivariate moments. The performance of this method is then evaluated using equity log-return data with Kolmogorov–Smirnov test statistics suggesting a good descriptive fit.

25.4.4 Spectral method

Some problems in mixture model estimation can be solved using spectral methods. In particular it becomes useful if data points *xi* are points in high-dimensional real space, and the hidden distributions are known to be log-concave (such as Gaussian distribution or Exponential distribution).

Spectral methods of learning mixture models are based on the use of Singular Value Decomposition of a matrix which contains data points. The idea is to consider the top k singular vectors, where k is the number of distributions to be learned. The projection of each data point to a linear subspace spanned by those vectors groups points originating from the same distribution very close together, while points from different distributions stay far apart.

One distinctive feature of the spectral method is that it allows us to prove that if distributions satisfy certain separation condition (e.g., not too close), then the estimated mixture will be very close to the true one with high probability.

25.4.5 Graphical Methods

Tarter and Lock [5] describe a graphical approach to mixture identification in which a kernel function is applied to an empirical frequency plot so to reduce intra-component variance. In this way one may more readily identify components having differing means. While this λ-method does not require prior knowledge of the number or functional form of the components its success does rely on the choice of the kernel parameters which to some extent implicitly embeds assumptions about the component structure.

25.4.6 Other methods

Some of them can even probably learn mixtures of heavy-tailed distributions including those with infinite variance (see links to papers below). In this setting, EM based methods would not work, since the Expectation step would diverge due to presence of outliers.

25.4.7 A simulation

To simulate a sample of size N that is from a mixture of distributions Fi, i=1 to n, with probabilities pi (sum= pi = 1):

1. Generate N random numbers from a categorical distribution of size n and probabilities pi for i= 1= to n. These tell you which of the Fi each of the N values will come from. Denote by mi the quantity of random numbers assigned to the i^{th} category.

2. For each i, generate mi random numbers from the Fi distribution.

25.5 Extensions

In a Bayesian setting, additional levels can be added to the graphical model defining the mixture model. For example, in the common latent Dirichlet allocation topic model, the observations are sets of words drawn from D different documents and the K mixture components represent topics that are shared across documents. Each document has a different set of mixture weights, which specify the topics prevalent in that document. All sets of mixture weights share common hyperparameters.

A very common extension is to connect the latent variables defining the mixture component identities into a Markov chain, instead of assuming that they are independent identically distributed random variables. The resulting model is termed a hidden Markov model and is one of the most common sequential hierarchical models. Numerous extensions of hidden Markov models have been developed; see the resulting article for more information.

25.6 History

Mixture distributions and the problem of mixture decomposition, that is the identification of its constituent components and the parameters thereof, has been cited in the literature as far back as 1846 (Quetelet in McLaughlan ,[10] 2000) although common reference is made to the work of Karl Pearson (1894) as the first author to explicitly address the decomposition problem in characterising non-normal attributes of forehead to body length ratios in female shore crab populations. The motivation for this work was provided by the zoologist Walter Frank Raphael Weldon who had speculated in 1893 (in Tarter and Lock[5]) that asymmetry in the histogram of these ratios could signal evolutionary divergence. Pearson's approach was to fit a univariate mixture of two normals to the data by choosing the five parameters of the mixture such that the empirical moments matched that of the model.

While his work was successful in identifying two potentially distinct sub-populations and in demonstrating the flexibility of mixtures as a moment matching tool, the formulation required the solution of a 9th degree (nonic) polynomial which at the time posed a significant computational challenge.

Subsequent works focused on addressing these problems, but it was not until the advent of the modern computer and the popularisation of Maximum Likelihood (ML) parameterisation techniques that research really took off.[13] Since that time there has been a vast body of research on the subject spanning areas such as Fisheries research, Agriculture, Botany, Economics, Medicine, Genetics, Psychology, Palaeontology, Electrophoresis, Finance, Sedimentology/Geology and Zoology.[14]

25.7 See also

25.7.1 Mixture

- Mixture density
- Mixture (probability)
- Flexible Mixture Model (FMM)

25.7.2 Hierarchical models

- Graphical model
- Hierarchical Bayes model

25.7.3 Outlier detection

- RANSAC

25.8 References

[1] Yu, Guoshen (2012). "Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity". *IEEE Transactions on Image Processing* **21** (5): 2481–2499. doi:10.1109/tip.2011.2176743.

[2] Dinov, ID. "Expectation Maximization and Mixture Modeling Tutorial". *California Digital Library*, Statistics Online Computational Resource, Paper EM_MM, http://repositories.cdlib.org/socr/EM_MM, December 9, 2008

[3] Bishop, Christopher (2006). *Pattern recognition and machine learning*. New York: Springer. ISBN 978-0-387-31073-2.

[4] Shen, Jianhong (Jackie) (2006). "A stochastic-variational model for soft Mumford-Shah segmentation". *Int'l J. Biomedical Imaging* **2006**: 2–16. doi:10.1155/IJBI/2006/92329.

[5] Tarter, Michael E. (1993), *Model Free Curve Estimation*, Chapman and Hall

[6] Figueiredo, M.A.T.; Jain, A.K. (March 2002). "Unsupervised Learning of Finite Mixture Models". *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (3): 381–396. doi:10.1109/34.990138.

[7] McWilliam, N.; Loh, K. (2008), *Incorporating Multidimensional Tail-Dependencies in the Valuation of Credit Derivatives (Working Paper)*

[8] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society, Series B* **39** (1): 1–38. JSTOR 2984875. CiteSeerX: 10.1.1.163.7580.

[9] Xu, L.; Jordan, M.I. (January 1996). "On Convergence Properties of the EM Algorithm for Gaussian Mixtures". *Neural Computation* **8** (1): 129–151. doi:10.1162/neco.1996.8.1.129.

[10] McLaughlan, G.J. (2000), *Finite Mixture Models*, Wiley

[11] Day, N. E. (1969). "Estimating the Components of a Mixture of Normal Distributions". *Biometrika* **56** (3): 463–474. doi:10.2307/2334652. JSTOR 2334652.

[12] Wang, J. (2001), "Generating daily changes in market variables using a multivariate mixture of normal distributions", *Proceedings of the 33rd winter conference on simulation* (IEEE Computer Society): 283–289

[13] McLaughlan, G.J. (1988), *Mixture Models: inference and applications to clustering*, Dekker

[14] Titterington, Smith & Makov 1985

25.9 Further reading

25.9.1 Books on mixture models

- Everitt, B.S.; Hand, D.J. (1981). *Finite mixture distributions*. Chapman & Hall. ISBN 0-412-22420-8.
- Lindsay, B. G. (1995). *Mixture Models: Theory, Geometry, and Applications*. NSF-CBMS Regional Conference Series in Probability and Statistics **5**. Hayward: Institute of Mathematical Statistics.
- Marin, J.M.; Mengersen, K.; Robert, C.P. (2011). "Bayesian modelling and inference on mixtures of distributions" (PDF). In Dey, D.; Rao, C.R. *Essential Bayesian models*. Handbook of statistics: Bayesian thinking - modeling and computation **25**. Elsevier. ISBN 9780444537324.
- McLachlan, G.J.; Peel, D. (2000). *Finite Mixture Models*. Wiley. ISBN 0-471-00626-2.

- Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 16.1. Gaussian Mixture Models and k-Means Clustering". *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.
- Titterington, D.; Smith, A.; Makov, U. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley. ISBN 0-471-90763-4.

25.9.2 Application of Gaussian mixture models

1. Reynolds, D.A.; Rose, R.C. (January 1995). "Robust text-independent speaker identification using Gaussian mixture speaker models". *IEEE Transactions on Speech and Audio Processing* **3** (1): 72–83. doi:10.1109/89.365379.
2. Permuter, H.; Francos, J.; Jermyn, I.H. (2003). *Gaussian mixture models of texture and colour for image database retrieval*. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP '03).The journal version
3. Lemke, Wolfgang (2005). *Term Structure Modeling and Estimation in a State Space Framework*. Springer Verlag. ISBN 978-3-540-28342-3.
4. Brigo, Damiano; Mercurio, Fabio (2001). *Displaced and Mixture Diffusions for Analytically-Tractable Smile Models*. Mathematical Finance — Bachelier Congress 2000. Proceedings. Springer Verlag.
5. Brigo, Damiano; Mercurio, Fabio (June 2002). "Lognormal-mixture dynamics and calibration to market volatility smiles". *International Journal of Theoretical and Applied Finance* **5** (4): 427. doi:10.1142/S0219024902001511.
6. Alexander, Carol (December 2004). "Normal mixture diffusion with uncertain volatility: Modelling short- and long-term smile effects" (PDF). *Journal of Banking & Finance* **28** (12): 2957–80. doi:10.1016/j.jbankfin.2003.10.017.
7. Stylianou, Yannis; Pantazis, Yannis; Calderero, Felipe; Larroy, Pedro; Severin, Francois; Schimke, Sascha; Bonal, Rolando; Matta, Federico; Valsamakis, Athanasios (2005). *GMM-Based Multimodal Biometric Verification* (PDF).

25.10 External links

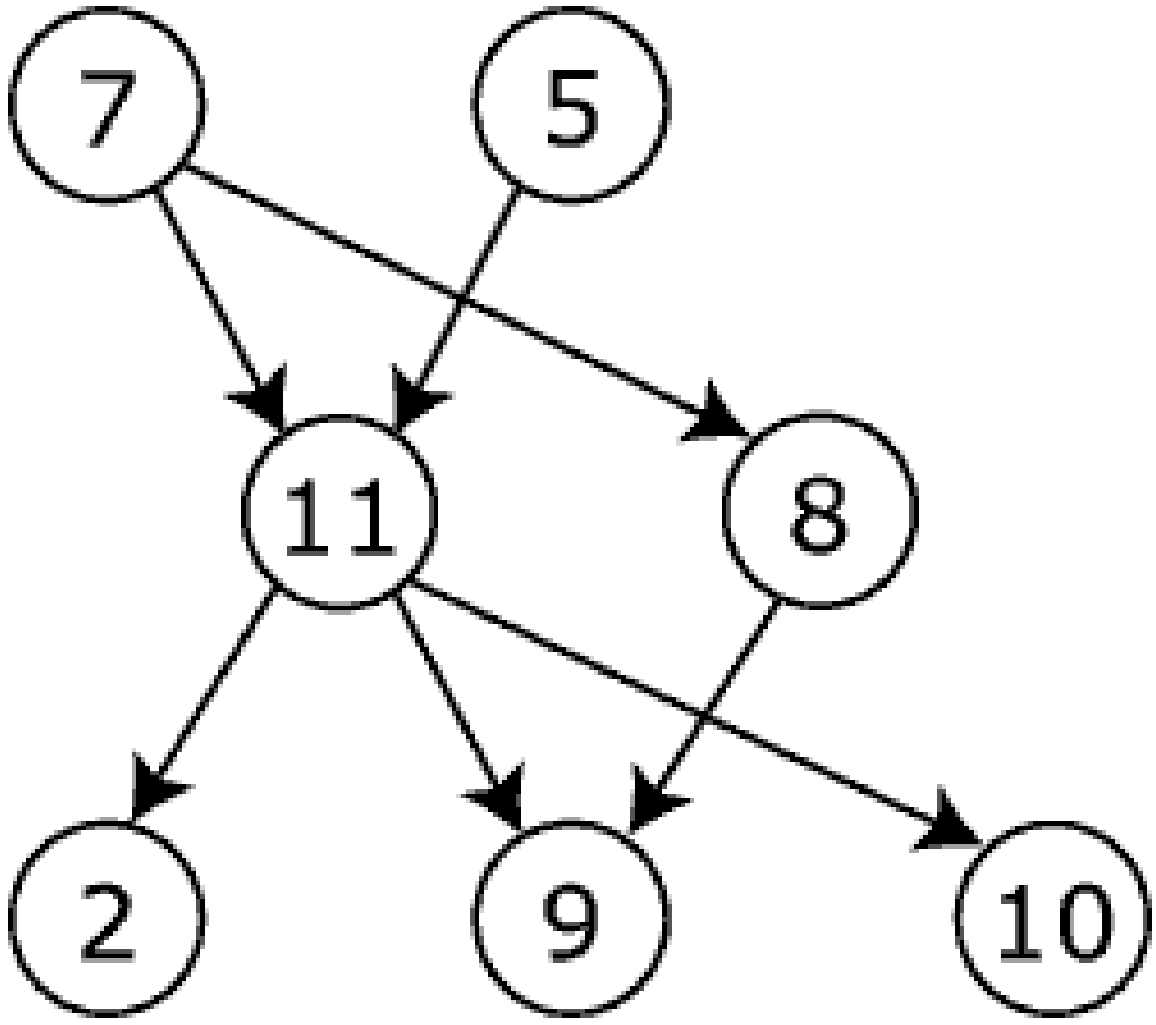
- Nielsen, Frank (23 March 2012). "k-MLE: A fast algorithm for learning statistical mixture models". arXiv:1203.5181.
- The SOCR demonstrations of EM and Mixture Modeling
- Mixture modelling page (and the Snob program for Minimum Message Length (MML) applied to finite mixture models), maintained by D.L. Dowe.
- PyMix — Python Mixture Package, algorithms and data structures for a broad variety of mixture model based data mining applications in Python
- scikit-learn.mixture.GMM — A Python package for learning Gaussian Mixture Models (and sampling from them), previously packaged with SciPy and now packaged as a SciKit
- GMM.m Matlab code for GMM Implementation
- GPUmix C++ implementation of Bayesian Mixture Models using EM and MCMC with 100x speed acceleration using GPGPU.
- Matlab code for GMM Implementation using EM algorithm
- jMEF: A Java open source library for learning and processing mixtures of exponential families (using duality with Bregman divergences). Includes a Matlab wrapper.
- Very Fast and clean C implementation of the Expectation Maximization (EM) algorithm for estimating Gaussian Mixture Models (GMMs).

Chapter 26

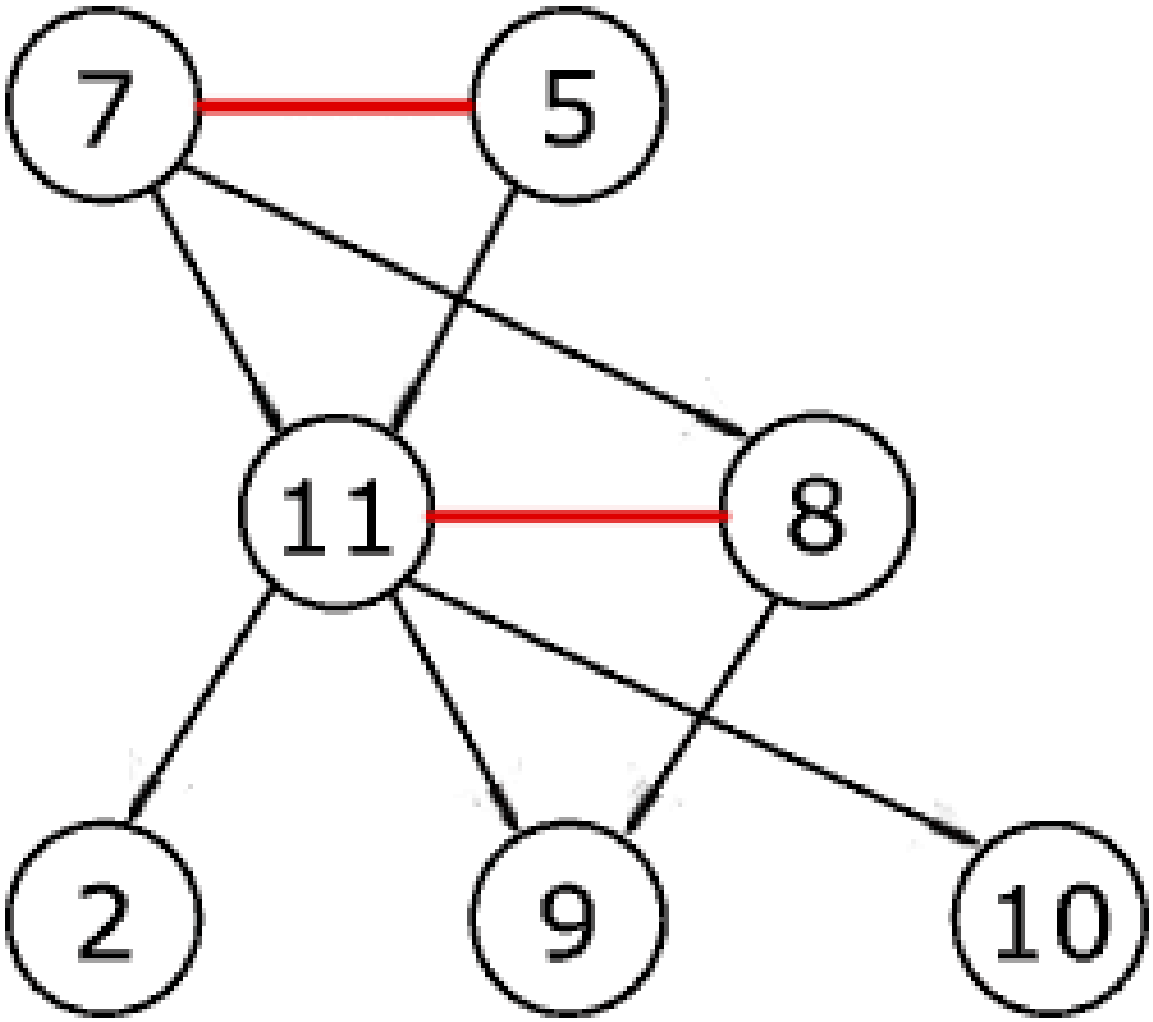
Moral graph

A **moral graph** is a concept in graph theory, used to find the equivalent undirected form of a directed acyclic graph. It is a key step of the junction tree algorithm, used in belief propagation on graphical models.

The moralized counterpart of a directed acyclic graph is formed by connecting nodes that have a common child, and then making all edges in the graph undirected. Equivalently, a moral graph of a directed acyclic graph G is an undirected graph in which each node of the original G is now connected to its Markov blanket. The name stems from the fact that, in a moral graph, two nodes that have a common child are required to be *married* by sharing an edge.



A directed acyclic graph.



The corresponding moral graph. The newly added arcs are shown in red in the moralized graph.

26.1 See also

- D-separation
- Tree decomposition

26.2 References

- Cowell, Robert G.; Dawid, A. Philip; Lauritzen, Steffen L., Spiegelhalter, David J. (1999). "3.2.1 Moralization". *Probabilistic Networks and Expert Systems*. Springer-Verlag New York. pp. 31–33. ISBN 0-387-98767-3.
- M. Studeny: On mathematical description of probabilistic conditional independence structures

Chapter 27

Naive Bayes classifier

In machine learning, **naive Bayes classifiers** are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s,[1]:488 and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate preprocessing, it is competitive in this domain with more advanced methods including support vector machines.[2] It also finds application in automatic medical diagnosis.[3]

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression,[1]:718 which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including **simple Bayes** and **independence Bayes**.[4] All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method;[4] Russell and Norvig note that "[naive Bayes] is sometimes called a **Bayesian classifier**, a somewhat careless usage that has prompted true Bayesians to call it the **idiot Bayes** model."[1]:482

27.1 Introduction

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3" in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers.[5] Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.[6]

An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

27.2 Probabilistic model

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing some n features (dependent variables), it assigns to this instance probabilities

$$p(C_k|x_1, \ldots, x_n)$$

for each of k possible outcomes or *classes*.[7]

The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}.$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C_k, x_1, \ldots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C_k, x_1, \ldots, x_n) &= p(C_k)\ p(x_1, \ldots, x_n|C_k) \\ &= p(C_k)\ p(x_1|C_k)\ p(x_2, \ldots, x_n|C_k, x_1) \\ &= p(C_k)\ p(x_1|C_k)\ p(x_2|C_k, x_1)\ p(x_3, \ldots, x_n|C_k, x_1, x_2) \\ &= p(C_k)\ p(x_1|C_k)\ p(x_2|C_k, x_1)\ \ldots p(x_n|C_k, x_1, x_2, x_3, \ldots, x_{n-1}) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$, given the category C . This means that

$$p(x_i|C_k, x_j) = p(x_i|C_k)$$

$$p(x_i|C_k, x_j, x_k) = p(x_i|C_k)$$

$$p(x_i|C_k, x_j, x_k, x_l) = p(x_i|C_k)$$

and so on, for $i \neq j, k, l$. Thus, the joint model can be expressed as

$$\begin{aligned} p(C_k|x_1, \ldots, x_n) &\propto p(C_k, x_1, \ldots, x_n) \\ &\propto p(C_k)\ p(x_1|C_k)\ p(x_2|C_k)\ p(x_3|C_k)\ \cdots \\ &\propto p(C_k) \prod_{i=1}^{n} p(x_i|C_k)\,. \end{aligned}$$

This means that under the above independence assumptions, the conditional distribution over the class variable C is:

$$p(C_k|x_1,\ldots,x_n) = \frac{1}{Z}p(C_k)\prod_{i=1}^{n}p(x_i|C_k)$$

where the evidence $Z = p(\mathbf{x})$ is a scaling factor dependent only on $x_1,\ldots,x_n$, that is, a constant if the values of the feature variables are known.

27.2.1 Constructing a classifier from the probability model

The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k)\prod_{i=1}^{n}p(x_i|C_k).$$

27.3 Parameter estimation and event models

A class' prior may be calculated by assuming equiprobable classes (i.e., priors = 1 / (number of classes)), or by calculating an estimate for the class probability from the training set (i.e., (prior for a given class) = (number of samples in the class) / (total number of samples)). To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set.[8]

The assumptions on distributions of features are called the *event model* of the Naive Bayes classifier. For discrete features like the ones encountered in document classification (include spam filtering), multinomial and Bernoulli distributions are popular. These assumptions lead to two distinct models, which are often confused.[9][10]

27.3.1 Gaussian naive Bayes

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contain a continuous attribute, x . We first segment the data by the class, and then compute the mean and variance of x in each class. Let μ_c be the mean of the values in x associated with class c, and let σ_c^2 be the variance of the values in x associated with class c. Then, the probability *distribution* of some value given a class, $p(x = v|c)$, can be computed by plugging v into the equation for a Normal distribution parameterized by μ_c and σ_c^2 . That is,

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}}\, e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

Another common technique for handling continuous values is to use binning to discretize the feature values, to obtain a new set of Bernoulli-distributed features; some literature in fact suggests that this is necessary to apply naive Bayes, but it is not, and the discretization may throw away discriminative information.[4]

27.3.2 Multinomial naive Bayes

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial $(p_1,\ldots,p_n)$ where p_i is the probability that event i occurs (or K such multinomials in the multiclass case). A feature vector $\mathbf{x} = (x_1,\ldots,x_n)$ is then a histogram, with x_i counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram **x** is given by

$$p(\mathbf{x}|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

The multinomial naive Bayes classifier becomes a linear classifier when expressed in log-space:[2]

$$\begin{aligned} \log p(C_k|\mathbf{x}) &\propto \log \left(p(C_k) \prod_{i=1}^{n} {p_{ki}}^{x_i} \right) \\ &= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki} \\ &= b + \mathbf{w}_k^\top \mathbf{x} \end{aligned}$$

where $b = \log p(C_k)$ and $w_{ki} = \log p_{ki}$.

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

Rennie *et al.* discuss problems with the multinomial assumption in the context of document classification and possible ways to alleviate those problems, including the use of tf–idf weights instead of raw term frequencies and document length normalization, to produce a naive Bayes classifier that is competitive with support vector machines.[2]

27.3.3 Bernoulli naive Bayes

In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks,[9] where binary term occurrence features are used rather than term frequencies. If x_i is a boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class C_k is given by[9]

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)}$$

where p_{ki} is the probability of class C_k generating the term w_i . This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

27.3.4 Semi-supervised parameter estimation

Given a way to train a naive Bayes classifier from labeled data, it's possible to construct a semi-supervised training algorithm that can learn from a combination of labeled and unlabeled data by running the supervised learning algorithm in a loop:[11]

> Given a collection $D = L \uplus U$ of labeled samples L and unlabeled samples U, start by training a naive Bayes classifier on L.
>
> Until convergence, do:
>
> > Predict class probabilities $P(C|x)$ for all examples x in D .
> >
> > Re-train the model based on the *probabilities* (not the labels) predicted in the previous step.

Convergence is determined based on improvement to the model likelihood $P(D|\theta)$, where θ denotes the parameters of the naive Bayes model.

This training algorithm is an instance of the more general expectation–maximization algorithm (EM): the prediction step inside the loop is the *E*-step of EM, while the re-training of naive Bayes is the *M*-step. The algorithm is formally justified by the assumption that the data are generated by a mixture model, and the components of this mixture model are exactly the classes of the classification problem.[11]

27.4 Discussion

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This helps alleviate problems stemming from the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features. While naive Bayes often fails to produce a good estimate for the correct class probabilities,[12] this may not be a requirement for many applications. For example, the naive Bayes classifier will make the correct MAP decision rule classification so long as the correct class is more probable than any other class. This is true regardless of whether the probability estimate is slightly, or even grossly inaccurate. In this manner, the overall classifier can be robust enough to ignore serious deficiencies in its underlying naive probability model.[3] Other reasons for the observed success of the naive Bayes classifier are discussed in the literature cited below.

27.4.1 Relation to logistic regression

In the case of discrete inputs (indicator or frequency features for discrete events), naive Bayes classifiers form a *generative-discriminative* pair with (multinomial) logistic regression classifiers: each naive Bayes classifier can be considered a way of fitting a probability model that optimizes the joint likelihood $p(C, \mathbf{x})$, while logistic regression fits the same probability model to optimize the conditional $p(C|\mathbf{x})$.[13]

The link between the two can be seen by observing that the decision function for naive Bayes (in the binary case) can be rewritten as "predict class C_1 if the odds of $p(C_1|\mathbf{x})$ exceed those of $p(C_2|\mathbf{x})$ ". Expressing this in log-space gives:

$$\log \frac{p(C_1|\mathbf{x})}{p(C_2|\mathbf{x})} = \log p(C_1|\mathbf{x}) - \log p(C_2|\mathbf{x}) > 0$$

The left-hand side of this equation is the log-odds, or *logit*, the quantity predicted by the linear model that underlies logistic regression. Since naive Bayes is also a linear model for the two "discrete" event models, it can be reparametrised as a linear function $b + \mathbf{w}^\top x > 0$. Obtaining the probabilities is then a matter of applying the logistic function to $b + \mathbf{w}^\top x$, or in the multiclass case, the softmax function.

Discriminative classifiers have lower asymptotic error than generative ones; however, research by Ng and Jordan has shown that in some practical cases naive Bayes can outperform logistic regression because it reaches its asymptotic error faster.[13]

27.5 Examples

27.5.1 Gender classification

Problem: classify whether a given person is a male or a female based on the measured features. The features include height, weight, and foot size.

Training

Example training set below.

The classifier created from the training set using a Gaussian distribution assumption would be (given variances are *unbiased* sample variances):

Let's say we have equiprobable classes so P(male)= P(female) = 0.5. This prior probability distribution might be based on our knowledge of frequencies in the larger population, or on frequency in the training set.

Testing

Below is a sample to be classified as a male or female.

We wish to determine which posterior is greater, male or female. For the classification as male the posterior is given by

$$posterior(male) = \frac{P(male)\, p(height|male)\, p(weight|male)\, p(footsize|male)}{evidence}$$

For the classification as female the posterior is given by

$$posterior(female) = \frac{P(female)\, p(height|female)\, p(weight|female)\, p(footsize|female)}{evidence}$$

The evidence (also termed normalizing constant) may be calculated:

$$evidence = P(male)\, p(height|male)\, p(weight|male)\, p(footsize|male)$$

$$+P(female)\, p(height|female)\, p(weight|female)\, p(footsize|female)$$

However, given the sample the evidence is a constant and thus scales both posteriors equally. It therefore does not affect classification and can be ignored. We now determine the probability distribution for the sex of the sample.

$$P(male) = 0.5$$

$$p(\text{height}|\text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789$$

where $\mu = 5.855$ and $\sigma^2 = 3.5033 \cdot 10^{-2}$ are the parameters of normal distribution which have been previously determined from the training set. Note that a value greater than 1 is OK here – it is a probability density rather than a probability, because height is a continuous variable.

$$p(\text{weight}|\text{male}) = 5.9881 \cdot 10^{-6}$$

$$p(\text{foot size}|\text{male}) = 1.3112 \cdot 10^{-3}$$

posterior numerator (male) = their product $= 6.1984 \cdot 10^{-9}$

$$P(\text{female}) = 0.5$$

$$p(\text{height}|\text{female}) = 2.2346 \cdot 10^{-1}$$

$$p(\text{weight}|\text{female}) = 1.6789 \cdot 10^{-2}$$

$$p(\text{foot size}|\text{female}) = 2.8669 \cdot 10^{-1}$$

posterior numerator (female) = their product $= 5.3778 \cdot 10^{-4}$

Since posterior numerator is greater in the female case, we predict the sample is female.

27.5.2 Document classification

Here is a worked example of naive Bayesian classification to the document classification problem. Consider the problem of classifying documents by their content, for example into spam and non-spam e-mails. Imagine that documents are drawn from a number of classes of documents which can be modelled as sets of words where the (independent) probability that the i-th word of a given document occurs in a document from class *C* can be written as

$$p(w_i|C)$$

(For this treatment, we simplify things further by assuming that words are randomly distributed in the document - that is, words are not dependent on the length of the document, position within the document with relation to other words, or other document-context.)

Then the probability that a given document *D* contains all of the words w_i , given a class *C*, is

$$p(D|C) = \prod_i p(w_i|C)$$

The question that we desire to answer is: "what is the probability that a given document *D* belongs to a given class *C*?" In other words, what is $p(C|D)$?

Now by definition

$$p(D|C) = \frac{p(D \cap C)}{p(C)}$$

and

$$p(C|D) = \frac{p(D \cap C)}{p(D)}$$

Bayes' theorem manipulates these into a statement of probability in terms of likelihood.

$$p(C|D) = \frac{p(C)}{p(D)}\, p(D|C)$$

Assume for the moment that there are only two mutually exclusive classes, *S* and ¬*S* (e.g. spam and not spam), such that every element (email) is in either one or the other;

$$p(D|S) = \prod_i p(w_i|S)$$

and

$$p(D|\neg S) = \prod_i p(w_i|\neg S)$$

Using the Bayesian result above, we can write:

$$p(S|D) = \frac{p(S)}{p(D)} \prod_i p(w_i|S)$$

$$p(\neg S|D) = \frac{p(\neg S)}{p(D)} \prod_i p(w_i|\neg S)$$

Dividing one by the other gives:

$$\frac{p(S|D)}{p(\neg S|D)} = \frac{p(S) \prod_i p(w_i|S)}{p(\neg S) \prod_i p(w_i|\neg S)}$$

Which can be re-factored as:

$$\frac{p(S|D)}{p(\neg S|D)} = \frac{p(S)}{p(\neg S)} \prod_i \frac{p(w_i|S)}{p(w_i|\neg S)}$$

Thus, the probability ratio p(*S* | *D*) / p(¬*S* | *D*) can be expressed in terms of a series of likelihood ratios. The actual probability p(*S* | *D*) can be easily computed from log (p(*S* | *D*) / p(¬*S* | *D*)) based on the observation that p(*S* | *D*) + p(¬*S* | *D*) = 1.

Taking the logarithm of all these ratios, we have:

$$\ln \frac{p(S|D)}{p(\neg S|D)} = \ln \frac{p(S)}{p(\neg S)} + \sum_i \ln \frac{p(w_i|S)}{p(w_i|\neg S)}$$

(This technique of "log-likelihood ratios" is a common technique in statistics. In the case of two mutually exclusive alternatives (such as this example), the conversion of a log-likelihood ratio to a probability takes the form of a sigmoid curve: see logit for details.)

Finally, the document can be classified as follows. It is spam if $p(S|D) > p(\neg S|D)$ (i.e., $\ln \frac{p(S|D)}{p(\neg S|D)} > 0$), otherwise it is not spam.

27.6 See also

- AODE
- Bayesian spam filtering
- Bayesian network
- Random naive Bayes
- Linear classifier
- Logistic regression
- Perceptron
- Take-the-best heuristic

27.7 References

[1] Russell, Stuart; Norvig, Peter (2003) [1995]. *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall. ISBN 978-0137903955.

[2] Rennie, J.; Shih, L.; Teevan, J.; Karger, D. (2003). *Tackling the poor assumptions of Naive Bayes classifiers* (PDF). ICML.

[3] Rish, Irina (2001). *An empirical study of the naive Bayes classifier* (PDF). IJCAI Workshop on Empirical Methods in AI.

[4] Hand, D. J.; Yu, K. (2001). "Idiot's Bayes — not so stupid after all?". *International Statistical Review* **69** (3): 385–399. doi:10.2307/1403452. ISSN 0306-7734.

[5] Zhang, Harry. *The Optimality of Naive Bayes* (PDF). FLAIRS2004 conference.

[6] Caruana, R.; Niculescu-Mizil, A. (2006). *An empirical comparison of supervised learning algorithms*. Proc. 23rd International Conference on Machine Learning. CiteSeerX: 10.1.1.122.5901.

[7] Narasimha Murty, M.; Susheela Devi, V. (2011). *Pattern Recognition: An Algorithmic Approach*. ISBN 0857294946.

[8] John, George H.; Langley, Pat (1995). *Estimating Continuous Distributions in Bayesian Classifiers*. Proc. Eleventh Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann. pp. 338–345.

[9] McCallum, Andrew; Nigam, Kamal (1998). *A comparison of event models for Naive Bayes text classification* (PDF). AAAI-98 workshop on learning for text categorization **752**.

[10] Metsis, Vangelis; Androutsopoulos, Ion; Paliouras, Georgios (2006). *Spam filtering with Naive Bayes—which Naive Bayes?*. Third conference on email and anti-spam (CEAS) **17**.

[11] Nigam, Kamal; McCallum, Andrew; Thrun, Sebastian; Mitchell, Tom (2000). "Learning to classify text from labeled and unlabeled documents using EM" (PDF). *Machine Learning*.

[12] Niculescu-Mizil, Alexandru; Caruana, Rich (2005). *Predicting good probabilities with supervised learning* (PDF). ICML. doi:10.1145/1102351.1102430.

[13] Ng, Andrew Y.; Jordan, Michael I. (2002). *On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes*. NIPS **14**.

27.7.1 Further reading

- Domingos, Pedro; Pazzani, Michael (1997). "On the optimality of the simple Bayesian classifier under zero-one loss". *Machine Learning* **29**: 103–137.
- Webb, G. I.; Boughton, J.; Wang, Z. (2005). "Not So Naive Bayes: Aggregating One-Dependence Estimators". *Machine Learning* (Springer) **58** (1): 5–24. doi:10.1007/s10994-005-4258-6.
- Mozina, M.; Demsar, J.; Kattan, M.; Zupan, B. (2004). *Nomograms for Visualization of Naive Bayesian Classifier* (PDF). Proc. PKDD-2004. pp. 337–348.
- Maron, M. E. (1961). "Automatic Indexing: An Experimental Inquiry". *JACM* **8** (3): 404–417. doi:10.1145/321075.321084.
- Minsky, M. (1961). *Steps toward Artificial Intelligence*. Proc. IRE **49** (1). pp. 8–30.

27.8 External links

- Book Chapter: Naive Bayes text classification, Introduction to Information Retrieval
- Naive Bayes for Text Classification with Unbalanced Classes
- Benchmark results of Naive Bayes implementations
- Hierarchical Naive Bayes Classifiers for uncertain data (an extension of the Naive Bayes classifier).

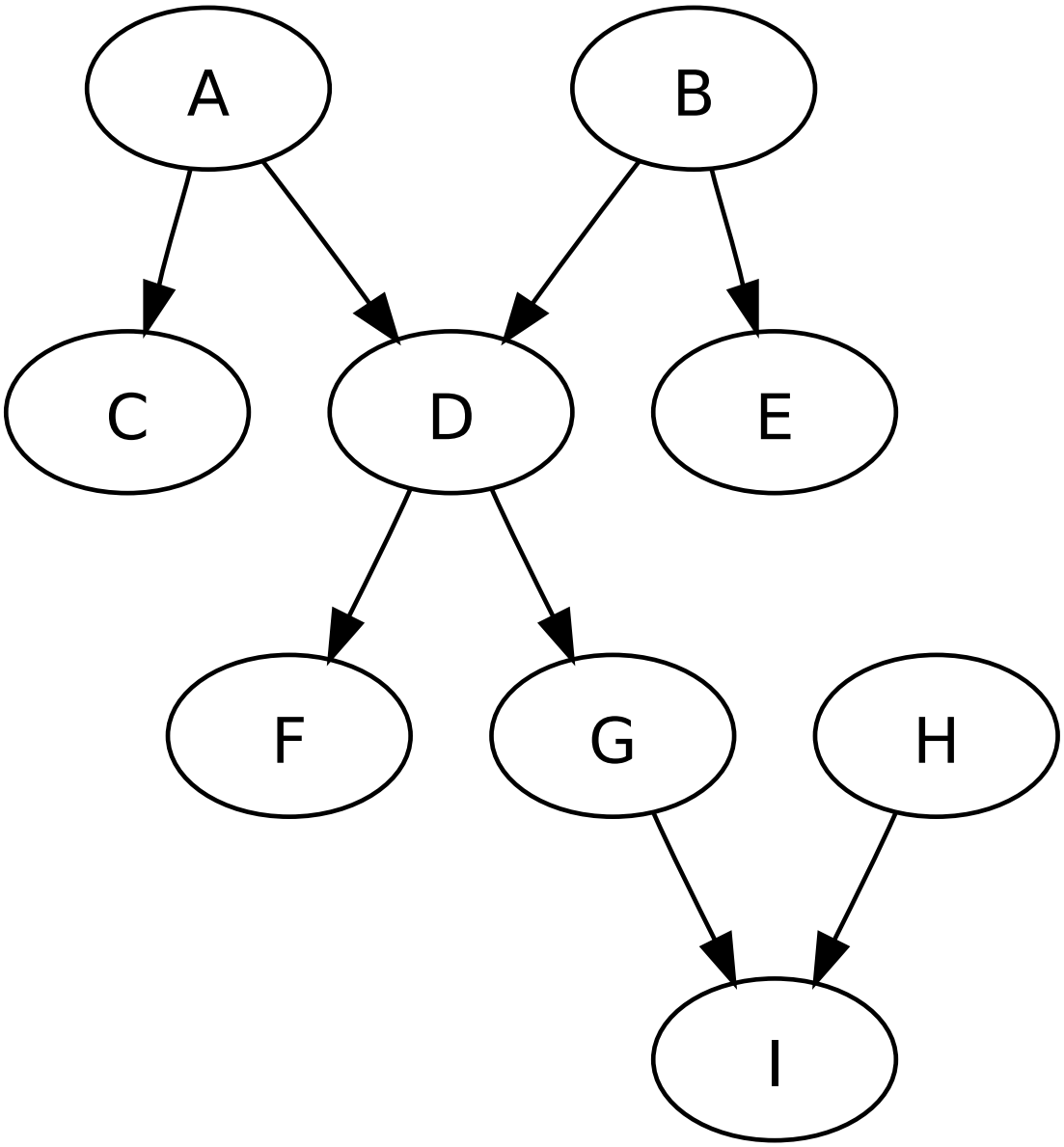
Software

- Naive Bayes classifiers are available in many general-purpose machine learning and NLP packages, including Apache Mahout, Mallet, NLTK, Orange, scikit-learn and Weka.
- IMSL Numerical Libraries Collections of math and statistical algorithms available in C/C++, Fortran, Java and C#/.NET. Data mining routines in the IMSL Libraries include a Naive Bayes classifier.
- Winnow content recommendation Open source Naive Bayes text classifier works with very small training and unbalanced training sets. High performance, C, any Unix.
- An interactive Microsoft Excel spreadsheet Naive Bayes implementation using VBA (requires enabled macros) with viewable source code.

- jBNC - Bayesian Network Classifier Toolbox
- Statistical Pattern Recognition Toolbox for Matlab.
- ifile - the first freely available (Naive) Bayesian mail/spam filter
- NClassifier - NClassifier is a .NET library that supports text classification and text summarization. It is a port of Classifier4J.
- Classifier4J - Classifier4J is a Java library designed to do text classification. It comes with an implementation of a Bayesian classifier.

Chapter 28

Polytree



A polytree

In mathematics, and more specifically in graph theory, a **polytree**[1] (also known as **oriented tree**[2][3] or **singly connected network**[4]) is a directed acyclic graph whose underlying undirected graph is a tree. In other words, if we replace its directed arcs with undirected edges, we obtain an undirected graph that is both connected and acyclic.

A polytree is an example of oriented graph.

The term *polytree* was coined in 1987 by Rebane and Pearl.[5]

28.1 Related structures

Every arborescence (a directed rooted tree, i.e. a directed acyclic graph in which there exists a single source node that has a unique path to every other node) is a polytree, but not every polytree is an arborescence. Every polytree is a multitree, a directed acyclic graph in which the subgraph reachable from any node forms a tree.

The reachability relationship among the nodes of a polytree forms a partial order that has order dimension at most three. If the order dimension is three, there must exist a subset of seven elements x, yi, and zi (for $i = 0, 1, 2$) such that, for each i, either $x \leq yi \geq zi$, or $x \geq yi \leq zi$, with these six inequalities defining the polytree structure on these seven elements.[6]

A fence or zigzag poset is a special case of a polytree in which the underlying tree is a path and the edges have orientations that alternate along the path. The reachability ordering in a polytree has also been called a *generalized fence*.[7]

28.2 Enumeration

The number of distinct polytrees on n unlabeled nodes, for $n = 1, 2, 3, ...$, is

1, 1, 3, 8, 27, 91, 350, 1376, 5743, 24635, 108968, 492180, ... (sequence A000238 in OEIS).

28.3 Sumner's conjecture

Sumner's conjecture, named after David Sumner, states that tournaments are universal graphs for polytrees, in the sense that every tournament with $2n - 2$ vertices contains every polytree with n vertices as a subgraph. Although it remains unsolved, it has been proven for all sufficiently large values of n.[8]

28.4 Applications

Polytrees have been used as a graphical model for probabilistic reasoning.[1] If a Bayesian network has the structure of a polytree, then belief propagation may be used to perform inference efficiently on it.[4][5]

The contour tree of a real-valued function on a vector space is a polytree that describes the level sets of the function. The nodes of the contour tree are the level sets that pass through a critical point of the function and the edges describe contiguous sets of level sets without a critical point. The orientation of an edge is determined by the comparison between the function values on the corresponding two level sets.[9]

28.5 See also

- Glossary of graph theory

28.6 Notes

[1] Dasgupta (1999).

[2] Harary & Sumner (1980).

[3] Simion (1991).

[4] Kim & Pearl (1983).

[5] Rebane & Pearl (1987).

[6] Trotter & Moore (1977).

[7] Ruskey, Frank (1989), "Transposition generation of alternating permutations", *Order* **6** (3): 227–233, doi:10.1007/BF00563523, MR 1048093

[8] Kühn et al. (2011).

[9] Carr et al. (2000).

28.7 References

- Carr, Hamish; Snoeyink, Jack; Axen, Ulrike (2000), "Computing contour trees in all dimensions", *in Proc. 11th ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pp. 918–926
- Dasgupta, Sanjoy (1999), "Learning polytrees", *in Proc. 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999), Stockholm, Sweden, July-August 1999* (PDF), pp. 134–141.
- Harary, Frank; Sumner, David (1980), "The dichromatic number of an oriented tree", *Journal of Combinatorics, Information & System Sciences* **5** (3): 184–187, MR 603363.
- Kim, Jin H.; Pearl, Judea (1983), "A computational model for causal and diagnostic reasoning in inference engines", *in Proc. 8th International Joint Conference on Artificial Intelligence (IJCAI 1983), Karlsruhe, Germany, August 1983* (PDF), pp. 190–193.
- Kühn, Daniela; Mycroft, Richard; Osthus, Deryk (2011), "A proof of Sumner's universal tournament conjecture for large tournaments", *Proceedings of the London Mathematical Society*, Third Series **102** (4): 731–766, arXiv:1010.4430, doi:10.1112/plms/pdq035, MR 2793448.
- Rebane, George; Pearl, Judea (1987), "The recovery of causal poly-trees from statistical data", *in Proc. 3rd Annual Conference on Uncertainty in Artificial Intelligence (UAI 1987), Seattle, WA, USA, July 1987* (PDF), pp. 222–228.
- Simion, Rodica (1991), "Trees with 1-factors and oriented trees", *Discrete Mathematics* **88** (1): 93–104, doi:10.1016/0012-365X(91)90061-6, MR 1099270.
- Trotter, William T., Jr.; Moore, John I., Jr. (1977), "The dimension of planar posets", *Journal of Combinatorial Theory, Series B* **22** (1): 54–67, doi:10.1016/0095-8956(77)90048-X.

Chapter 29

Probabilistic latent semantic analysis

Probabilistic latent semantic analysis (PLSA), also known as **probabilistic latent semantic indexing** (**PLSI**, especially in information retrieval circles) is a statistical technique for the analysis of two-mode and co-occurrence data. In effect, one can derive a low-dimensional representation of the observed variables in terms of their affinity to certain hidden variables, just as in latent semantic analysis, from which PLSA evolved.

Compared to standard latent semantic analysis which stems from linear algebra and downsizes the occurrence tables (usually via a singular value decomposition), probabilistic latent semantic analysis is based on a mixture decomposition derived from a latent class model.

29.1 Model

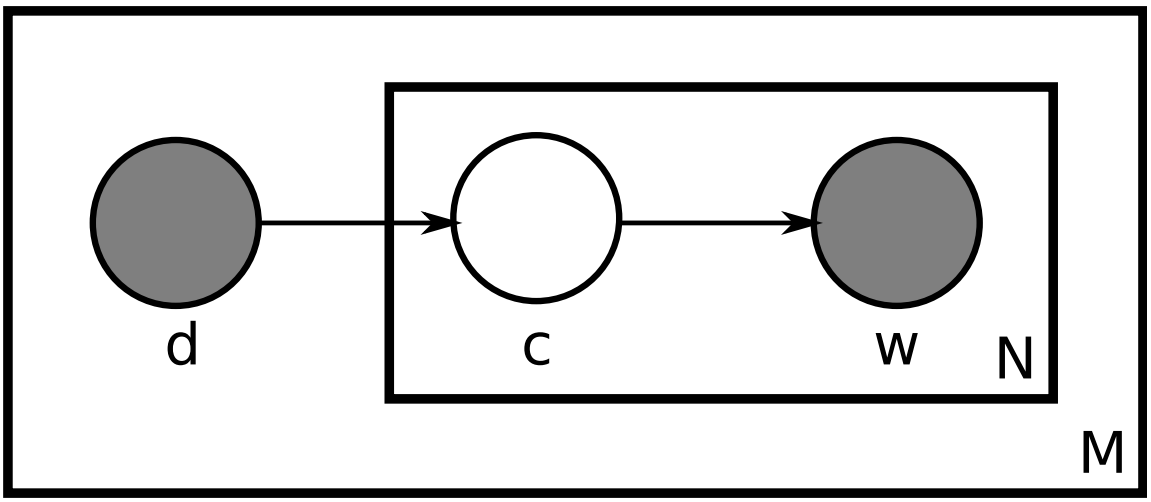


Plate notation representing the PLSA model ("asymmetric" formulation). d is the document index variable, c is a word's topic drawn from the document's topic distribution, $P(c|d)$, and w is a word drawn from the word distribution of this word's topic, $P(w|c)$. The d and w are observable variables, the topic c is a latent variable.

Considering observations in the form of co-occurrences (w, d) of words and documents, PLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions:

$$P(w, d) = \sum_c P(c)P(d|c)P(w|c) = P(d) \sum_c P(c|d)P(w|c)$$

being c the words' topic. The first formulation is the *symmetric* formulation, where w and d are both generated from the latent class c in similar ways (using the conditional probabilities $P(d|c)$ and $P(w|c)$), whereas the second formulation is the *asymmetric* formulation, where, for each document d , a latent class is chosen conditionally to the

document according to $P(c|d)$, and a word is then generated from that class according to $P(w|c)$. Although we have used words and documents in this example, the co-occurrence of any couple of discrete variables may be modelled in exactly the same way.

So, the number of parameters is equal to $cd + wc$. The number of parameters grows linearly with the number of documents. In addition, although PLSA is a generative model of the documents in the collection it is estimated on, it is not a generative model of new documents.

Their parameters are learned using the EM algorithm.

29.2 Application

PLSA may be used in a discriminative setting, via Fisher kernels.[1]

PLSA has applications in information retrieval and filtering, natural language processing, machine learning from text, and related areas.

It is reported that the aspect model used in the probabilistic latent semantic analysis has severe overfitting problems.[2]

In 2012, pLSA has also been used in the bioinformatics context, for prediction of Gene Ontology biomolecular annotations.[3]

29.3 Extensions

- Hierarchical extensions:
  - Asymmetric: MASHA ("Multinomial ASymmetric Hierarchical Analysis") [4]
  - Symmetric: HPLSA ("Hierarchical Probabilistic Latent Semantic Analysis") [5]
- Generative models: The following models have been developed to address an often-criticized shortcoming of PLSA, namely that it is not a proper generative model for new documents.
  - Latent Dirichlet allocation - adds a Dirichlet prior on the per-document topic distribution
- Higher-order data: Although this is rarely discussed in the scientific literature, PLSA extends naturally to higher order data (three modes and higher), i.e. it can model co-occurrences over three or more variables. In the symmetric formulation above, this is done simply by adding conditional probability distributions for these additional variables. This is the probabilistic analogue to non-negative tensor factorisation.

29.4 History

This is an example of a latent class model (see references therein), and it is related [6] to non-negative matrix factorization. The present terminology was coined in 1999 by Thomas Hofmann.[7]

29.5 References and notes

[1] Thomas Hofmann, *Learning the Similarity of Documents : an information-geometric approach to document retrieval and categorization*, Advances in Neural Information Processing Systems 12, pp-914-920, MIT Press, 2000

[2] Blei, David M.; Andrew Y. Ng; Michael I. Jordan (2003). "Latent Dirichlet Allocation" (PDF). *Journal of Machine Learning Research* **3**: 993–1022. doi:10.1162/jmlr.2003.3.4-5.993.

[3] *"Probabilistic Latent Semantic Analysis for prediction of Gene Ontology annotations"*, Marco Masseroli, Davide Chicco, Pietro Pinoli. IEEE WCCI 2012 - the 2012 IEEE World Congress on Computational Intelligence proceedings. Brisbane, Australia, June 2012. (.pdf)

[4] Alexei Vinokourov and Mark Girolami, A Probabilistic Framework for the Hierarchic Organisation and Classification of Document Collections, in *Information Processing and Management*, 2002

[5] Eric Gaussier, Cyril Goutte, Kris Popat and Francine Chen, A Hierarchical Model for Clustering and Categorising Documents, in "Advances in Information Retrieval -- Proceedings of the 24th BCS-IRSG European Colloquium on IR Research (ECIR-02)", 2002

[6] Chris Ding, Tao Li, Wei Peng (2006). "Nonnegative Matrix Factorization and Probabilistic Latent Semantic Indexing: Equivalence Chi-Square Statistic, and a Hybrid Method. AAAI 2006

[7] Thomas Hofmann, *Probabilistic Latent Semantic Indexing*, Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), 1999

29.6 See also

- Compound term processing
- Latent Dirichlet allocation
- Latent semantic analysis
- Pachinko allocation
- Vector space model

29.7 External links

- Probabilistic Latent Semantic Analysis
- Complete PLSA DEMO in C#

Chapter 30

Recursive Bayesian estimation

Recursive Bayesian estimation, also known as a **Bayes filter**, is a general probabilistic approach for estimating an unknown probability density function recursively over time using incoming measurements and a mathematical process model.

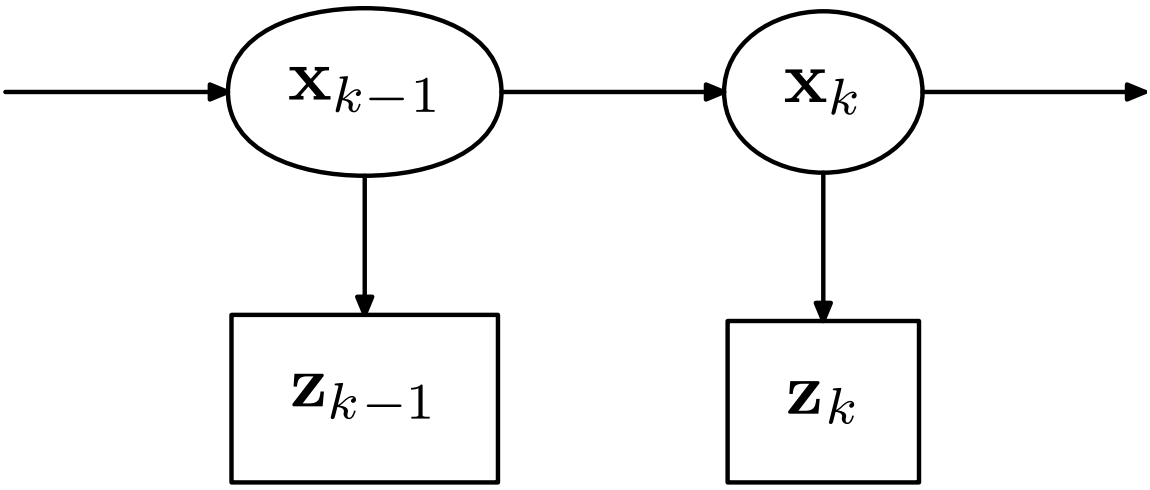
30.1 In robotics

A Bayes filter is an algorithm used in computer science for calculating the probabilities of multiple beliefs to allow a robot to infer its position and orientation. Essentially, Bayes filters allow robots to continuously update their most likely position within a coordinate system, based on the most recently acquired sensor data. This is a recursive algorithm. It consists of two parts: prediction and innovation. If the variables are linear and normally distributed the Bayes filter becomes equal to the Kalman filter.

In a simple example, a robot moving throughout a grid may have several different sensors that provide it with information about its surroundings. The robot may start out with certainty that it is at position (0,0). However, as it moves farther and farther from its original position, the robot has continuously less certainty about its position; using a Bayes filter, a probability can be assigned to the robot's belief about its current position, and that probability can be continuously updated from additional sensor information.

30.2 Model

The true state x is assumed to be an unobserved Markov process, and the measurements z are the observed states of a Hidden Markov Model (HMM). The following picture presents a Bayesian Network of a HMM.



Hidden Markov Model

Because of the Markov assumption, the probability of the current true state given the immediately previous one is conditionally independent of the other earlier states.

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots, \mathbf{x}_0) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

Similarly, the measurement at the *k*-th timestep is dependent only upon the current state, so is conditionally independent of all other states given the current state.

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_0) = p(\mathbf{z}_k|\mathbf{x}_k)$$

Using these assumptions the probability distribution over all states of the HMM can be written simply as:

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_k, \mathbf{z}_1, \ldots, \mathbf{z}_k) = p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i) p(\mathbf{x}_i|\mathbf{x}_{i-1}).$$

However, when using the Kalman filter to estimate the state $\mathbf{x}$, the probability distribution of interest is associated with the current states conditioned on the measurements up to the current timestep. (This is achieved by marginalising out the previous states and dividing by the probability of the measurement set.)

This leads to the *predict* and *update* steps of the Kalman filter written probabilistically. The probability distribution associated with the predicted state is the sum (integral) of the products of the probability distribution associated with the transition from the (k - 1)-th timestep to the *k*-th and the probability distribution associated with the previous state, over all possible x_{k-1} .

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1}$$

The probability distribution of update is proportional to the product of the measurement likelihood and the predicted state.

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} = \alpha\, p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$$

The denominator

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k$$

is constant relative to x , so we can always substitute it for a coefficient α , which can usually be ignored in practice. The numerator can be calculated and then simply normalized, since its integral must be unity.

30.3 Applications

- Kalman filter, a recursive Bayesian filter for multivariate normal distributions
- Particle filter, a sequential Monte Carlo (SMC) based technique, which models the PDF using a set of discrete points
- **Grid-based estimators**, which subdivide the PDF into a discrete grid

30.4 Sequential Bayesian filtering

Sequential Bayesian filtering is the extension of the Bayesian estimation for the case when the observed value changes in time. It is a method to estimate the real value of an observed variable that evolves in time.

The method is named:

filtering when we estimate the *current* value given past and current observations,

smoothing when estimating *past* values given present and past measures, and

prediction when estimating a probable *future* value given the present and the past measures.

The notion of Sequential Bayesian filtering is extensively used in control and robotics.

30.5 External links

- Arulampalam, M. Sanjeev; Maskell, Simon; Gordon, Neil (2002). "A Tutorial on Particle Filters for Online Non-linear/Non-Gaussian Bayesian Tracking". *IEEE Transactions on Signal Processing* **50**: 174–188. doi:10.1109/78.978374.
- Diard, Julien; Bessière, Pierre; Mazer, Emmanuel (2003). "A survey of probabilistic models, using the Bayesian Programming methodology as a unifying framework" (PDF). cogprints.org.
- Volkov, Alexander (2015). "Accuracy bounds of non-Gaussian Bayesian tracking in a NLOS environment". *Signal Processing* **108**: 498–508. doi:10.1016/j.sigpro.2014.10.025.
- Feynman-Kac models and interacting particle algorithms (a.k.a. Particle Filtering) Theoretical aspects and a list of application domains of particle filters
- Särkkä, Simo (2013). *Bayesian Filtering and Smoothing* (PDF). Cambridge University Press.

Chapter 31

Structured prediction

Structured prediction or **structured (output) learning** is an umbrella term for supervised machine learning techniques that involve predicting structured objects, rather than scalar discrete or real values.[1]

For example, the problem of translating a natural language sentence into a syntactic representation such as a parse tree can be seen as a structured prediction problem in which the structured output domain is the set of all possible parse trees.

Probabilistic graphical models form a large class of structured prediction models. In particular, Bayesian networks and random fields are popularly used to solve structured prediction problems in a wide variety of application domains including bioinformatics, natural language processing, speech recognition, and computer vision. Other algorithms and models for structured prediction include inductive logic programming, structured SVMs, Markov logic networks and constrained conditional models.

Similar to commonly used supervised learning techniques, structured prediction models are typically trained by means of observed data in which the true prediction value is used to adjust model parameters. Due to the complexity of the model and the interrelations of predicted variables the process of prediction using a trained model and of training itself is often computationally infeasible and approximate inference and learning methods are used.

31.1 Example: sequence tagging

Sequence tagging is a class of problems prevalent in natural language processing, where input data are often sequences (e.g. sentences of text). The sequence tagging problem appears in several guises, e.g. part-of-speech tagging and named entity recognition. In POS tagging, each word in a sequence must receive a "tag" (class label) that expresses its "type" of word:

> This DT
>
> is VBZ
>
> a DT
>
> tagged JJ
>
> sentence NN
>
> . .

The main challenge in this problem is to resolve ambiguity: the word "sentence" can also be a verb in English, and so can "tagged".

While this problem can be solved by simply performing classification of individual tokens, that approach does not take into account the empirical fact that tags do not occur independently; instead, each tag displays a strong conditional dependence on the tag of the previous word. This fact can be exploited in a sequence model such as a hidden Markov model or conditional random field[2] that predicts the entire tag sequence for a sentence, rather than just individual tags, by means of the Viterbi algorithm.

31.2 Structured perceptron

One of the easiest ways to understand algorithms for general structured prediction is the structured perceptron of Collins.[3] This algorithm combines the venerable perceptron algorithm for learning linear classifiers with an inference algorithm (classically the Viterbi algorithm when used on sequence data) and can be described abstractly as follows. First define a "joint feature function" $\Phi(\mathbf{x}, \mathbf{y})$ that maps a training sample $\mathbf{x}$ and a candidate prediction $\mathbf{y}$ to a vector of length n ($\mathbf{x}$ and $\mathbf{y}$ may have any structure; n is problem-dependent, but must be fixed for each model). Let GEN be a function that generates candidate predictions. Then:

Let $\mathbf{w}$ be a weight vector of length n

For a pre-determined number of iterations:

For each sample $\mathbf{x}$ in the training set with true output $\mathbf{t}$:

Make a prediction $\hat{\mathbf{y}} = \arg\max \{\mathbf{y} \in \text{GEN}(\mathbf{x})\} (\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}))$

Update $\mathbf{w}$, from $\hat{\mathbf{y}}$ to $\mathbf{t}$: $\mathbf{w}=\mathbf{w}+c(-\Phi(\mathbf{x}, \hat{\mathbf{y}})+ \Phi(\mathbf{x}, \mathbf{t}))$, c is learning rate

In practice, finding the argmax over GEN($\mathbf{x}$) will be done using an algorithm such as Viterbi or max-sum, rather than an exhaustive search through an exponentially large set of candidates.

The idea of learning is similar to multiclass perceptron.

31.3 See also

- Conditional random field
- Structured support vector machine
- Recurrent neural network, in particular Elman networks (SRNs)

31.4 References

[1] Gökhan BakIr, Ben Taskar, Thomas Hofmann, Bernhard Schölkopf, Alex Smola and SVN Vishwanathan (2007), Predicting Structured Data, MIT Press.

[2] Lafferty, J., McCallum, A., Pereira, F. (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data" (PDF). *Proc. 18th International Conf. on Machine Learning*. pp. 282–289.

[3] Collins, Michael (2002). *Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms* (PDF). Proc. EMNLP **10**.

- Noah Smith, Linguistic Structure Prediction, 2011.

31.5 External links

- Implementation of Collins structured perceptron

Chapter 32

Variable elimination

Variable elimination (VE) is a simple and general exact inference algorithm in probabilistic graphical models, such as Bayesian networks and Markov random fields.[1][2] It can be used for inference of maximum a posteriori (MAP) state or estimation of marginal distribution over a subset of variables. The algorithm has exponential time complexity, but could be efficient in practice for the low-treewidth graphs, if the proper elimination order is used.

32.1 Inference

The most common query type is in the form $p(X|E = e)$ where X and E are disjoint subsets of U , and E is observed taking value e . A basic algorithm to computing p(X|E = e) is called *variable elimination* (VE), first put forth in.[2]
Algorithm 1, called sum-out (SO), eliminates a single variable v from a set ϕ of potentials,[3] and returns the resulting set of potentials. The algorithm collect-relevant simply returns those potentials in ϕ involving variable v .

Algorithm 1 sum-out(v , ϕ)

Φ = collect-relevant(v , ϕ)

Ψ = the product of all potentials in Φ

$\tau = \sum_v \Psi$

return $(\phi - \Psi) \cup \{\tau\}$

Algorithm 2, taken from,[2] computes $p(X|E = e)$ from a discrete Bayesian network B. VE calls SO to eliminate variables one by one. More specifically, in Algorithm 2, ϕ is the set C of CPTs for B, X is a list of query variables, E is a list of observed variables, e is the corresponding list of observed values, and σ is an elimination ordering for variables $U - XE$, where XE denotes $X \cup E$.

Algorithm 2 VE(ϕ, X, E, e, σ)

Multiply evidence potentials with appropriate CPTs While σ is not empty

Remove the first variable v from σ

ϕ = sum-out (v, ϕ)

$p(X, E = e)$ = the product of all potentials $\Psi \in \phi$

return $p(X, E = e) / \sum_X p(X, E = e)$

32.2 References

[1] Zhang, N.L., Poole, D.: A Simple Approach to Bayesian Network Computations. In:7th Canadian Conference on Artificial Intelligence, pp. 171–178. Springer, New York(1994)

[2] Zhang, N.L., Poole, D.:A Simple Approach to Bayesian Network Computations.In: 7th Canadian Conference on Artificial Intelligence,pp. 171-−178. Springer, New York (1994)

[3] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge, MA (2009)

Chapter 33

Variable-order Bayesian network

Variable-order Bayesian network (VOBN) models provide an important extension of both the Bayesian network models and the variable-order Markov models. VOBN models are used in machine learning in general and have shown great potential in bioinformatics applications.[1][2] These models extend the widely used position weight matrix (PWM) models, Markov models, and Bayesian network (BN) models.

In contrast to the BN models, where each random variable depends on a fixed subset of random variables, in VOBN models these subsets may vary based on the specific realization of observed variables. The observed realizations are often called the context and, hence, VOBN models are also known as context-specific Bayesian networks.[3] The flexibility in the definition of conditioning subsets of variables turns out to be a real advantage in classification and analysis applications, as the statistical dependencies between random variables in a sequence of variables (not necessarily adjacent) may be taken into account efficiently, and in a position-specific and context-specific manner.

33.1 See also

- Markov chain
- Examples of Markov chains
- Variable order Markov models
- Markov process
- Markov chain Monte Carlo
- Semi-Markov process
- Artificial intelligence

33.2 References

[1] Ben-Gal, I.; Shani A., Gohr A., Grau J., Arviv S., Shmilovici A., Posch S. and Grosse I. (2005). "Identification of Transcription Factor Binding Sites with Variable-order Bayesian Networks". *Bioinformatics* **21** (11): 2657–2666. doi:10.1093/bioinformatics/bti410. PMID 15797905.

[2] Grau, J.; Ben-Gal I.; Posch S.; Grosse I. (2006). "VOMBAT: Prediction of Transcription Factor Binding Sites using Variable Order Bayesian Trees" (PDF). *Nucleic Acids Research* **34** (Web Server issue): 529–533. doi:10.1093/nar/gkl212. PMC 1538886. PMID 16845064.

[3] Boutilier, C.; Friedman N.; Goldszmidt M.; Koller D. (August 1–4, 1996, Reed College, Portland, Oregon, USA). "Context-specific independence in Bayesian networks". *In Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*: 115–123. Check date values in: |date= (help)

33.3 External links

- VOMBAT: https://www2.informatik.uni-halle.de:8443/VOMBAT/

Chapter 34

Variational Bayesian methods

For the method of approximation in quantum mechanics, see Variational method (quantum mechanics).

Variational Bayesian methods are a family of techniques for approximating intractable integrals arising in Bayesian inference and machine learning. They are typically used in complex statistical models consisting of observed variables (usually termed "data") as well as unknown parameters and latent variables, with various sorts of relationships among the three types of random variables, as might be described by a graphical model. As is typical in Bayesian inference, the parameters and latent variables are grouped together as "unobserved variables". Variational Bayesian methods are primarily used for two purposes:

1. To provide an analytical approximation to the posterior probability of the unobserved variables, in order to do statistical inference over these variables.

2. To derive a lower bound for the marginal likelihood (sometimes called the "evidence") of the observed data (i.e. the marginal probability of the data given the model, with marginalization performed over unobserved variables). This is typically used for performing model selection, the general idea being that a higher marginal likelihood for a given model indicates a better fit of the data by that model and hence a greater probability that the model in question was the one that generated the data. (See also the Bayes factor article.)

In the former purpose (that of approximating a posterior probability), variational Bayes is an alternative to Monte Carlo sampling methods — particularly, Markov chain Monte Carlo methods such as Gibbs sampling — for taking a fully Bayesian approach to statistical inference over complex distributions that are difficult to directly evaluate or sample from. In particular, whereas Monte Carlo techniques provide a numerical approximation to the exact posterior using a set of samples, Variational Bayes provides a locally-optimal, exact analytical solution to an approximation of the posterior.

Variational Bayes can be seen as an extension of the EM (expectation-maximization) algorithm from maximum a posteriori estimation (MAP estimation) of the single most probable value of each parameter to fully Bayesian estimation which computes (an approximation to) the entire posterior distribution of the parameters and latent variables. As in EM, it finds a set of optimal parameter values, and it has the same alternating structure as does EM, based on a set of interlocked (mutually dependent) equations that cannot be solved analytically.

For many applications, variational Bayes produces solutions of comparable accuracy to Gibbs sampling at greater speed. However, deriving the set of equations used to iteratively update the parameters often requires a large amount of work compared with deriving the comparable Gibbs sampling equations. This is the case even for many models that are conceptually quite simple, as is demonstrated below in the case of a basic non-hierarchical model with only two parameters and no latent variables.

34.1 Mathematical derivation of the mean-field approximation

In variational inference, the posterior distribution over a set of unobserved variables $\mathbf{Z} = \{Z_1 \dots Z_n\}$ given some data $\mathbf{X}$ is approximated by a variational distribution, $Q(\mathbf{Z})$:

$P(\mathbf{Z} \mid \mathbf{X}) \approx Q(\mathbf{Z})$.

The distribution $Q(\mathbf{Z})$ is restricted to belong to a family of distributions of simpler form than $P(\mathbf{Z} \mid \mathbf{X})$, selected with the intention of making $Q(\mathbf{Z})$ similar to the true posterior, $P(\mathbf{Z} \mid \mathbf{X})$. The lack of similarity is measured in terms of a dissimilarity function $d(Q; P)$ and hence inference is performed by selecting the distribution $Q(\mathbf{Z})$ that minimizes $d(Q; P)$.

The most common type of variational Bayes, known as *mean-field variational Bayes*, uses the Kullback–Leibler divergence (KL-divergence) of *P* from *Q* as the choice of dissimilarity function. This choice makes this minimization tractable. The KL-divergence is defined as

$$D_{\mathrm{KL}}(Q||P) = \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log \frac{Q(\mathbf{Z})}{P(\mathbf{Z} \mid \mathbf{X})}.$$

Note that *Q* and *P* are reversed from what one might expect. This use of reversed KL-divergence is conceptually similar to the expectation-maximization algorithm. (Using the KL-divergence in the other way produces the expectation propagation algorithm.)

The KL-divergence can be written as

$$D_{\mathrm{KL}}(Q||P) = \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log \frac{Q(\mathbf{Z})}{P(\mathbf{Z}, \mathbf{X})} + \log P(\mathbf{X}),$$

or

$$\log P(\mathbf{X}) = D_{\mathrm{KL}}(Q||P) - \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log \frac{Q(\mathbf{Z})}{P(\mathbf{Z}, \mathbf{X})} = D_{\mathrm{KL}}(Q||P) + \mathcal{L}(Q).$$

As the *log evidence* $\log P(\mathbf{X})$ is fixed with respect to Q , maximizing the final term $\mathcal{L}(Q)$ minimizes the KL divergence of P from Q . By appropriate choice of Q , $\mathcal{L}(Q)$ becomes tractable to compute and to maximize. Hence we have both an analytical approximation Q for the posterior $P(\mathbf{Z} \mid \mathbf{X})$, and a lower bound $\mathcal{L}(Q)$ for the evidence $\log P(\mathbf{X})$. The lower bound $\mathcal{L}(Q)$ is known as the (negative) *variational free energy* because it can also be expressed as an "energy" $\mathrm{E}_Q[\log P(\mathbf{Z}, \mathbf{X})]$ plus the entropy of Q .

34.2 In practice

The variational distribution $Q(\mathbf{Z})$ is usually assumed to factorize over some partition of the latent variables, i.e. for some partition of the latent variables $\mathbf{Z}$ into $\mathbf{Z}_1 \dots \mathbf{Z}_M$,

$$Q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z}_i \mid \mathbf{X})$$

It can be shown using the calculus of variations (hence the name "variational Bayes") that the "best" distribution q_j^* for each of the factors q_j (in terms of the distribution minimizing the KL divergence, as described above) can be expressed as:

$$q_j^*(\mathbf{Z}_j \mid \mathbf{X}) = \frac{e^{\mathrm{E}_{i \neq j}[\ln p(\mathbf{Z}, \mathbf{X})]}}{\int e^{\mathrm{E}_{i \neq j}[\ln p(\mathbf{Z}, \mathbf{X})]} \, d\mathbf{Z}_j}$$

where $\mathrm{E}_{i \neq j}[\ln p(\mathbf{Z}, \mathbf{X})]$ is the expectation of the logarithm of the joint probability of the data and latent variables, taken over all variables not in the partition.

In practice, we usually work in terms of logarithms, i.e.:

$$\ln q_j^*(\mathbf{Z}_j \mid \mathbf{X}) = \mathrm{E}_{i \neq j}[\ln p(\mathbf{Z}, \mathbf{X})] + \text{constant}$$

The constant in the above expression is related to the normalizing constant (the denominator in the expression above for q_j^*) and is usually reinstated by inspection, as the rest of the expression can usually be recognized as being a known type of distribution (e.g. Gaussian, gamma, etc.).

Using the properties of expectations, the expression $\mathrm{E}_{i \neq j}[\ln p(\mathbf{Z}, \mathbf{X})]$ can usually be simplified into a function of the fixed hyperparameters of the prior distributions over the latent variables and of expectations (and sometimes higher moments such as the variance) of latent variables not in the current partition (i.e. latent variables not included in $\mathbf{Z}_j$). This creates circular dependencies between the parameters of the distributions over variables in one partition and the expectations of variables in the other partitions. This naturally suggests an iterative algorithm, much like EM (the expectation-maximization algorithm), in which the expectations (and possibly higher moments) of the latent variables are initialized in some fashion (perhaps randomly), and then the parameters of each distribution are computed in turn using the current values of the expectations, after which the expectation of the newly computed distribution is set appropriately according to the computed parameters. An algorithm of this sort is guaranteed to converge.[1] Furthermore, if the distributions in question are part of the exponential family, which is usually the case, convergence will be to a global maximum, since the exponential family is convex.[2]

In other words, for each of the partitions of variables, by simplifying the expression for the distribution over the partition's variables and examining the distribution's functional dependency on the variables in question, the family of the distribution can usually be determined (which in turn determines the value of the constant). The formula for the distribution's parameters will be expressed in terms of the prior distributions' hyperparameters (which are known constants), but also in terms of expectations of functions of variables in other partitions. Usually these expectations can be simplified into functions of expectations of the variables themselves (i.e. the means); sometimes expectations of squared variables (which can be related to the variance of the variables), or expectations of higher powers (i.e. higher moments) also appear. In most cases, the other variables' distributions will be from known families, and the formulas for the relevant expectations can be looked up. However, those formulas depend on those distributions' parameters, which depend in turn on the expectations about other variables. The result is that the formulas for the parameters of each variable's distributions can be expressed as a series of equations with mutual, nonlinear dependencies among the variables. Usually, it is not possible to solve this system of equations directly. However, as described above, the dependencies suggest a simple iterative algorithm, which in most cases is guaranteed to converge. An example will make this process clearer.

34.3 A basic example

Consider a simple non-hierarchical Bayesian model consisting of a set of i.i.d. observations from a Gaussian distribution, with unknown mean and variance.[3] In the following, we work through this model in great detail to illustrate the workings of the variational Bayes method.

For mathematical convenience, in the following example we work in terms of the precision — i.e. the reciprocal of the variance (or in a multivariate Gaussian, the inverse of the covariance matrix) — rather than the variance itself. (From a theoretical standpoint, precision and variance are equivalent since there is a one-to-one correspondence between the two.)

34.3.1 The mathematical model

We place conjugate prior distributions on the unknown mean and variance, i.e. the mean also follows a Gaussian distribution while the precision follows a gamma distribution. In other words:

$$\begin{aligned} \mu &\sim \mathcal{N}(\mu_0, (\lambda_0 \tau)^{-1}) \\ \tau &\sim \text{Gamma}(a_0, b_0) \\ \{x_1, \ldots, x_N\} &\sim \mathcal{N}(\mu, \tau^{-1}) \\ N &= \text{points data of number} \end{aligned}$$

We are given N data points $\mathbf{X} = \{x_1, \ldots, x_N\}$ and our goal is to infer the posterior distribution $q(\mu, \tau) = p(\mu, \tau \mid x_1, \ldots, x_N)$ of the parameters μ and τ .

The hyperparameters μ_0 , λ_0 , a_0 and b_0 are fixed, given values. They can be set to small positive numbers to give broad prior distributions indicating ignorance about the prior distributions of μ and τ .

34.3.2 The joint probability

The joint probability of all variables can be rewritten as

$$p(\mathbf{X}, \mu, \tau) = p(\mathbf{X} \mid \mu, \tau)p(\mu \mid \tau)p(\tau)$$

where the individual factors are

$$\begin{aligned} p(\mathbf{X} \mid \mu, \tau) &= \prod_{n=1}^{N} \mathcal{N}(x_n \mid \mu, \tau^{-1}) \\ p(\mu \mid \tau) &= \mathcal{N}(\mu \mid \mu_0, (\lambda_0 \tau)^{-1}) \\ p(\tau) &= \text{Gamma}(\tau \mid a_0, b_0) \end{aligned}$$

where

$$\begin{aligned} \mathcal{N}(x \mid \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \\ \text{Gamma}(\tau \mid a, b) &= \frac{1}{\Gamma(a)} b^a \tau^{a-1} e^{-b\tau} \end{aligned}$$

34.3.3 Factorized approximation

Assume that $q(\mu, \tau) = q(\mu)q(\tau)$, i.e. that the posterior distribution factorizes into independent factors for μ and τ . This type of assumption underlies the variational Bayesian method. The true posterior distribution does not in fact factor this way (in fact, in this simple case, it is known to be a Gaussian-gamma distribution), and hence the result we obtain will be an approximation.

34.3.4 Derivation of q(μ)

Then

$$
\begin{aligned}
\ln q^*_\mu(\mu) &= \mathrm{E}_\tau\left[\ln p(\mathbf{X} \mid \mu, \tau) + \ln p(\mu \mid \tau) + \ln p(\tau)\right] + C \\
&= \mathrm{E}_\tau\left[\ln p(\mathbf{X} \mid \mu, \tau)\right] + \mathrm{E}_\tau\left[\ln p(\mu \mid \tau)\right] + \mathrm{E}_\tau\left[\ln p(\tau)\right] + C \\
&= \mathrm{E}_\tau\left[\ln \prod_{n=1}^{N} \mathcal{N}(x_n \mid \mu, \tau^{-1})\right] + \mathrm{E}_\tau\left[\ln \mathcal{N}(\mu \mid \mu_0, (\lambda_0\tau)^{-1})\right] + C_2 \\
&= \mathrm{E}_\tau\left[\ln \prod_{n=1}^{N} \sqrt{\frac{\tau}{2\pi}} e^{-\frac{(x_n-\mu)^2\tau}{2}}\right] + \mathrm{E}_\tau\left[\ln \sqrt{\frac{\lambda_0\tau}{2\pi}} e^{-\frac{(\mu-\mu_0)^2\lambda_0\tau}{2}}\right] + C_2 \\
&= \mathrm{E}_\tau\left[\sum_{n=1}^{N}\left(\frac{1}{2}(\ln\tau - \ln 2\pi) - \frac{(x_n-\mu)^2\tau}{2})\right)\right] + \mathrm{E}_\tau\left[\frac{1}{2}(\ln\lambda_0 + \ln\tau - \ln 2\pi) - \frac{(\mu-\mu_0)^2\lambda_0\tau}{2}\right] + C_2 \\
&= \mathrm{E}_\tau\left[\sum_{n=1}^{N} -\frac{(x_n-\mu)^2\tau}{2}\right] + \mathrm{E}_\tau\left[-\frac{(\mu-\mu_0)^2\lambda_0\tau}{2}\right] + \mathrm{E}_\tau\left[\sum_{n=1}^{N}\frac{1}{2}(\ln\tau - \ln 2\pi)\right] + \mathrm{E}_\tau\left[\frac{1}{2}(\ln\lambda_0 + \ln\tau - \ln 2\pi)\right] + C_2 \\
&= \mathrm{E}_\tau\left[\sum_{n=1}^{N} -\frac{(x_n-\mu)^2\tau}{2}\right] + \mathrm{E}_\tau\left[-\frac{(\mu-\mu_0)^2\lambda_0\tau}{2}\right] + C_3 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\left\{\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right\} + C_3
\end{aligned}
$$

In the above derivation, C , C_2 and C_3 refer to values that are constant with respect to μ . Note that the term $\mathrm{E}_\tau[\ln p(\tau)]$ is not a function of μ and will have the same value regardless of the value of μ . Hence in line 3 we can absorb it into the constant term at the end. We do the same thing in line 7.

The last line is simply a quadratic polynomial in μ . Since this is the logarithm of $q^*_\mu(\mu)$, we can see that $q^*_\mu(\mu)$ itself is a Gaussian distribution.

With a certain amount of tedious math (expanding the squares inside of the braces, separating out and grouping the terms involving μ and μ^2 and completing the square over μ), we can derive the parameters of the Gaussian distribution:

$$
\begin{aligned}
\ln q_\mu^*(\mu) &= -\frac{\mathrm{E}_\tau[\tau]}{2}\{\sum_{n=1}^{N}(x_n-\mu)^2+\lambda_0(\mu-\mu_0)^2\}+C_3 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\{\sum_{n=1}^{N}(x_n^2-2x_n\mu+\mu^2)+\lambda_0(\mu^2-2\mu_0\mu+\mu_0^2)\}+C_3 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\{(\sum_{n=1}^{N}x_n^2)-2(\sum_{n=1}^{N}x_n)\mu+(\sum_{n=1}^{N}\mu^2)+\lambda_0\mu^2-2\lambda_0\mu_0\mu+\lambda_0\mu_0^2\}+C_3 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\{(\lambda_0+N)\mu^2-2(\lambda_0\mu_0+\textstyle\sum_{n=1}^{N}x_n)\mu+(\sum_{n=1}^{N}x_n^2)+\lambda_0\mu_0^2\}+C_3 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\{(\lambda_0+N)\mu^2-2(\lambda_0\mu_0+\textstyle\sum_{n=1}^{N}x_n)\mu\}+C_4 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\left\{(\lambda_0+N)\mu^2-2\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}(\lambda_0+N)\mu\right\}+C_4 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\left\{(\lambda_0+N)\left(\mu^2-2\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\mu\right)\right\}+C_4 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\left\{(\lambda_0+N)\left(\mu^2-2\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\mu+\left(\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\right)^2-\left(\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\right)^2\right)\right\}+C_4 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\left\{(\lambda_0+N)\left(\mu^2-2\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\mu+\left(\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\right)^2\right)\right\}+C_5 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\left\{(\lambda_0+N)\left(\mu-\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\right)^2\right\}+C_5 \\
&= -\frac{1}{2}\left\{(\lambda_0+N)\,\mathrm{E}_\tau[\tau]\left(\mu-\frac{\lambda_0\mu_0+\sum_{n=1}^{N}x_n}{\lambda_0+N}\right)^2\right\}+C_5
\end{aligned}
$$

Note that all of the above steps can be shortened by using the formula for the sum of two quadratics.

In other words:

$$
\begin{aligned}
q_\mu^*(\mu) &\sim \mathcal{N}(\mu \mid \mu_N, \lambda_N^{-1}) \\
\mu_N &= \frac{\lambda_0\mu_0+N\bar{x}}{\lambda_0+N} \\
\lambda_N &= (\lambda_0+N)\,\mathrm{E}[\tau] \\
\bar{x} &= \frac{1}{N}\sum_{n=1}^{N}x_n
\end{aligned}
$$

34.3.5 Derivation of q(τ)

The derivation of $q_\tau^*(\tau)$ is similar to above, although we omit some of the details for the sake of brevity.

$$
\begin{aligned}
\ln q_\tau^*(\tau) &= \mathrm{E}_\mu[\ln p(\mathbf{X} \mid \mu, \tau) + \ln p(\mu \mid \tau)] + \ln p(\tau) + \text{constant} \\
&= (a_0-1)\ln\tau - b_0\tau + \frac{1}{2}\ln\tau + \frac{N}{2}\ln\tau - \frac{\tau}{2}\,\mathrm{E}_\mu[\sum_{n=1}^{N}(x_n-\mu)^2+\lambda_0(\mu-\mu_0)^2] + \text{constant}
\end{aligned}
$$

Exponentiating both sides, we can see that $q_\tau^*(\tau)$ is a gamma distribution. Specifically:

$$
\begin{aligned}
q_\tau^*(\tau) &\sim \text{Gamma}(\tau \mid a_N, b_N) \\
a_N &= a_0 + \frac{N+1}{2} \\
b_N &= b_0 + \frac{1}{2}\,\mathrm{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right]
\end{aligned}
$$

34.3.6 Algorithm for computing the parameters

Let us recap the conclusions from the previous sections:

$$
\begin{aligned}
q_\mu^*(\mu) &\sim \mathcal{N}(\mu \mid \mu_N, \lambda_N^{-1}) \\
\mu_N &= \frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N} \\
\lambda_N &= (\lambda_0 + N)\,\mathrm{E}[\tau] \\
\bar{x} &= \frac{1}{N}\sum_{n=1}^{N} x_n
\end{aligned}
$$

and

$$
\begin{aligned}
q_\tau^*(\tau) &\sim \text{Gamma}(\tau \mid a_N, b_N) \\
a_N &= a_0 + \frac{N+1}{2} \\
b_N &= b_0 + \frac{1}{2}\,\mathrm{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right]
\end{aligned}
$$

In each case, the parameters for the distribution over one of the variables depend on expectations taken with respect to the other variable. We can expand the expectations, using the standard formulas for the expectations of moments of the Gaussian and gamma distributions:

$$
\begin{aligned}
\mathrm{E}[\tau \mid a_N, b_N] &= \frac{a_N}{b_N} \\
\mathrm{E}[\mu \mid \mu_N, \lambda_N^{-1}] &= \mu_N \\
\mathrm{E}\left[X^2\right] &= \text{Var}(X) + (\mathrm{E}[X])^2 \\
\mathrm{E}[\mu^2 \mid \mu_N, \lambda_N^{-1}] &= \lambda_N^{-1} + \mu_N^2
\end{aligned}
$$

Applying these formulas to the above equations is trivial in most cases, but the equation for b_N takes more work:

$$
\begin{aligned}
b_N &= b_0 + \frac{1}{2}\,\mathrm{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right] \\
&= b_0 + \frac{1}{2}\,\mathrm{E}_\mu\left[(\lambda_0+N)\mu^2 - 2(\lambda_0\mu_0 + \textstyle\sum_{n=1}^{N} x_n)\mu + (\sum_{n=1}^{N} x_n^2) + \lambda_0\mu_0^2\right] \\
&= b_0 + \frac{1}{2}\left[(\lambda_0+N)\,\mathrm{E}_\mu[\mu^2] - 2(\lambda_0\mu_0 + \textstyle\sum_{n=1}^{N} x_n)\,\mathrm{E}_\mu[\mu] + (\sum_{n=1}^{N} x_n^2) + \lambda_0\mu_0^2\right] \\
&= b_0 + \frac{1}{2}\left[(\lambda_0+N)(\lambda_N^{-1} + \mu_N^2) - 2(\lambda_0\mu_0 + \textstyle\sum_{n=1}^{N} x_n)\mu_N + (\sum_{n=1}^{N} x_n^2) + \lambda_0\mu_0^2\right]
\end{aligned}
$$

We can then write the parameter equations as follows, without any expectations:

$$\mu_N = \frac{\lambda_0 \mu_0 + N\bar{x}}{\lambda_0 + N}$$
$$\lambda_N = (\lambda_0 + N)\frac{a_N}{b_N}$$
$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$
$$a_N = a_0 + \frac{N+1}{2}$$
$$b_N = b_0 + \frac{1}{2}\left[(\lambda_0 + N)(\lambda_N^{-1} + \mu_N^2) - 2(\lambda_0\mu_0 + \textstyle\sum_{n=1}^{N} x_n)\mu_N + (\sum_{n=1}^{N} x_n^2) + \lambda_0\mu_0^2\right]$$

Note that there are circular dependencies among the formulas for μ_N , λ_N and b_N . This naturally suggests an EM-like algorithm:

1. Compute $\sum_{n=1}^{N} x_n$ and $\sum_{n=1}^{N} x_n^2$. Use these values to compute μ_N and a_N.
2. Initialize λ_N to some arbitrary value.
3. Use the current value of λ_N, along with the known values of the other parameters, to compute b_N .
4. Use the current value of b_N, along with the known values of the other parameters, to compute λ_N .
5. Repeat the last two steps until convergence (i.e. until neither value has changed more than some small amount).

We then have values for the hyperparameters of the approximating distributions of the posterior parameters, which we can use to compute any properties we want of the posterior — e.g. its mean and variance, a 95% highest-density region (the smallest interval that includes 95% of the total probability), etc.

It can be shown that this algorithm is guaranteed to converge to a local maximum, and since both posterior distributions are in the exponential family, this local maximum will be a global maximum.

Note also that the posterior distributions have the same form as the corresponding prior distributions. We did *not* assume this; the only assumption we made was that the distributions factorize, and the form of the distributions followed naturally. It turns out (see below) that the fact that the posterior distributions have the same form as the prior distributions is not a coincidence, but a general result whenever the prior distributions are members of the exponential family, which is the case for most of the standard distributions.

34.4 Further discussion

34.4.1 Step-by-step recipe

The above example shows the method by which the variational-Bayesian approximation to a posterior probability density in a given Bayesian network is derived:

1. Describe the network with a graphical model, identifying the observed variables (data) $\mathbf{X}$ and unobserved variables (parameters $\boldsymbol{\Theta}$ and latent variables $\mathbf{Z}$) and their conditional probability distributions. Variational Bayes will then construct an approximation to the posterior probability $p(\mathbf{Z}, \boldsymbol{\Theta} \mid \mathbf{X})$. The approximation has the basic property that it is a factorized distribution, i.e. a product of two or more independent distributions over disjoint subsets of the unobserved variables.
2. Partition the unobserved variables into two or more subsets, over which the independent factors will be derived. There is no universal procedure for doing this; creating too many subsets yields a poor approximation, while creating too few makes the entire variational Bayes procedure intractable. Typically, the first split is to separate the parameters and latent variables; often, this is enough by itself to produce a tractable result. Assume that the partitions are called $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$.
3. For a given partition $\mathbf{Z}_j$, write down the formula for the best approximating distribution $q_j^*(\mathbf{Z}_j \mid \mathbf{X})$ using the basic equation $\ln q_j^*(\mathbf{Z}_j \mid \mathbf{X}) = \mathrm{E}_{i \neq j}[\ln p(\mathbf{Z}, \mathbf{X})] + \text{constant}$.

4. Fill in the formula for the joint probability distribution using the graphical model. Any component conditional distributions that don't involve any of the variables in $\mathbf{Z}_j$ can be ignored; they will be folded into the constant term.

5. Simplify the formula and apply the expectation operator, following the above example. Ideally, this should simplify into expectations of basic functions of variables not in $\mathbf{Z}_j$ (e.g. first or second raw moments, expectation of a logarithm, etc.). In order for the variational Bayes procedure to work well, these expectations should generally be expressible analytically as functions of the parameters and/or hyperparameters of the distributions of these variables. In all cases, these expectation terms are constants with respect to the variables in the current partition.

6. The functional form of the formula with respect to the variables in the current partition indicates the type of distribution. In particular, exponentiating the formula generates the probability density function (PDF) of the distribution (or at least, something proportional to it, with unknown normalization constant). In order for the overall method to be tractable, it should be possible to recognize the functional form as belonging to a known distribution. Significant mathematical manipulation may be required to convert the formula into a form that matches the PDF of a known distribution. When this can be done, the normalization constant can be reinstated by definition, and equations for the parameters of the known distribution can be derived by extracting the appropriate parts of the formula.

7. When all expectations can be replaced analytically with functions of variables not in the current partition, and the PDF put into a form that allows identification with a known distribution, the result is a set of equations expressing the values of the optimum parameters as functions of the parameters of variables in other partitions.

8. When this procedure can be applied to all partitions, the result is a set of mutually linked equations specifying the optimum values of all parameters.

9. An expectation maximization (EM) type procedure is then applied, picking an initial value for each parameter and the iterating through a series of steps, where at each step we cycle through the equations, updating each parameter in turn. This is guaranteed to converge.

34.4.2 Most important points

Due to all of the mathematical manipulations involved, it is easy to lose track of the big picture. The important things are:

1. The idea of variational Bayes is to construct an analytical approximation to the posterior probability of the set of unobserved variables (parameters and latent variables), given the data. This means that the form of the solution is similar to other Bayesian inference methods, such as Gibbs sampling — i.e. a distribution that seeks to describe everything that is known about the variables. As in other Bayesian methods — but unlike e.g. in expectation maximization (EM) or other maximum likelihood methods — both types of unobserved variables (i.e. parameters and latent variables) are treated the same, i.e. as random variables. Estimates for the variables can then be derived in the standard Bayesian ways, e.g. calculating the mean of the distribution to get a single point estimate or deriving a credible interval, highest density region, etc.

2. "Analytical approximation" means that a formula can be written down for the posterior distribution. The formula generally consists of a product of well-known probability distributions, each of which *factorizes* over a set of unobserved variables (i.e. it is conditionally independent of the other variables, given the observed data). This formula is not the true posterior distribution, but an approximation to it; in particular, it will generally agree fairly closely in the lowest moments of the unobserved variables, e.g. the mean and variance.

3. The result of all of the mathematical manipulations is (1) the identity of the probability distributions making up the factors, and (2) mutually dependent formulas for the parameters of these distributions. The actual values of these parameters are computed numerically, through an alternating iterative procedure much like EM.

34.4.3 Compared with expectation maximization (EM)

Variational Bayes (VB) is often compared with expectation maximization (EM). The actual numerical procedure is quite similar, in that both are alternating iterative procedures that successively converge on optimum parameter

values. The initial steps to derive the respective procedures are also vaguely similar, both starting out with formulas for probability densities and both involving significant amounts of mathematical manipulations.

However, there are a number of differences. Most important is *what* is being computed.

- EM computes point estimates of posterior distribution of those random variables that can be categorized as "parameters", but estimates of the actual posterior distributions of the latent variables (at least in "soft EM", and often only when the latent variables are discrete). The point estimates computed are the modes of these parameters; no other information is available.
- VB, on the other hand, computes estimates of the actual posterior distribution of all variables, both parameters and latent variables. When point estimates need to be derived, generally the mean is used rather than the mode, as is normal in Bayesian inference. Concomitant with this, it should be noted that the parameters computed in VB do *not* have the same significance as those in EM. EM computes optimum values of the parameters of the Bayes network itself. VB computes optimum values of the parameters of the distributions used to approximate the parameters and latent variables of the Bayes network. For example, a typical Gaussian mixture model will have parameters for the mean and variance of each of the mixture components. EM would directly estimate optimum values for the parameters. VB, however, would first fit a distribution to these parameters — typically in the form of a prior distribution, e.g. a normal-scaled inverse gamma distribution — and would then compute values for the parameters of this prior distribution, i.e. essentially hyperparameters. In this case, VB would compute optimum estimates of the four parameters of the normal-scaled inverse gamma distribution that describes the joint distribution of the mean and variance of the component.

34.5 A more complex example

Imagine a Bayesian Gaussian mixture model described as follows:[4]

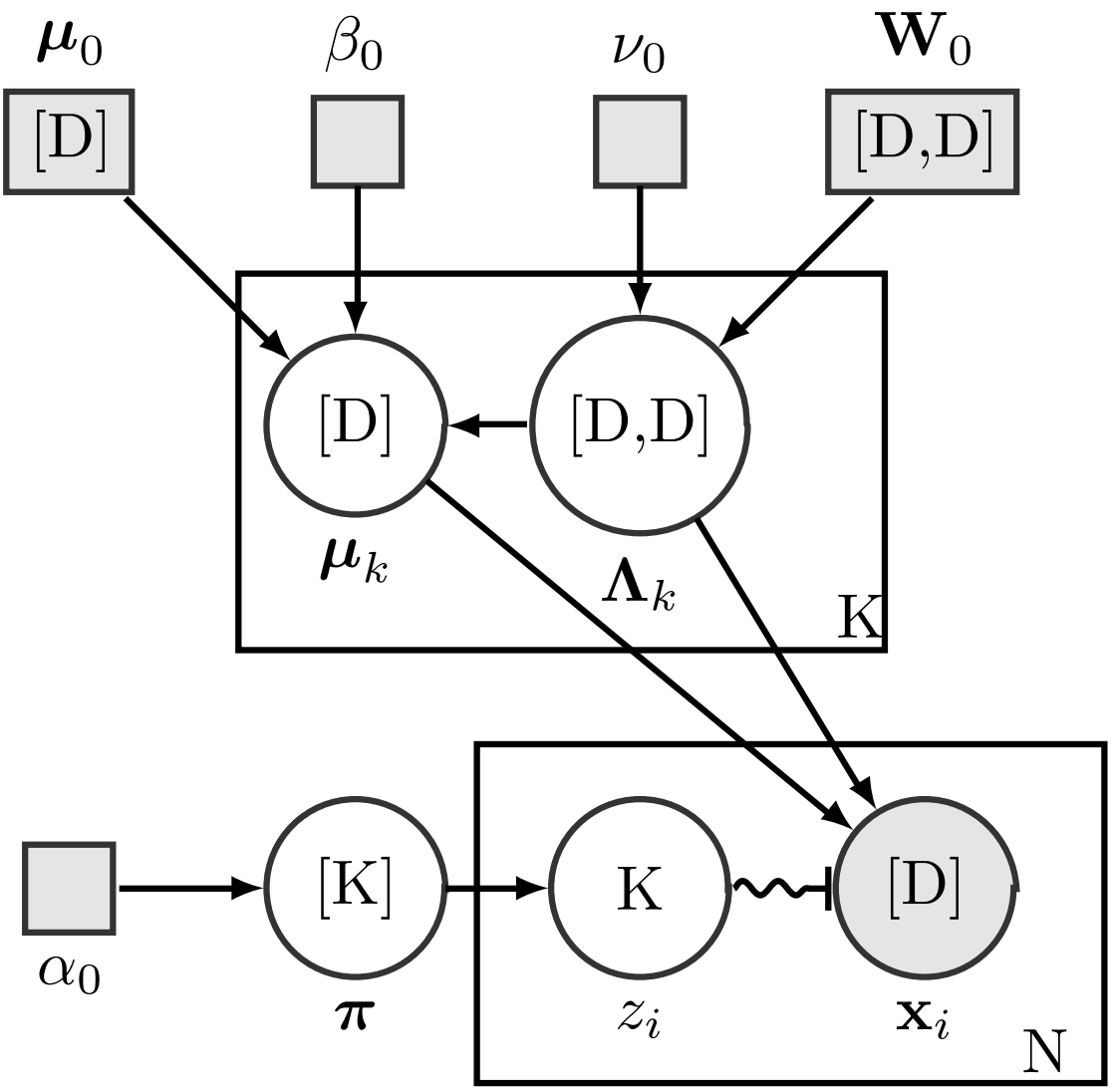
$$
\begin{aligned}
\pi &\sim \text{SymDir}(K, \alpha_0) \\
\Lambda_{i=1\dots K} &\sim \mathcal{W}(\mathbf{W}_0, \nu_0) \\
\mu_{i=1\dots K} &\sim \mathcal{N}(\mu_0, (\beta_0 \Lambda_i)^{-1}) \\
\mathbf{z}[i = 1 \dots N] &\sim \text{Mult}(1, \pi) \\
\mathbf{x}_{i=1\dots N} &\sim \mathcal{N}(\mu_{z_i}, \Lambda_{z_i}{}^{-1}) \\
K &= \text{components mixing of number} \\
N &= \text{points data of number}
\end{aligned}
$$

Note:

- SymDir() is the symmetric Dirichlet distribution of dimension K , with the hyperparameter for each component set to α_0 . The Dirichlet distribution is the conjugate prior of the categorical distribution or multinomial distribution.
- $\mathcal{W}()$ is the Wishart distribution, which is the conjugate prior of the precision matrix (inverse covariance matrix) for a multivariate Gaussian distribution.
- Mult() is a multinomial distribution over a single observation (equivalent to a categorical distribution). The state space is a "one-of-K" representation, i.e. a K -dimensional vector in which one of the elements is 1 (specifying the identity of the observation) and all other elements are 0.
- $\mathcal{N}()$ is the Gaussian distribution, in this case specifically the multivariate Gaussian distribution.

The interpretation of the above variables is as follows:

- $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is the set of N data points, each of which is a K -dimensional vector distributed according to a multivariate Gaussian distribution.



Bayesian Gaussian mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size K*; [D,D] means a matrix of size* D×D*; K alone means a categorical variable with* K *outcomes. The squiggly line coming from* z *ending in a crossbar indicates a* switch *— the value of this variable selects, for the other incoming variables, which value to use out of the size-*K *array of possible values.*

- $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ is a set of latent variables, one per data point, specifying which mixture component the corresponding data point belongs to, using a "one-of-K" vector representation with components z_{nk} for $k = 1 \ldots K$, as described above.

- π is the mixing proportions for the K mixture components.

- $\mu_{i=1\ldots K}$ and $\square_{i=1\ldots K}$ specify the parameters (mean and precision) associated with each mixture component.

The joint probability of all variables can be rewritten as

$$p(\mathbf{X}, \mathbf{Z}, \pi, \mu, \square) = p(\mathbf{X} \mid \mathbf{Z}, \mu, \square)p(\mathbf{Z} \mid \pi)p(\pi)p(\mu \mid \square)p(\square)$$

where the individual factors are

$$p(\mathbf{X} \mid \mathbf{Z}, \mu, \boxed{?}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n \mid \mu_k, \boxed{?}_k^{-1})^{z_{nk}}$$

$$p(\mathbf{Z} \mid \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}}$$

$$p(\pi) = \frac{\Gamma(K\alpha_0)}{\Gamma(\alpha_0)^K} \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1}$$

$$p(\mu \mid \boxed{?}) = \prod_{k=1}^{K} \mathcal{N}(\mu_k \mid \mu_0, (\beta_0 \boxed{?}_k)^{-1})$$

$$p(\boxed{?}) = \prod_{k=1}^{K} \mathcal{W}(\boxed{?}_k \mid \mathbf{W}_0, \nu_0)$$

where

$$\mathcal{N}(\mathbf{x} \mid \mu, \boxed{?}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boxed{?}|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mu)^{\mathrm{T}} \boxed{?}^{-1} (\mathbf{x} - \mu) \right\}$$

$$\mathcal{W}(\boxed{?} \mid \mathbf{W}, \nu) = B(\mathbf{W}, \nu) |\boxed{?}|^{(\nu - D - 1)/2} \exp\left(-\frac{1}{2} \operatorname{Tr}(\mathbf{W}^{-1} \boxed{?}) \right)$$

$$B(\mathbf{W}, \nu) = |\mathbf{W}|^{-\nu/2} \left\{ 2^{\nu D/2} \pi^{D(D-1)/4} \prod_{i=1}^{D} \Gamma\left(\frac{\nu + 1 - i}{2} \right) \right\}^{-1}$$

$$D = \text{point data each of dimensionality}$$

Assume that $q(\mathbf{Z}, \pi, \mu, \boxed{?}) = q(\mathbf{Z}) q(\pi, \mu, \boxed{?})$.

Then

$$\begin{aligned} \ln q^*(\mathbf{Z}) &= \mathrm{E}_{\pi, \mu, \boxed{?}}[\ln p(\mathbf{X}, \mathbf{Z}, \pi, \mu, \boxed{?})] + \text{constant} \\ &= \mathrm{E}_{\pi}[\ln p(\mathbf{Z} \mid \pi)] + \mathrm{E}_{\mu, \boxed{?}}[\ln p(\mathbf{X} \mid \mathbf{Z}, \mu, \boxed{?})] + \text{constant} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \rho_{nk} + \text{constant} \end{aligned}$$

where we have defined

$$\ln \rho_{nk} = \mathrm{E}[\ln \pi_k] + \frac{1}{2} \mathrm{E}[\ln |\boxed{?}_k|] - \frac{D}{2} \ln(2\pi) - \frac{1}{2} \mathrm{E}_{\mu_k, \boxed{?}_k}[(\mathbf{x}_n - \mu_k)^{\mathrm{T}} \boxed{?}_k (\mathbf{x}_n - \mu_k)]$$

Exponentiating both sides of the formula for $\ln q^*(\mathbf{Z})$ yields

$$q^*(\mathbf{Z}) \propto \prod_{n=1}^{N} \prod_{k=1}^{K} \rho_{nk}^{z_{nk}}$$

Requiring that this be normalized ends up requiring that the ρ_{nk} sum to 1 over all values of k , yielding

$$q^*(\mathbf{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} r_{nk}^{z_{nk}}$$

where

$$r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K} \rho_{nj}}$$

In other words, $q^*(\mathbf{Z})$ is a product of single-observation multinomial distributions, and factors over each individual $\mathbf{z}_n$, which is distributed as a single-observation multinomial distribution with parameters r_{nk} for $k = 1 \dots K$.

Furthermore, we note that

$$\mathrm{E}[z_{nk}] = r_{nk}$$

which is a standard result for categorical distributions.

Now, considering the factor $q(\pi, \mu, ?)$, note that it automatically factors into $q(\pi) \prod_{k=1}^{K} q(\mu_k, ?_k)$ due to the structure of the graphical model defining our Gaussian mixture model, which is specified above.

Then,

$$\begin{aligned} \ln q^*(\pi) &= \ln p(\pi) + \mathrm{E}_{\mathbf{Z}}[\ln p(\mathbf{Z} \mid \pi)] + \text{constant} \\ &= (\alpha_0 - 1) \sum_{k=1}^{K} \ln \pi_k + \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \ln \pi_k + \text{constant} \end{aligned}$$

Taking the exponential of both sides, we recognize $q^*(\pi)$ as a Dirichlet distribution

$$q^*(\pi) \sim \mathrm{Dir}(\alpha)$$

where

$$\alpha_k = \alpha_0 + N_k$$

where

$$N_k = \sum_{n=1}^{N} r_{nk}$$

Finally

$$\ln q^*(\mu_k, ?_k) = \ln p(\mu_k, ?_k) + \sum_{n=1}^{N} \mathrm{E}[z_{nk}] \ln \mathcal{N}(\mathbf{x}_n \mid \mu_k, ?_k^{-1}) + \text{constant}$$

Grouping and reading off terms involving μ_k and $?_k$, the result is a Gaussian-Wishart distribution given by

$$q^*(\mu_k, ?_k) = \mathcal{N}(\mu_k \mid \mathbf{m}_k, (\beta_k ?_k)^{-1}) \mathcal{W}(?_k \mid \mathbf{W}_k, \nu_k)$$

given the definitions

$$
\begin{aligned}
\beta_k &= \beta_0 + N_k \\
\mathbf{m}_k &= \frac{1}{\beta_k}(\beta_0 \mu_0 + N_k \bar{\mathbf{x}}_k) \\
\mathbf{W}_k^{-1} &= \mathbf{W}_0^{-1} + N_k \mathbf{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\mathbf{x}}_k - \mu_0)(\bar{\mathbf{x}}_k - \mu_0)^{\mathrm{T}} \\
\nu_k &= \nu_0 + N_k \\
N_k &= \sum_{n=1}^{N} r_{nk} \\
\bar{\mathbf{x}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \mathbf{x}_n \\
\mathbf{S}_k &= \frac{1}{N_k} \sum_{n=1}^{N} (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^{\mathrm{T}}
\end{aligned}
$$

Finally, notice that these functions require the values of r_{nk} , which make use of ρ_{nk} , which is defined in turn based on $\mathrm{E}[\ln \pi_k]$, $\mathrm{E}[\ln |\boldsymbol{\Lambda}_k|]$, and $\mathrm{E}_{\mu_k, \boldsymbol{\Lambda}_k}[(\mathbf{x}_n - \mu_k)^{\mathrm{T}} \boldsymbol{\Lambda}_k (\mathbf{x}_n - \mu_k)]$. Now that we have determined the distributions over which these expectations are taken, we can derive formulas for them:

$$
\begin{aligned}
\mathrm{E}_{\mu_k, \boldsymbol{\Lambda}_k}[(\mathbf{x}_n - \mu_k)^{\mathrm{T}} \boldsymbol{\Lambda}_k (\mathbf{x}_n - \mu_k)] &= D\beta_k^{-1} + \nu_k (\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}} \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k) \\
\ln \tilde{\Lambda}_k \equiv \mathrm{E}[\ln |\boldsymbol{\Lambda}_k|] &= \sum_{i=1}^{D} \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D \ln 2 + \ln |\mathbf{W}_k| \\
\ln \tilde{\pi}_k \equiv \mathrm{E}\left[\ln |\pi_k|\right] &= \psi(\alpha_k) - \psi\left(\sum_{i=1}^{K} \alpha_i\right)
\end{aligned}
$$

These results lead to

$$
r_{nk} \propto \tilde{\pi}_k \tilde{\Lambda}_k^{1/2} \exp\left\{-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}} \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k)\right\}
$$

These can be converted from proportional to absolute values by normalizing over k so that the corresponding values sum to 1.

Note that:

1. The update equations for the parameters β_k , $\mathbf{m}_k$, $\mathbf{W}_k$ and ν_k of the variables μ_k and $\boldsymbol{\Lambda}_k$ depend on the statistics N_k , $\bar{\mathbf{x}}_k$, and $\mathbf{S}_k$, and these statistics in turn depend on r_{nk} .
2. The update equations for the parameters $\alpha_{1\ldots K}$ of the variable π depend on the statistic N_k , which depends in turn on r_{nk} .
3. The update equation for r_{nk} has a direct circular dependence on β_k , $\mathbf{m}_k$, $\mathbf{W}_k$ and ν_k as well as an indirect circular dependence on $\mathbf{W}_k$, ν_k and $\alpha_{1\ldots K}$ through $\tilde{\pi}_k$ and $\tilde{\Lambda}_k$.

This suggests an iterative procedure that alternates between two steps:

1. An E-step that computes the value of r_{nk} using the current values of all the other parameters.
2. An M-step that uses the new value of r_{nk} to compute new values of all the other parameters.

Note that these steps correspond closely with the standard EM algorithm to derive a maximum likelihood or maximum a posteriori (MAP) solution for the parameters of a Gaussian mixture model. The responsibilities r_{nk} in the E step correspond closely to the posterior probabilities of the latent variables given the data, i.e. $p(\mathbf{Z} \mid \mathbf{X})$; the computation of the statistics N_k , $\bar{\mathbf{x}}_k$, and $\mathbf{S}_k$ corresponds closely to the computation of corresponding "soft-count" statistics over the data; and the use of those statistics to compute new values of the parameters corresponds closely to the use of soft counts to compute new parameter values in normal EM over a Gaussian mixture model.

34.6 Exponential-family distributions

Note that in the previous example, once the distribution over unobserved variables was assumed to factorize into distributions over the "parameters" and distributions over the "latent data", the derived "best" distribution for each variable was in the same family as the corresponding prior distribution over the variable. This is a general result that holds true for all prior distributions derived from the exponential family.

34.7 See also

- Variational message passing: a modular algorithm for variational Bayesian inference.
- Expectation-maximization algorithm: a related approach which corresponds to a special case of variational Bayesian inference.
- Generalized filtering: a variational filtering scheme for nonlinear state space models.
- Calculus of variations: the field of mathematical analysis that deals with maximizing or minimizing functionals.

34.8 Notes

[1] Boyd, Stephen P.; Vandenberghe, Lieven (2004). *Convex Optimization* (PDF). Cambridge University Press. ISBN 978-0-521-83378-3. Retrieved October 15, 2011.

[2] Christopher Bishop, *Pattern Recognition and Machine Learning*, 2006

[3] Based on Chapter 10 of *Pattern Recognition and Machine Learning* by Christopher M. Bishop

[4] Based on Chapter 10 of *Pattern Recognition and Machine Learning* by Christopher M. Bishop

34.9 References

- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer. ISBN 0-387-31073-8.

34.10 External links

- Variational-Bayes Repository A repository of papers, software, and links related to the use of variational methods for approximate Bayesian learning
- The on-line textbook: Information Theory, Inference, and Learning Algorithms, by David J.C. MacKay provides an introduction to variational methods (p. 422).
- Variational Algorithms for Approximate Bayesian Inference, by M. J. Beal includes comparisons of EM to Variational Bayesian EM and derivations of several models including Variational Bayesian HMMs.
- A Tutorial on Variational Bayes. Fox, C. and Roberts, S. 2012. Artificial Intelligence Review, doi:10.1007/s10462-011-9236-8.
- High-Level Explanation of Variational Inference by Jason Eisner may be worth reading before a more mathematically detailed treatment.

34.11 Text and image sources, contributors, and licenses

34.11.1 Text

- **Bayes' theorem** *Source:* http://en.wikipedia.org/wiki/Bayes'{}%20theorem?oldid=661572996 *Contributors:* AxelBoldt, Zundark, The Anome, Taw, Ap, Fcueto, Miguel~enwiki, DavidLevinson, Heron, Stevertigo, Michael Hardy, Eliasen, Chinju, Dcljr, TakuyaMurata, Karada, Alfio, Snoyes, Randywombat, Cherkash, EdH, Lancevortex, Hike395, Bjcairns, Charles Matthews, Timwi, Dcoetzee, Jitse Niesen, Colin Marquardt, Doradus, Jogloran, Schutz, Henrygb, Academic Challenger, Bkell, Josh Griffith, Spellbinder, Paul Murray, Wile E. Heresiarch, Snobot, Giftlite, WiseWoman, Lupin, Dratman, Rick Block, Duncharris, Maroux, Geni, Pcarbonn, Ja malcolm, MarkSweep, DragonflySixtyseven, Maximaximax, Sam Hocevar, Tietew, Urhixidur, Vsb, Sonett72, Guppyfinsoup, Guanabot, Antaeus Feldspar, El C, Chalst, Kwamikagami, Marco Polo, 3mta3, Landroni, Terrycojones, Monado, InBalance, Oleg Alexandrov, Splintax, Kzollman, Duncan.france, Facetious, Btyner, SqueakBox, Stoni, Graham87, BD2412, MauriceJFox3, Rjwilmsi, Billjefferys, FlaBot, Mathbot, Narxysus, GJ, Jameshfisher, Gurch, Quuxplusone, Fresheneesz, Rrenner, Sodin, Marlow4, Butros, CiaPan, Jmorgan, Chobot, YurikBot, Wavelength, Petiatil, Ogai, Makana, Pseudomonas, Ritchy, Terra Green, NawlinWiki, ENeville, Yserarau, Dysmorodrepanis~enwiki, Calumny, Trovatore, Aldux, Conradl, Shotgunlee, Ajarmst, Smaines, Xabian40409, Arthur Rubin, Drallim, Zvika, Cmglee, Marquez~enwiki, AeroIllini, SmackBot, RDBury, Tomyumgoong, BeteNoir, Khfan93, Rtc, InverseHypercube, Melchoir, McGeddon, Verne Equinox, Dandin1, SmartGuy Old, Mcld, JCSantos, ChuckHG, Jprg1966, Emufarmers, MartinPoulter, Karen Huyser, Metacomet, Nbarth, DHN-bot~enwiki, Zven, Dragice, Tsca.bot, Benet Allen, Cybercobra, RolandR, G716, Zadignose, Jóna Þórunn, Theme97, SashatoBot, Lambiam, Srays, Tim bates, Nijdam, Weather Man, BillFlis, Noah Salzman, Akitstika, Kripkenstein, Stephen B Streater, Happy-melon, Ceran, Gustavh, Sharp11, Wafulz, Shorespirit, Amosfolarin, Toms2866, Requestion, Neelix, Gregbard, Mattbuck, Kupirijo, Pianoroy, Mathew5000, Abtract, Lindsay658, Thijs!bot, Wikid77, Haakondahl, Martin Hogbin, Andyjsmith, Helgus, NorwegianBlue, Daniel il, J.e, Nobar, Urdutext, Marvoir, AntiVandalBot, Jvstone, Hannes Eder, Salgueiro~enwiki, Deflective, MER-C, Primarscources, Plynn9, Coffee2theorems, Hurmata, P64, Sisson, Ranger2006, Artegis, Baccyak4H, Trioculite, Gimboid, Sullivan.t.j, User A1, Jfmarchini, Jodi.a.schneider, MartinBot, Svyatoslav, Gill110951, DJ1AM, Gombang, SteveMacIntyre, NewEnglandYankee, Policron, Dessources, Jordanolsommer, Idioma-bot, VolkovBot, JustinHagstrom, Maghnus, TXiKiBoT, Daniel347x, JeffKo427, Weakishspeller, Blocter, Jmath666, Karinhunter, Andrewaskew, Lamro, Lova Falk, Forwardmeasure, Enkyo2, SieBot, JeffJor, Bryangeneolson, SE16, Cwkmail, Smsarmad, TimothyFreeman, Lightmouse, Htmort, AlanUS, Melcombe, Lazarus1907, Francvs, Llywelyn2000, ClueBot, Rumping, Justin W Smith, Vespertineflora, Freefall322, Manishearth, Lgstarn, Apelbaum, Iner22, Llamapez, Sun Creator, PhySusie, Brianbjparker, Hercule, Subash.chandran007, Qwfp, MairAW, XLinkBot, Pichpich, Gerhardvalentin, Ost316, C. A. Russell, WikHead, Paulpeeling, Addbot, Tcncv, MrOllie, Download, Tide rolls, Zorrobot, John.St, Krtschil, Luckas-bot, Yobot, Themfromspace, Charleswallingford, Bikramac, Newportm, Amirobot, Jean.julius, AnomieBOT, VanishedUser sdu9aya9fasdsopa, Materialscientist, ArthurBot, Shadowjams, Kaslanidi, Samwb123, Hushpuckena, FrescoBot, Olexa Riznyk, Sanpitch, X7q, Lunae, Citation bot 1, Kiefer.Wolfowitz, Riitoken, Fentlehan, Swapnil.iims, MedicineMan555, Siddhartha Ghai, Docemc, Jauhienij, Gnathan87, Fergusq, Hateweaver, Duoduoduo, RjwilmsiBot, Marabiloso, Wfunction, WikitanvirBot, Lemeza, Mo ainm, Davegroulx, Savh, ZéroBot, JA(000)Davidson, Qniemiec, Makecat, Donner60, Chewings72, Ebehn, Voomoo, Thunderfish24, Wcherowi, Wikidilworth, KoleTang, DramaticTheory, Thepigdog, Sirovsky, Penguin929, Helpful Pixie Bot, Curb Chain, BG19bot, Danachandler, Sailing to Byzantium, Solomon7968, Ramos1990, Limnoski, WikiAnoopRao, Glacialfox, Arpit.chauhan1, Dkrangi, New questions, Mmattb, Makecat-bot, Paul2520, Cramadur, Alexpiet, Fowlslegs, Exzession, Vieque, FSB1614, Mario Castelán Castro, Prof John Peacock, Jwmin15, SolidPhase and Anonymous: 464
- **Bayesian inference** *Source:* http://en.wikipedia.org/wiki/Bayesian%20inference?oldid=661273487 *Contributors:* The Anome, Fubar Obfusco, DavidSJ, Jinian, Edward, JohnOwens, Michael Hardy, Lexor, Karada, Ronz, Suisui, Den fjättrade ankan~enwiki, LouI, EdH, Jonik, Hike395, Novum, Timwi, WhisperToMe, Selket, SEWilco, Jose Ramos, Insightaction, Banno, Robbot, Kiwibird, Benwing, Meduz, Henrygb, AceMyth, Wile E. Heresiarch, Ancheta Wis, Giftlite, DavidCary, Dratman, Leonard G., JimD, Wmahan, Pcarbonn, MarkSweep, L353a1, FelineAvenger, APH, Sam Hocevar, Perey, Discospinster, Rich Farmbrough, Bender235, ZeroOne, Donsimon~enwiki, MisterSheik, El C, Edward Z. Yang, DimaDorfman, Cje~enwiki, John Vandenberg, LeonardoGregianin, Jung dalglish, Hooperbloob, Landroni, Arcenciel, Nurban, Avenue, Cburnett, Jheald, Facopad, Sjara, Oleg Alexandrov, Roylee, Joriki, Mindmatrix, BlaiseFEgan, Btyner, Magister Mathematicae, Tlroche, Rjwilmsi, Ravik, Jeffmcneill, Billjefferys, FlaBot, Brendan642, Kri, Chobot, Reetep, Gdrbot, Adoniscik, Wavelength, Pacaro, Gaius Cornelius, ENeville, Dysmorodrepanis~enwiki, Snek01, BenBildstein, Modify, Mastercampbell, Nothlit, NielsenGW, Mebden, Bo Jacoby, Cmglee, Boggie~enwiki, Harthacnut, SmackBot, Mmernex, Rtc, Mcld, Cunya, Gilliam, DoctorW, Nbarth, G716, Jbergquist, Turms, Bejnar, Gh02t, Wyxel, Josephsieh, JeonghunNoh, Thermochap, BoH, Basar, TheRegicider, Farzaneh, Lindsay658, Tdunning, Helgus, EdJohnston, Jvstone, Mack2, Lfstevens, Makohn, Stephanhartmannde, Comrade jo, Ph.eyes, Coffee2theorems, Ling.Nut, Charlesbaldo, DAGwyn, User A1, Tercer, STBot, Tobyr2, LittleHow, Policron, Jeffbadge, Bhepburn, Robcalver, James Kidd, VolkovBot, Thedjatclubrock, Maghnus, TXiKiBoT, Andrewaskew, GirasoleDE, SieBot, Doctorfree, Natta.d, Anchor Link Bot, Melcombe, Kvihill, Rfinchdavis, Smithpith, GeneCallahan, Krogstadt, Reovalis, Hussainshafqat, Charledl, ERosa, Qwfp, Tdslk, XLinkBot, Erreip, Addbot, K-MUS, Metagraph, LaaknorBot, Ozob, Legobot, Yobot, Gongshow, AnomieBOT, Citation bot, Shadak, Danielshin, VladimirReshetnikov, KingScot, JonDePlume, Thehelpfulbot, FrescoBot, Olexa Riznyk, WhatWasDone, Haeinous, JFK0502, Kiefer.Wolfowitz, 124Nick, Night Jaguar, Scientist2, Trappist the monk, Gnathan87, Philocentric, Jonkerz, Jowa fan, EmausBot, Blumehua, Montgolfière, Moswento, McPastry, Bagrowjp, SporkBot, Willy.pregliasco, Floombottle, Epdeloso, ClueBot NG, Mathstat, Bayes Puppy, Jj1236, Albertttt, Thepigdog, Helpful Pixie Bot, Michael.d.larkin, Jeraphine Gryphon, Whyking thc, Intervallic, CitationCleanerBot, DaleSpam, Kaseton, Simonsm21, Danielribeirosilva, ChrisGualtieri, Alialamifard, Yongli Han, 90b56587, MittensR, Mark viking, Boomx09, Waynechew87, Hamoudafg, Promise her a definition, Abacenis, Engheta, Avehtari, SolidPhase, LadyLeodia and Anonymous: 225
- **Bayesian network** *Source:* http://en.wikipedia.org/wiki/Bayesian%20network?oldid=660234761 *Contributors:* Ap, Fnielsen, SimonP, Fccoelho, ChangChienFu, Axon, Edward, Michael Hardy, Modster, Lexor, Ctwardy, Kku, Delirium, Eric119, Angela, Jordi Burguet Castell, Hike395, Kgajos, Hyacinth, Roachmeister, Benwing, Gidonb, Wile E. Heresiarch, Cutler, David Gerard, Giftlite, BenFrantzDale, Bfinn, Everyking, Neilc, Estel~enwiki, MarkSweep, Gene s, Urhixidur, Robin klein, Naku~enwiki, Rich Farmbrough, Guanabot, Mattlaabs~enwiki, Neko-chan, MisterSheik, Shadow demon, Skinkie~enwiki, 3mta3, Haham hanuka, Terrycojones, Samohyl Jan, Ceyockey, Oleg Alexandrov, Linas, Henrik, Mindmatrix, Bluegrass, BlaiseFEgan, Marudubshinki, Qwertyus, Kbdank71, Rjwilmsi, Salix alba, Oblivious, Mathbot, CarolGray, Fresheneesz, Chobot, Adoniscik, YurikBot, Wavelength, KamuiShirou, Piet Delport, Ogai, Buster79, Jpbowen, Balizarde, Kyle Cronan, DaveWF, Bruyninc~enwiki, SmackBot, Zanetu, UmassThrower, David Poole, Commander Keane bot, RDBrown, MalafayaBot, Jdthood, RichardHudson, Trucmuche~enwiki, Ohconfucius, Lambiam, Freewol, Arialblack, IronGargoyle, Dfass, Codesimian, Hu12, Kurtan~enwiki, Joostvandeputte~enwiki, CmdrObot, PuerExMachina, Phobius, Vizier, An-

drewHowse, Skittleys, The.snake, Paddles, Lfrittelli, Letranova, Headbomb, EdJohnston, AnAj, Tomixdf, Mack2, AndreasWittenstein, Ph.eyes, Magioladitis, Jarekt, Johnbibby, Trioculite, A3nm, Gregheth, Rajashar, Mpanahi, Iccaldwell, MoA)gnome, Andre.holzner, Absurdburger, AgarwalSumeet, Causenet, LiveLearn, MarcoLittel, Daniel5Ko, Normanfenton, DavidCBryant, The enemies of god, VolkovBot, Jim.Callahan,Orlando, Camrn86, JohnBlackburne, AlnoktaBOT, Maghnus, Mtanti, Neversay.misher, Silya, Daniel347x, Andrewaskew, Nathanpowell, Kbkorb, Chilti, SieBot, IradBG, Garde, Ddxc, OKBot, CharlesGillingham, Melcombe, Kvihill, Sonarpulse, Leisink, Wsun, Tomas e, Mild Bill Hiccup, Angelferrer, ClickStudent, Thomas Kist, DragonBot, Tsourakakis, Brews ohare, Practical321, Qwfp, MairAW, Kolyma, Erreip, J Hazard, Addbot, DOI bot, Zariski83, AndrewHZ, Esolu, Lightbot, Gail, Zorrobot, Wireless friend, Luckas-bot, Yobot, Twexcom, Abhikshah, FoxLemmor, Ausaen, AnomieBOT, Rubinbot, Wrongfilter, Puhfyn, Citation bot, Zhlsh1113, Xqbot, J04n, Omnipaedista, RibotBOT, SassoBot, Mattg82, Lebd, Citation bot 1, Jonesey95, Meborsuk, Trappist the monk, PapaJupe, RjwilmsiBot, Ripchip Bot, EmausBot, John of Reading, WikitanvirBot, Hous21, Converge on truth, HiW-Bot, Cskudzu, Shaider iba, Erget2005, Akseli.palen, JGS2010, ClueBot NG, Probinf, Tatome, Arrandale, Frietjes, Habil zare, Rxnt, BG19bot, Danielkorzekwa, Papadim.G, Chafe66, Jwatson89, Giliev, Raspabill, APerson, Peripattikos, Foerstj, Mogism, Mark viking, Sieste, Fpetitjean~enwiki, Jodosma, Mikayé, Djsyclik, Paul2520, Anrnusna, Stamptrader, Paheld, Monkbot, Priyankp87, Abacenis, Ibsen 13, Mathewk1300, Laeciosantosbsb, Widianpear, Qqq06 and Anonymous: 194

- **Bayesian probability** *Source:* http://en.wikipedia.org/wiki/Bayesian%20probability?oldid=661658045 *Contributors:* AxelBoldt, Matthew Woodcraft, The Anome, Taw, Ap, Andre Engels, Cable Hills, Stevertigo, Edward, JohnOwens, Michael Hardy, Fred Bauder, MartinHarper, Dcljr, Arthur3030, Ahoerstemeier, Snoyes, Den fjättrade ankan~enwiki, Cyan, Poor Yorick, Jonik, Jm34harvey, AC, Pheon, Populus, MH~enwiki, Henrygb, KellyCoinGuy, Bkell, Wile E. Heresiarch, SpellBott, Unfree, Snobot, Giftlite, Jao, BenFrantzDale, Bfinn, Cathy Linton, Dratman, Duncharris, Macrakis, Pcarbonn, Quadell, L353a1, Gene s, Miorea, BlairZajac, Discospinster, Smyth, Xezbeth, Dbachmann, Nybbles, El C, Cretog8, O18, Sebastianlutz, Mcdonaldsguy, Jung dalglish, Flammifer, KarlHallowell, Larry V, Hooperbloob, ClementSeveillac, Zachlipton, Diego Moya, Moanzhu, John Quiggin, Monado, Avenue, Schaefer, Samohyl Jan, Jheald, Count Iblis, RainbowOfLight, Oleg Alexandrov, Brookie, INic, Tomlillis, GregorB, BlaiseFEgan, Marudubshinki, Graham87, Rjwilmsi, Jweiss11, Commander, MarSch, Ravik, Billjefferys, Wragge, Mathbot, RexNL, Valor, Exelban, Fresheneesz, Chobot, Jdannan, Beanyk, MacMog, Jules.LT, Che829, Daniel roy, Finell, Capitalist, Roydanroy, SmackBot, Tomyumgoong, Incnis Mrsi, Cazort, Aaadddaaammm, MartinPoulter, Snori, Nbarth, Jdthood, Ladislav Mecir, Trekphiler, Jahiegel, BenE, Cybercobra, Dwchin, G716, OverInsured, Harryboyles, Aroundthewayboy, Nijdam, RichardF, AdjustablePliers, Hetar, Emote, DavidGSDavies, Alfpooh~enwiki, Link2009, Panda17, N2e, ShelfSkewed, Requestion, Moreschi, Basar, Vizier, Gregbard, FilipeS, Logicus, Hebrides, Anthonyhcole, CNMIN, Daa89563, Helgus, Mr pand, EdJohnston, Jvstone, Tomixdf, Mack2, Storkk, Knotwork, Stephanhartmannde, Coffee2theorems, Avjoska, Ranger2006, Freddie McPhyll, Robin S, Topagae, Gwern, EtienneDolet, CommonsDelinker, Nono64, AgarwalSumeet, Rlsheehan, Aleksandr Grigoryev, Gill110951, Coppertwig, LittleHow, Normanfenton, Policron, Robcalver, JohnBlackburne, The Tetrast, Econterms, Lambyte, Andrewaskew, Enkyo2, PlanetStar, Doctorfree, GentDave, Mateat, Janopus, Ddxc, Melcombe, ClueBot, Reovalis, Viviannevilar, Pot, Charledl, ERosa, Qwfp, Tdslk, Erreip, Gerhardvalentin, Ost316, Imperial Star Destroyer, Hossdave, WMdeMuynck, NjardarBot, Lihaas, LemmeyBOT, Tassedethe, Legobot, PlankBot, Yobot, Mindbuilder, UNSEENUNHEARD, AnomieBOT, Materialscientist, Citation bot, Glenn Stokowski, Jockocampbell, Shadowjams, Borkert, X7q, Argumzio, Eurdem, DrilBot, Kiefer.Wolfowitz, Trappist the monk, Gnathan87, Richardcherron, Dinamik-bot, Arrowzf, EmausBot, JA(000)Davidson, Matteo.taiana, AManWithNoPlan, GreenMachine86, Peter M. Brown, Udaya.s.k, Iratheclimber, ClueBot NG, Bayes Puppy, BarrelProof, Habil zare, Helpful Pixie Bot, Scochran4, Dasonk, Lolapellicer, Lifeformnoho, Joydeep, Ellewarren, CeraBot, Acuppert, ChrisGualtieri, Dexbot, Lakun.patra, Monkbot, Sennsationalist and Anonymous: 189
- **Bayesian programming** *Source:* http://en.wikipedia.org/wiki/Bayesian%20programming?oldid=656888352 *Contributors:* Michael Hardy, Phoebe, Rjwilmsi, Mcld, RomanSpa, Jpmatthews, Thenub314, Dodger67, NicDumZ, Erreip, Rankersbo, Yobot, AnomieBOT, Citation bot, Ancechu, Brycehughes, BG19bot, MrBill3, SnippyHolloW, Monkbot, Walden2 and Anonymous: 6
- **Belief propagation** *Source:* http://en.wikipedia.org/wiki/Belief%20propagation?oldid=660087634 *Contributors:* Michael Hardy, CesarB, A5, Francs2000, SimonMayer, Giftlite, PeR, Andreas Kaufmann, Rich Farmbrough, Mecanismo, Emin63, YUL89YYZ, Xezbeth, 3mta3, Soultaco, Linas, Kbdank71, Rjwilmsi, ElKevbo, Mathbot, Vsion, Adoniscik, Piet Delport, Ikcotyck, CharlesHBennett, Lserni, SmackBot, Mcld, KYN, Gilliam, Averisk, Cantalamessa, Lage~enwiki, Synergy, Tomixdf, Dougher, AndreasWittenstein, Ph.eyes, David Eppstein, EssRon, Edratzer, Wikip rhyre, Maghnus, Jamelan, Quietbritishjim, Iknowyourider, Melcombe, Kvihill, Simon04, Zxcv2000, D.scain.farenzena, CohesionBot, Addbot, Yobot, Gdewilde, Yangtseyangtse, AnomieBOT, Citation bot, Hwymeers, Tokidokix, Citation bot 1, Trappist the monk, Jamesmcoughlan, Tapirtrust, EmausBot, John of Reading, Victolunik, EdoBot, FreePeter3000, Helpful Pixie Bot, BG19bot, Xuxing716, Tuonawa, Max Libbrecht, Intervallic, ChrisGualtieri, Jfwk, Primal dual, Snookerr, Monkbot, Chenglongjiang, Mpkuse and Anonymous: 66
- **Causal graph** *Source:* http://en.wikipedia.org/wiki/Causal%20graph?oldid=654037744 *Contributors:* Michael Hardy, Rjwilmsi, Rankersbo, Yobot, RevelationDirect, Mz7, Stamptrader, Isthishowiusername and Brrryant
- **Causal inference** *Source:* http://en.wikipedia.org/wiki/Causal%20inference?oldid=660888214 *Contributors:* Michael Hardy, Bender235, BD2412, Rjwilmsi, Magioladitis, AngelOfSadness, Tayste, Yobot, AnomieBOT, Sda030, Omnipaedista, BiObserver, ClueBot NG, Antiqueight, Potatoman54, Wuerzele, Monkbot, Lohit27, Sobbor and Anonymous: 5
- **Causal loop diagram** *Source:* http://en.wikipedia.org/wiki/Causal%20loop%20diagram?oldid=638838702 *Contributors:* Sbwoodside, Aetheling, RJFJR, Ligulem, Bgwhite, Conscious, Saittam, SmackBot, RDBury, Bluebot, Lambiam, CBM, Pgr94, AndrewHowse, Chasingsol, Sumadartson~enwiki, SoftwareDeveloper, Kruckenberg.1, Magioladitis, Rich257, D-rew, CommonsDelinker, Erkan Yilmaz, VolkovBot, Umar420e, Synthebot, Tamorlan, ImageRemovalBot, Traveler100, Ideal gas equation, Brews ohare, Addbot, MrOllie, Luckas-bot, Yobot, Trevithj, Rojogrande, Xkryj03, Crbnblu, Patrhoue, Addihockey10 (automated), SRaemiA, Weamari and Anonymous: 15
- **Causal Markov condition** *Source:* http://en.wikipedia.org/wiki/Causal%20Markov%20condition?oldid=609274502 *Contributors:* Hike395, Charles Matthews, Oleg Alexandrov, Kzollman, Rjwilmsi, MarSch, SmackBot, RDBrown, Munibert, CBM, Peterdjones, Melcombe, AnomieBOT, Erik9bot, Madbix and Anonymous: 3
- **Darwinian network** *Source:* http://en.wikipedia.org/wiki/Darwinian%20network?oldid=662228259 *Contributors:* McGeddon, Yobot, DemocraticLuntz, Jhonatanoliveira08 and Andre.eds
- **Dempster–Shafer theory** *Source:* http://en.wikipedia.org/wiki/Dempster%E2%80%93Shafer%20theory?oldid=647177269 *Contributors:* Michael Hardy, Evercat, Martha2000, Charles Matthews, Phoebe, Meduz, Wally, Chris-gore, Wile E. Heresiarch, Cutler, Alan Liefting, Giftlite, Jason Quinn, Gadfium, Urhixidur, Bender235, MisterSheik, Bantman, Oleg Alexandrov, Joriki, Linas, GregorB, BlaiseFEgan, Btyner, BD2412, Rjwilmsi, Lockley, John Deas, Incompetnce, Floriang, Adoniscik, Roboto de Ajvol, YurikBot, Anomalocaris, Yahya Abdal-Aziz, SmackBot, Dicklyon, Mathsci, CapitalR, Gregbard, Sadeghd, Helgus, BenJWoodcroft, Mfriesel, Belush,

Farquaadhnchmn, Curdeius, Davidmanheim, Jodi.a.schneider, Yaron K., LordAnubisBOT, M-le-mot-dit, BrianOfRugby, HyDeckar, Llorenzi, Part Deux, Phe-bot, Basharcse, Ducleotide, Kvihill, Josang, Mtroffaes, PixelBot, Addbot, Yobot, AnomieBOT, Citation bot, LilHelpa, K731, Omnipaedista, Hxd1011, LucienBOT, Citation bot 1, Eduard.semsch, Bazsola, NameIsRon, John of Reading, Suslindis-ambiguator, TwerpySugar, Jwollbold, EdoBot, ClueBot NG, Helpful Pixie Bot, Constant2011, Swatig20sg, Andyhowlett, Limit-theorem, Nigellwh, Nickjrose, Fabio Cuzzolin and Anonymous: 70

- **Dynamic Bayesian network** *Source:* http://en.wikipedia.org/wiki/Dynamic%20Bayesian%20network?oldid=659383100 *Contributors:* Zeno Gantner, Charles Matthews, Phil Boswell, Discospinster, MarkWahl, Kri, Melchoir, Morgaladh, Tomixdf, AndreasWittenstein, Mel-combe, Addbot, Arodichevski, ChristopherKingChemist, FrescoBot, Obankston, Binabik, Timflutre, BG19bot, Kfriston, Cerabot~enwiki, Ezdets, Conkywkpmacro, Guptapankaj1993 and Anonymous: 14
- **Expectation–maximization algorithm** *Source:* http://en.wikipedia.org/wiki/Expectation%E2%80%93maximization%20algorithm?oldid=662388702 *Contributors:* Rodrigob, Michael Hardy, Karada, Jrauser, BAxelrod, Hike395, Phil Boswell, Owenman, Robbyjo~enwiki, Benwing, Wile E. Heresiarch, Giftlite, Paisa, Vadmium, Onco p53, MarkSweep, Piotrus, Cataphract, Rama, MisterSheik, Alex Kosorukoff, O18, John Vandenberg, Jjmerelo~enwiki, 3mta3, Terrycojones, B k, Eric Kvaalen, Cburnett, Finfobia, Jheald, Forderud, Sergey Dmitriev, Igny, Bkkbrad, Bluemoose, Btyner, Qwertyus, Rjwilmsi, KYPark, Salix alba, Hild, Mathbot, Glopk, Kri, BradBeattie, YurikBot, Nils Grimsmo, Schmock, Régis B., Klutzy, Hakeem.gadi, Maechler, Ladypine, M.A.Dabbah, SmackBot, Mcld, Nbarth, Tekhnofiend, Iwa-terpolo, Bilgrau, Joeyo, Raptur, Derek farn, Jrouquie, Dicklyon, Alex Selby, Saviourmachine, Lavaka, Requestion, Cydebot, A876, Kallerdis, Libro0, Blaisorblade, Skittleys, Andyrew609, Talgalili, Tiedyeina, Rusmike, Headbomb, RobHar, LachlanA, AnAj, Zzpmarco, Dekimasu, JamesBWatson, Richard Bartholomew, Livingthingdan, Nkwatra, User A1, Edratzer, Osquar F, Numbo3, Salih, GongYi, Douglas-Lanman, Bigredbrain, Market Efficiency, Lamro, Daviddoria, Pine900, Tambal, Mosaliganti1.1, Melcombe, Sitush, Pratx, Alexbot, Hbeigi, Jakarr, Jwmarck, XLinkBot, Jamshidian, Addbot, Sunjuren, Fgnievinski, LaaknorBot, Aanthony1243, Peni, Luckas-bot, Yobot, LeonardoWeiss, AnomieBOT, Citation bot, TechBot, Chuanren, FrescoBot, Nageh, Erhanbas, Nocheenlatierra, Qiemem, Kiefer.Wolfowitz, Jmc200, Stpasha, Jszymon, GeypycGn, Trappist the monk, Thái Nhi, Ismailari, Dropsciencenotbombs, RjwilmsiBot, Slon02, EmausBot, Mikealandewar, John of Reading, Ш, Chire, Statna, ClueBot NG, Rezabot, Meea, Qwerty9967, Helpful Pixie Bot, Rxnt, Bibcode Bot, BG19bot, Chafe66, Whym, Lvilnis, BattyBot, Yasuo2, Illia Connell, JYBot, Blegat, Yogtad, Tentinator, Marko0991, Ginsuloft, Wccsnow, Ronniemaor, Monkbot, Nboley, Faror91, DilumA, Rider ranger47, Velvel2, Crimsonslide, Megadata tensor, Surbut and Anonymous: 149
- **Factor graph** *Source:* http://en.wikipedia.org/wiki/Factor%20graph?oldid=660086680 *Contributors:* Zundark, Bernhard Bauer, Michael Devore, Ablewisuk, 3mta3, Soultaco, Linas, Shae, Ikcotyck, Grafen, Mahdim, Reyk, SmackBot, OrphanBot, Tomixdf, R'n'B, Tom Minka, Melcombe, PerryTachett, Bender2k14, Nitinjain 4, Teleprinter Sleuth, Twri, Hwymeers, Miym, Citation bot 1, Aeckford, Helpful Pixie Bot, ChrisGualtieri and Anonymous: 25
- **Graphical model** *Source:* http://en.wikipedia.org/wiki/Graphical%20model?oldid=660082884 *Contributors:* Fnielsen, Michael Hardy, Den fjättrade ankan~enwiki, Hike395, Charles Matthews, Bernhard Bauer, Giftlite, MarkSweep, YUL89YYZ, Bender235, 3mta3, Ar-cenciel, Nvrmnd, Oleg Alexandrov, Bkkbrad, Shae, Kbdank71, Chobot, Renaud.richardet, SmackBot, Unyoyega, Bsilverthorn, Cyhatch, Dicklyon, Rgiuly, Thamelry, Perimosocordiae, Headbomb, Rkrish67, STBot, AgarwalSumeet, Tom Minka, GongYi, Melcombe, Kvi-hill, Tsourakakis, Sandit27, Qwfp, Jaelee11, Addbot, Lauyukpui, JimVC3, X7q, Citation bot 1, Dcirovic, Narges.sharif, ClueBot NG, Lawrence87, Helpful Pixie Bot, Rxnt, IkamusumeFan, ﻻﻫﻼ, SteenthIWbot, Monkbot, Velvel2 and Anonymous: 37
- **Influence diagram** *Source:* http://en.wikipedia.org/wiki/Influence%20diagram?oldid=656237704 *Contributors:* Michael Hardy, Ronz, Charles Matthews, A2Kafir, GJeffery, Rjwilmsi, Ligulem, Bgwhite, The Rambling Man, Welsh, Modify, RDBrown, DMS, Can't sleep, clown will eat me, Maxentrope, Natta.d, Kvihill, Niceguyedc, Kwhitten, SchreiberBike, Addbot, Behappyrightnow, Helpful Pixie Bot, GuySh, BattyBot and Anonymous: 35
- **Junction tree algorithm** *Source:* http://en.wikipedia.org/wiki/Junction%20tree%20algorithm?oldid=623019481 *Contributors:* Michael Hardy, Andreas Kaufmann, Rich Farmbrough, 3mta3, Alai, Piet Delport, SmackBot, AbsolutBildung, Paskin~enwiki, DOI bot, Citation bot 1, Max Libbrecht, Monkbot and Anonymous: 9
- **Latent variable** *Source:* http://en.wikipedia.org/wiki/Latent%20variable?oldid=635651588 *Contributors:* Michael Hardy, Karada, Samw, Charles Matthews, Topbanana, Phil Boswell, Giftlite, Pgan002, Nova77, MisterSheik, Spalding, Mdd, John Quiggin, Chrisjohnson, Rjwilmsi, Splintercellguy, Gaius Cornelius, Ssurendra~enwiki, Holon, Mkill, SmackBot, RDBrown, AdamSmithee, Royboycrashfan, Radagast83, Antonielly, Aleenf1, Luke Maurits, Laurens-af, CBM, WeggeBot, AndrewHowse, Olaf, Coffee2theorems, Nick Connolly, Melcombe, Hariva, Qwfp, Jasoegaard, Addbot, Yobot, NOrbeck, FrescoBot, Haeinous, BenzolBot, Mtanana, Enisrat, Timothyjlayton, Chire, Claraevallensis, Loraof, UnicornExplorer and Anonymous: 26
- **M-separation** *Source:* http://en.wikipedia.org/wiki/M-separation?oldid=288060125 *Contributors:* Michael Hardy, Kbdank71, Michael Slone, SmackBot, Bluebot, Gloy, Melcombe and Anonymous: 1
- **Markov blanket** *Source:* http://en.wikipedia.org/wiki/Markov%20blanket?oldid=612240560 *Contributors:* Rodrigob, Hike395, Charles Matthews, Hao2lian, Kgajos, Witbrock, Pgan002, MarkSweep, KingTT, Oleg Alexandrov, Linas, Kbdank71, RDBrown, OrphanBot, Skittleys, Tomixdf, Sunbeam44, Melcombe, Kvihill, Kl4m, Thomas Tvileren, Addbot, Josevellezcaldas, Arodichevski, ZéroBot, Laugh-sinthestocks, QualitycontrolUS, Rules 1324, Colbert Sesanker and Anonymous: 12
- **Markov logic network** *Source:* http://en.wikipedia.org/wiki/Markov%20logic%20network?oldid=637676919 *Contributors:* Bryan Derk-sen, Jogloran, 3mta3, Linas, Qwertyus, Guslacerda, RDBrown, Jxm, Dbtfz, CBM, Gregbard, RiedelCastro, Valeria.depaiva, Melcombe, Jan1nad, Aliotra and Anonymous: 22
- **Markov random field** *Source:* http://en.wikipedia.org/wiki/Markov%20random%20field?oldid=660093492 *Contributors:* Michael Hardy, Kku, Delirium, Poor Yorick, Witbrock, Giftlite, BenFrantzDale, Dratman, Eric Harris-Braun, Adam McMaster, Nparikh, Pavel Voze-nilek, 3mta3, Delius, Evil Monkey, Oleg Alexandrov, Soultaco, Linas, Shreevatsa, Qwertyus, Kbdank71, Fresheneesz, Chobot, Schmock, DaveWF, Saravask, Took, JonHarder, Salamurai, Dbtfz, Rgiuly, Nitchell, Headbomb, Xact, David Eppstein, Gacelo, Stimpak, Episte-menical, Jduchi, Prakash Nadkarni, Melcombe, Alexbot, Sameer0s, Addbot, LaaknorBot, Silverrocker, Yobot, AnomieBOT, Ciphers, Ziyuang, JElhrig, RandomDSdevel, WreckLoose, EmausBot, John of Reading, Njsg, ZéroBot, Vikram360, ClueBot NG, Lucianoedipo, Ditkekov, Helpful Pixie Bot, BG19bot, Eferrante, Fpetitjean~enwiki, PierreYvesLouis, Leduoba, Qiyang Zhao, Airwoz, Velvel2 and Anonymous: 48
- **Mixture distribution** *Source:* http://en.wikipedia.org/wiki/Mixture%20distribution?oldid=654156099 *Contributors:* Edward, Michael Hardy, Tomi, Benwing, Wile E. Heresiarch, Rich Farmbrough, MisterSheik, Eric Kvaalen, Arthena, MartinSpacek, Woohookitty, Shawn@garbett.org, Some guy, SmackBot, Nbarth, Radagast83, P199, AnRtist, Holopoj, DrMicro, Melcombe, Xiawi, Qwfp, J kabudian, SpBot, Yobot,

AnomieBOT, Eumolpo, Noelduku, Stpasha, Duoduoduo, RjwilmsiBot, Smason79, Ualbinoni, Bencwallace, Rferreira1204, Brownerthanu, Dhadfieldmenell and Anonymous: 16

- **Mixture model** *Source:* http://en.wikipedia.org/wiki/Mixture%20model?oldid=662054603 *Contributors:* Marj Tiefert, Michael Hardy, Kku, Tomi, Palfrey, Owen, Phil Boswell, Benwing, Giftlite, Seabhcan, BenFrantzDale, Pgan002, MisterSheik, O18, 3mta3, Eric Kvaalen, Oleg Alexandrov, MartinSpacek, Qwertyus, Rjwilmsi, Gareth McCaughan, Ligulem, Mathbot, Wavelength, Dake~enwiki, Gareth Jones, Chris Paulse, Mebden, SmackBot, Mm100100, Incnis Mrsi, Reedy, Eskimbot, Mcld, Bluebot, RDBrown, Nbarth, Iwaterpolo, Memming, Ianmacm, Jim.belk, Ben Moore, P199, Negrulio, MarkSandler, GeordieMcBain, JohnCD, ShelfSkewed, Kupirijo, Blaisorblade, Omicronpersei8, Smartcat, STBot, Andreas Mueller, Rumpuscat, Bahramisharif, Bulusun, RJASE1, Prakash Nadkarni, Legion fi, Melcombe, WikipedianMarlith, Tbmurphy, Mjaniec, Tayste, Addbot, DOI bot, Lov090, Illywhacker, Luckas-bot, Yobot, AnomieBOT, Twri, Gilo1969, Piloter, Miym, Wahoo-j, Stpasha, Duoduoduo, John of Reading, PaulTheOctopus, Markiewp, Aberdeen01, Helpful Pixie Bot, Jldurrieu, Gapar2, SciCompTeacher, Illia Connell, Nomar65, Pllull1, Vivek146 and Anonymous: 98
- **Moral graph** *Source:* http://en.wikipedia.org/wiki/Moral%20graph?oldid=562929343 *Contributors:* Michael Hardy, Andreas Kaufmann, Kbdank71, Maustrauser, SmackBot, Took, Isj-wikipedia, Byelf2007, AbsolutBildung, David Eppstein, Melcombe, Mattiamonga, Nexcis~enwiki, Citation bot, Arodichevski, Solomon7968 and Anonymous: 6
- **Naive Bayes classifier** *Source:* http://en.wikipedia.org/wiki/Naive%20Bayes%20classifier?oldid=653270025 *Contributors:* The Anome, Awaterl, Olivier, Michael Hardy, Bewildebeast, Zeno Gantner, Karada, Cyp, Den fjättrade ankan~enwiki, Hike395, Njoshi3, WhisperToMe, Toreau, Phil Boswell, RedWolf, Bkell, Wile E. Heresiarch, Giftlite, Akella, JimD, Bovlb, Macrakis, Neilc, Pgan002, MarkSweep, Gene s, Cagri, Anirvan, Trevor MacInnis, Thorwald, Splatty, Rich Farmbrough, Violetriga, Peterjoel, Smalljim, John Vandenberg, BlueNovember, Jason Davies, Caesura, Oleg Alexandrov, KKramer~enwiki, Btyner, Mandarax, Qwertyus, Rjwilmsi, Hgkamath, Johnnyw, Mathbot, Intgr, Sderose, YurikBot, Wavelength, PiAndWhippedCream, Cancan101, Bovineone, Arichnad, Karipuf, BOT-Superzerocool, Evryman, Johndburger, Mebden, XAVeRY, SmackBot, InverseHypercube, CommodiCast, Stimpy, ToddDeLuca, Gilliam, NickGarvey, Chris the speller, OrangeDog, PerVognsen, Can't sleep, clown will eat me, Memming, Mitar, Neshatian, Jklin, Ringger, WMod-NS, Tobym, Shorespirit, Mat1971, Dstanfor, Arauzo, Dantiston, Sytelus, Vera Rita~enwiki, Dkemper, Prolog, Ninjakannon, Jrennie, MSBOT, Coffee2theorems, Tremilux, Saurabh911, Robotman1974, David Eppstein, User A1, HebrewHammerTime, AllenDowney, Troos, AntiSpamBot, Newtman, STBotD, Mike V, RJASE1, VolkovBot, Maghnus, Anna Lincoln, Mbusux, Anders gorm, EverGreg, Fcady2007, Jojalozzo, Ddxc, Dchwalisz, AlanUS, Melcombe, Headlesspladder, Kotsiantis, Justin W Smith, Motmahp, Calimo, Dianegarey, Doobliebop, Alousybum, Sunsetsky, XLinkBot, Herlocker, Addbot, RPHv, Tsunanet, MrOllie, LaaknorBot, Yobot, TaBOT-zerem, Twexcom, AnomieBOT, Rubinbot, Smk65536, The Almighty Bob, Cantons-de-l'Est, گس‌ت‌ه‌م, FrescoBot, X7q, Proffviktor, Svourdroculed, Rickyphyllis, Jonesey95, Geoffrey I Webb, Classifier1234, Mwojnars, Wingiii, Helwr, EmausBot, Orphan Wiki, Tommy2010, GarouDan, Joseagonzalez, ClueBot NG, Hofmic, NilsHaldenwang, Luoli2000, BG19bot, MusikAnimal, Chafe66, Kavishwar.wagholikar, Geduowenyang, Hipponix, Fcbarbi, Librawill, ChrisGualtieri, XMU zhangy, Alialamifard, CorvetteC6RVip, Jamesmcmahon0, Tonytonov, Jmagasin, ScienceRandomness, Qingyuanxingsi, Micpalmia, Sofia Koutsouveli, Yuchsiao, Mvdyck, Don neufeld, YoniSmolin, Rapanshi, Ananth.sankar.1963, Hmerzic and Anonymous: 181
- **Polytree** *Source:* http://en.wikipedia.org/wiki/Polytree?oldid=641142721 *Contributors:* Michael Hardy, Charles Matthews, Doradus, Roachmeister, McKay, Giftlite, Andreas Kaufmann, Zaslav, Kundor, Oleg Alexandrov, LuisPedroCoelho, Jamarcus, SmackBot, Yeoil, Headbomb, MarshBot, David Eppstein, Squids and Chips, RatnimSnave, Kvihill, JP.Martin-Flatin, Addbot, Wireless friend, JakobVoss, Xqbot, RedBot, BG19bot, Dstjacques and Anonymous: 12
- **Probabilistic latent semantic analysis** *Source:* http://en.wikipedia.org/wiki/Probabilistic%20latent%20semantic%20analysis?oldid=648024583 *Contributors:* Fnielsen, Michael Hardy, Kku, Jitse Niesen, Vishvas vasuki, Rama, CheekyMonkey, Jonsafari, Arcenciel, Oleg Alexandrov, Bkkbrad, Wavelength, Ste1n, SmackBot, Mcld, Bluebot, CmdrObot, Keretapi~enwiki, Mbell, Sylenius, Transcendence, A3nm, Chiccodoro, Maghnus, Sunny house, Seo01, Melcombe, Addbot, DOI bot, Johnchallis, Pmj005, Yobot, Efsunselin, Mambomatx, Mehdiym, EduardoValle, Larry.europe, Alfaisanomega, Erniesgrove, DarafshBot, Hmainsbot1, Xin Alan Rong, Monkbot, MinorImprovements and Anonymous: 14
- **Recursive Bayesian estimation** *Source:* http://en.wikipedia.org/wiki/Recursive%20Bayesian%20estimation?oldid=660314063 *Contributors:* Rodrigob, Michael Hardy, Ronz, Den fjättrade ankan~enwiki, Mottzo, Giftlite, Qef, Nwerneck, Cmdrjameson, Forderud, Rjwilmsi, Mathbot, Nehalem, Adoniscik, Amit man, SmackBot, RDBrown, Nbarth, Radagast83, Lyst, Harej bot, Mikehead, Cydebot, Jiuguang Wang, Chris demonsen, Martarius, Yobot, Ego White Tray and Anonymous: 27
- **Structured prediction** *Source:* http://en.wikipedia.org/wiki/Structured%20prediction?oldid=643965303 *Contributors:* Edward, Kku, Nowozin, Qwertyus, Brendan642, Semifinalist, Geo g guy, Yobot, AnomieBOT, Venustas 12, Alfaisanomega, SwimmingFox, Weiping.thu, Papertoys, Mathewk1300 and Anonymous: 3
- **Variable elimination** *Source:* http://en.wikipedia.org/wiki/Variable%20elimination?oldid=596964892 *Contributors:* Michael Hardy, Bearcat, King of Hearts, Malcolma, Semifinalist, BG19bot, Jhonatanoliveira08 and Anonymous: 2
- **Variable-order Bayesian network** *Source:* http://en.wikipedia.org/wiki/Variable-order%20Bayesian%20network?oldid=618519613 *Contributors:* Michael Hardy, Skittleys, DavidCBryant, IradBG, Melcombe, SanderEvers, DOI bot, Citation bot, Arodichevski, Citation bot 1, Skyerise, Khazar2, Monkbot and Anonymous: 4
- **Variational Bayesian methods** *Source:* http://en.wikipedia.org/wiki/Variational%20Bayesian%20methods?oldid=658808571 *Contributors:* Edward, Michael Hardy, Benwing, Dmolla, Vadmium, Jheald, Oleg Alexandrov, Eclecticos, Qwertyus, Rjwilmsi, Pruneau, Brendan642, RussBot, ArmadniGeneral, Fram, Mebden, SmackBot, Mcld, Bluebot, Paulinus, Falk Lieder, WVhybrid, Lfstevens, Sanchom, Cnilep, Melcombe, Niceguyedc, LilHelpa, Nathanielvirgo, FrescoBot, Zfeinst, Timflutre, Helpful Pixie Bot, KLBot2, Kfriston, DarafshBot, Olivierkeke and Anonymous: 33

34.11.2 Images

- **File:Adoption_CLD.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/e/ea/Adoption_CLD.svg *License:* CC-BY-SA-3.0 *Contributors:*
- Adoption_CLD.gif *Original artist:* Adoption_CLD.gif: Original uploader was Apdevries at en.wikipedia
- **File:Ambox_important.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/b/b4/Ambox_important.svg *License:* Public domain *Contributors:* Own work, based off of Image:Ambox scales.svg *Original artist:* Dsmurat (talk · contribs)

- **File:Animation2.gif** *Source:* http://upload.wikimedia.org/wikipedia/commons/c/c0/Animation2.gif *License:* CC-BY-SA-3.0 *Contributors:* Own work *Original artist:* MG (talk · contribs)
- **File:Bayes'{}_Theorem_MMB_01.jpg** *Source:* http://upload.wikimedia.org/wikipedia/commons/1/18/Bayes%27_Theorem_MMB_01.jpg *License:* CC BY-SA 3.0 *Contributors:* Own work by *mattbuck*. *Original artist:* *mattbuck* (category)
- **File:Bayes_continuous_diagram.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/5/5b/Bayes_continuous_diagram.svg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Bayes_icon.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/e/ed/Bayes_icon.svg *License:* CC0 *Contributors:* iThe source code of this SVG is <a data-x-rel='nofollow' class='external text' href='http://validator.w3.org/check?uri=http%3A%2F%2Fcommons.wikimedia.org%2Fwiki%2FSpecial%3AFilepath%2FBayes_icon.svg,,&,,ss=1#source'>valid.
  Original artist: Mikhail Ryazanov
- **File:Bayes_theorem_drugs_example_tree.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/8/88/Bayes_theorem_drugs_example_tree.svg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Bayes_theorem_simple_example_tree.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/7/74/Bayes_theorem_simple_example_tree.svg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Bayes_theorem_tree_diagrams.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/6/61/Bayes_theorem_tree_diagrams.svg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Bayes_theorem_visualisation.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/b/bf/Bayes_theorem_visualisation.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Cmglee
- **File:Bayesian-categorical-mixture.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/c/c7/Bayesian-categorical-mixture.svg *License:* CC BY 3.0 *Contributors:* Created using LaTeX, TikZ *Original artist:* Benwing
- **File:Bayesian-gaussian-mixture-vb.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/2/2a/Bayesian-gaussian-mixture-vb.svg *License:* CC BY 3.0 *Contributors:* Created using LaTeX, TikZ *Original artist:* Benwing
- **File:Bayesian-gaussian-mixture.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/2/28/Bayesian-gaussian-mixture.svg *License:* CC BY 3.0 *Contributors:* Created using LaTeX, TikZ *Original artist:* Benwing
- **File:Bayesian_inference_archaeology_example.jpg** *Source:* http://upload.wikimedia.org/wikipedia/commons/6/6d/Bayesian_inference_archaeology_example.jpg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Bayesian_inference_event_space.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/a/ad/Bayesian_inference_event_space.svg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Bellcurve.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/d/df/Bellcurve.svg *License:* Copyrighted free use *Contributors:* ? *Original artist:* ?
- **File:CLD_links_ANI.gif** *Source:* http://upload.wikimedia.org/wikipedia/commons/d/d8/CLD_links_ANI.gif *License:* Public domain *Contributors:* Own work *Original artist:* Patrhoue
- **File:CLD_positive_ANI.gif** *Source:* http://upload.wikimedia.org/wikipedia/commons/4/43/CLD_positive_ANI.gif *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Patrhoue
- **File:Causal_Loop_Diagram_of_a_Model.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/f/f6/Causal_Loop_Diagram_of_a_Model.png *License:* Public domain *Contributors:* "Feedback". In: *U.S. Department of Energy's Introduction to System Dynamics.* *Original artist:* Robert A. Taylor, U.S. Department of Energy
- **File:College.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/a/ab/College.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Brrryant
- **File:College_notID.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/e/ea/College_notID.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Brrryant
- **File:College_notID_proj.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/0/02/College_notID_proj.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Brrryant
- **File:College_proj.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/9/97/College_proj.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Brrryant
- **File:Commons-logo.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* ? *Contributors:* ? *Original artist:* ?
- **File:Continuous_event_space_specification.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/1/17/Continuous_event_space_specification.svg *License:* CC0 *Contributors:* Own work *Original artist:* Gnathan87
- **File:Darwinian_Network_vs_Bayesian_Network,_2015.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/f/f4/Darwinian_Network_vs_Bayesian_Network%2C_2015.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Andre.eds
- **File:Dempster_in_Brest.JPG** *Source:* http://upload.wikimedia.org/wikipedia/commons/d/d4/Dempster_in_Brest.JPG *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Llorenzi
- **File:Diagram_of_a_Markov_blanket.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/e/eb/Diagram_of_a_Markov_blanket.svg *License:* CC0 *Contributors:* I made this diagram using Inkscape. It is patterned after the public domain image MarkovBlanket.png. *Original artist:* Laughsinthestocks

- **File:EM_Clustering_of_Old_Faithful_data.gif** *Source:* http://upload.wikimedia.org/wikipedia/commons/6/69/EM_Clustering_of_Old_Faithful_data.gif *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Chire
- **File:Ebits2c.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/2/2c/Ebits2c.png *License:* GFDL *Contributors:* Own work *Original artist:* P. Fraundorf
- **File:Em_old_faithful.gif** *Source:* http://upload.wikimedia.org/wikipedia/commons/a/a7/Em_old_faithful.gif *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* 3mta3 (talk) 16:55, 23 March 2009 (UTC)
- **File:Factorgraph.jpg** *Source:* http://upload.wikimedia.org/wikipedia/commons/3/32/Factorgraph.jpg *License:* Public domain *Contributors:* Own work *Original artist:* Hwymeers
- **File:Figure22.png** *Source:* http://upload.wikimedia.org/wikipedia/en/c/c2/Figure22.png *License:* PD *Contributors:*
  Self-made
  Original artist:
  R.D. Shachter and A. Detwarasiti
- **File:Fisher_iris_versicolor_sepalwidth.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/4/40/Fisher_iris_versicolor_sepalwidth.svg *License:* CC BY-SA 3.0 *Contributors:* en:Image:Fisher iris versicolor sepalwidth.png *Original artist:* en:User:Qwfp (original); Pbroks13 (talk) (redraw)
- **File:Folder_Hexagonal_Icon.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/4/48/Folder_Hexagonal_Icon.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Gaussian-mixture-example.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/7/71/Gaussian-mixture-example.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Smason79
- **File:Graph_model.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/3/39/Graph_model.svg *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* IkamusumeFan
- **File:HMM_Kalman_Filter_Derivation.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/8/81/HMM_Kalman_Filter_Derivation.svg *License:* Public domain *Contributors:* Own work *Original artist:* Qef
- **File:Internet_map_1024.jpg** *Source:* http://upload.wikimedia.org/wikipedia/commons/d/d2/Internet_map_1024.jpg *License:* CC BY 2.5 *Contributors:* Originally from the English Wikipedia; description page is/was here. *Original artist:* The Opte Project
- **File:LampFlowchart.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/9/91/LampFlowchart.svg *License:* CC-BY-SA-3.0 *Contributors:* vector version of Image:LampFlowchart.png *Original artist:* svg by Booyabazooka
- **File:Markov_random_field_example.png** *Source:* http://upload.wikimedia.org/wikipedia/en/f/f7/Markov_random_field_example.png *License:* CC-BY-SA-3.0 *Contributors:* ? *Original artist:* ?
- **File:MoralGraph-DAG1.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/3/3d/MoralGraph-DAG1.png *License:* Public domain *Contributors:* Slightly modified version of Image:Directed_acyclic_graph.png, “A directed acyclic graph, created by Derrick Coetzee in Illustrator and Photoshop.”, a file in the public domain.
  Original artist: AbsolutBildung at English Wikipedia
- **File:Moralized_graph1.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/8/80/Moralized_graph1.png *License:* Public domain *Contributors:* This is a lightly modified version of Image:Directed_acyclic_graph.png,
  Original artist: AbsolutBildung at English Wikipedia
- **File:Nonbayesian-categorical-mixture.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/8/8c/Nonbayesian-categorical-mixture.svg *License:* CC BY 3.0 *Contributors:* Created using LaTeX, TikZ *Original artist:* Benwing
- **File:Nonbayesian-gaussian-mixture.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/e/ed/Nonbayesian-gaussian-mixture.svg *License:* CC BY 3.0 *Contributors:* Created using LaTeX, TikZ *Original artist:* Benwing
- **File:Normal_distribution_pdf.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/1/1b/Normal_distribution_pdf.png *License:* CC-BY-SA-3.0 *Contributors:* ? *Original artist:* ?
- **File:Nuvola_apps_atlantik.png** *Source:* http://upload.wikimedia.org/wikipedia/commons/7/77/Nuvola_apps_atlantik.png *License:* LGPL *Contributors:* http://icon-king.com *Original artist:* David Vignoni / ICON KING
- **File:People_icon.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/3/37/People_icon.svg *License:* CC0 *Contributors:* OpenClipart *Original artist:* OpenClipart
- **File:Plsi_1.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/e/ec/Plsi_1.svg *License:* CC BY-SA 3.0 *Contributors:* http://en.wikipedia.org/wiki/File:Plsi.svg *Original artist:* Bkkbrad, EduardoValle
- **File:Polytree.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/6/61/Polytree.svg *License:* Public domain *Contributors:* Own work *Original artist:* luis@luispedro.org. Converted to SVG by Oleg Alexandrov 02:44, 3 August 2007 (UTC)
- **File:Portal-puzzle.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/f/fd/Portal-puzzle.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:SimpleBayesNet.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/0/0e/SimpleBayesNet.svg *License:* Public domain *Contributors:* Own work (Original text: *self-made*) *Original artist:* AnAj
- **File:SimpleBayesNetNodes.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/f/fd/SimpleBayesNetNodes.svg *License:* Public domain *Contributors:* http://en.wikipedia.org/wiki/File:SimpleBayesNet.svg *Original artist:* AnAj
- **File:Text_document_with_red_question_mark.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/a/a4/Text_document_with_red_question_mark.svg *License:* Public domain *Contributors:* Created by bdesham with Inkscape; based upon Text-x-generic.svg from the Tango project. *Original artist:* Benjamin D. Esham (bdesham)
- **File:Wiki_letter_w.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/6/6c/Wiki_letter_w.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Wiki_letter_w_cropped.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/1/1c/Wiki_letter_w_cropped.svg *License:* CC-BY-SA-3.0 *Contributors:*
- Wiki_letter_w.svg *Original artist:* Wiki_letter_w.svg: Jarkko Piiroinen

34.11.3 Content license