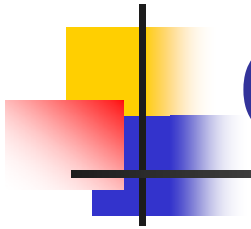
Research of Theories and Methods of Classification and Dimensionality Reduction

Jie Gui (桂杰)
中科院合肥智能机械研究所
2016.09.07



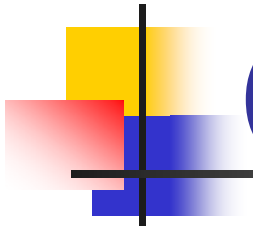
Outline

- Part I: Classification
- Part II: Dimensionality reduction
  - Feature selection
  - Subspace learning

Classifiers

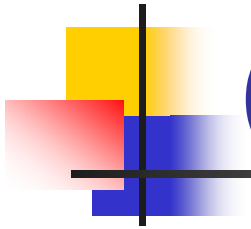
- **NN:** Nearest neighbor classifier
- **NC:** Nearest centriod classifier
- **NFL:** Nearest feature line classifier
- **NFP:** Nearest feature plane classifier
- **NFS:** Nearest feature space classifier
- **SVM:** Support vector machines
- **SRC:** Sparse representation-based classification
- **…**

Nearest neighbor classifier (NN)



- Given a new example, NN classifies the example as the class of the nearest training example to the observation.

Nearest centriod classifier (NC)



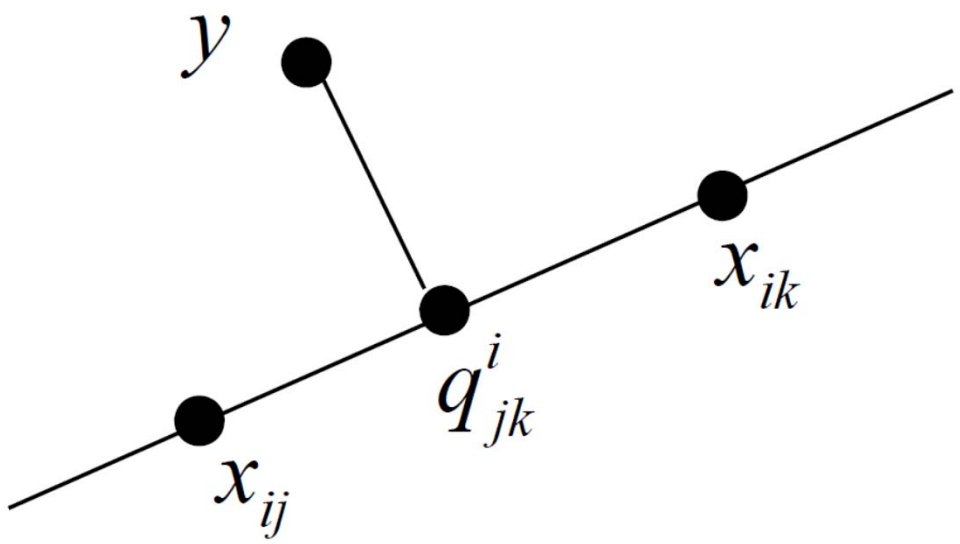
- Maybe NC is the simplest classifier.
- Two steps:
  - The mean vector of each class in the training set m_i is computed.
  - For each test example y, the distance to each centroid is then given by

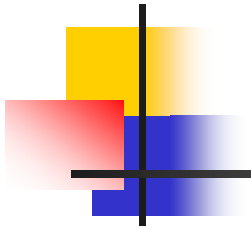
    $d_i(y) = \|y - m_i\|$.

    NC assigns y to class j if $d_j(y)$ is the minimum.

Nearest feature line classifier (NFL)

- Any two examples of the same class are generalized by the feature line (FL) passing through the two examples.



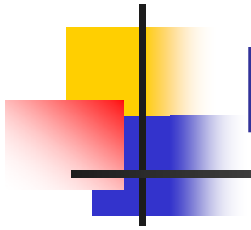


- The FL distance between y and $\overline{x_{ij}x_{ik}}$ is defined as $d\left(y, \overline{x_{ij}x_{ik}}\right) = d\left(y, q_{jk}^{i}\right)$.
- The decision function of class i is

$$d_i(y) = \min_{\substack{j,k=1,\cdots,n_i \\ j \neq k}} d\left(y, q_{jk}^{i}\right),\ i = 1,2,\cdots,c$$

- NFL assigns y to class m if $d_m(y)$ is the minimum.

S. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Netw.*, vol. 10, no. 2, pp. 439–443, Mar. 1999.

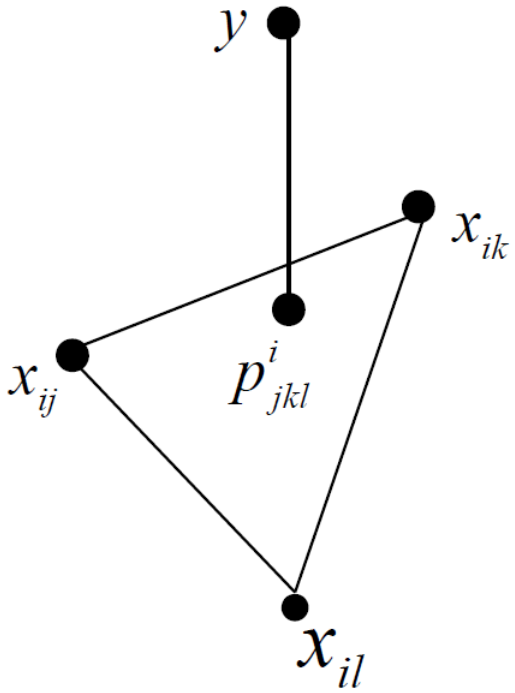


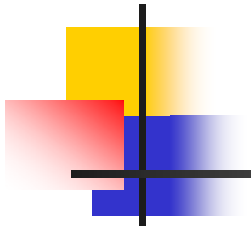
Motivation of NFL

- NFL can be seen as a variant of NN.
- NN can only use n_i examples while NFL can use $C_{n_i}^2$ lines for the ith class. For example, if $n_i = 5$, then $C_{n_i}^2$=10.
- Thus, NFL generalizes the representation capacity in case of only a small number of examples available per class.

Nearest feature plane classifier (NFP)

- Any three examples of the same class are generalized by the feature plane (FP) passing through the three examples





- The FP distance between y and $\overline{x_{ij}x_{ik}x_{il}}$ is defined as $d\left(y, \overline{x_{ij}x_{ik}x_{il}}\right) = d\left(y, p_{jkl}^{i}\right)$.
- The decision function of class i is

$$d_i(y) = \min_{\substack{j,k,l=1,\cdots,n_i \\ j \neq k \neq l}} d\left(y, p_{jkl}^{i}\right),\ i = 1,2,\cdots,c$$

- NFP assigns y to class m if $d_m(y)$ is the minimum.

Nearest feature space classifier (NFS)

- NFS assigns a test example y to class i if the distance from y to the subspace spanned by all examples X_i of class i:

$$d_i(y) = \min_{\beta_i} \|y - X_i\beta_i\|$$

is the minimum among all classes.

- Nearest neighbor classifier (NN)
- Nearest feature line classifier (NFL)
- Nearest feature plane classifier (NFP)
- Nearest feature space classifier (NFS)

NN (Point) -> NFL (Line) -> NFP (Plane) -> NFS (Space)

Representative vector machines (RVM)

- Although the motivations of the aforementioned classifiers vary, they can be unified in the form of "representative vector machines (RVM)" as follows:

representative vector to represent the ith class for *y*

$$k = \arg\ \min_i\ \|y - a_i\|$$

current test example

predicted class label for *y*

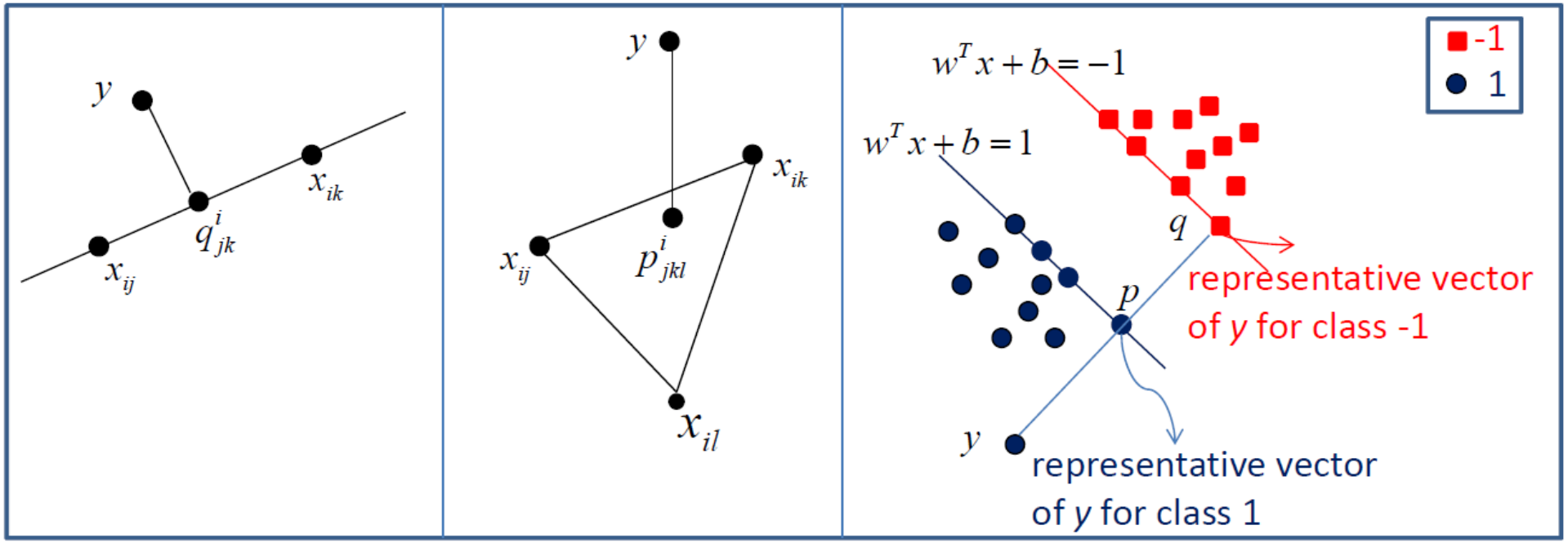


Figure 1. (a) Nearest feature line classifier. Generalizing two samples v_{ij} and v_{ik} by the feature line $\overline{v_{ij}v_{ik}}$. The sample y is projected onto the line as point q^i_{jk}. (b) Nearest feature plane classifier. Generalizing three samples v_{ij}, v_{ik} and v_{il} by the feature plane $\overline{v_{ij}v_{ik}v_{il}}$. The sample y is projected onto the plane as point p^i_{jkl}. (c) Support vector machines. The sample y is projected onto the plane $w^T x + b = 1$ as point p and projected onto the plane $w^T x + b = -1$ as point q.

SVM-> Large Margin Distribution Machine (LDM)

SVM

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \ \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.} \ \ y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \geq 1-\xi_i,$$
$$\xi_i \geq 0, \ i = 1, \ldots, m.$$

LDM

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \ \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \lambda_1\hat{\gamma} - \lambda_2\bar{\gamma} + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.} \ \ y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \geq 1-\xi_i,$$
$$\xi_i \geq 0, \ i = 1, \ldots, m.$$

margin mean

$$\bar{\gamma} = \frac{1}{m}\sum_{i=1}^{m} y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) = \frac{1}{m}(\boldsymbol{X}\boldsymbol{y})^\top \boldsymbol{w},$$

margin variance

$$\hat{\gamma} = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) - y_j \boldsymbol{w}^\top \phi(\boldsymbol{x}_j))^2$$

$$= \frac{2}{m^2}(m\boldsymbol{w}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{w}^\top \boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}^\top \boldsymbol{w}).$$

T. Zhang and Z.-H. Zhou. Large margin distribution machine. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'14), 2014, pp.313-322.

The representative vectors of classical classifiers

Methods	Representative vector for class i
NN	$\arg\ \min_{x_{ij}} \|y - x_{ij}\|$ s.t. $j = 1, \cdots, n_i$
NFL	$\arg\ \min_{q^i_{jk}} d\left(y, q^i_{jk}\right)$ s.t. $j, k = 1, \cdots, n_i; j \neq k$
NFP	$\arg\ \min_{p^i_{jkl}} d\left(y, p^i_{jkl}\right)$ s.t. $j, k, l = 1, \cdots, n_i; j \neq k \neq l$
NFS	$X_i \beta_i$
NC	the mean vector of class i : $m_i = \left(\sum_{j=1}^{n_i} x_{ij}\right) / n_i$
SRC	$X_i \hat{\alpha}_i$
SVMs	$y + \left(\left(i - b - w^T y\right) / w^T w\right) w$

Comparison of a number of classifiers

Classifier	Advantage	Disadvantage
NN	• Easy implementation • Learning is fast	Sensitive to noise
NFL/NFP	• Easy in software programming • Generalize the representation capacity in case of only a small number of examples available per class [23]–[25]	• Sensitive to noise • Costly and time-consuming when there are a large number of training examples per class
NFS	• Easy to implement • Quick at learning	Poor performance when classes are highly correlated [27]
NC	• Easy to be programmed • Computationally efficient	Poor performance when the number of the examples of some class is small
SRC	Impressive performance in some practical applications such as face recognition [27]	Computationally expensive [33]
SVMs	Good generalization ability on unseen data	Great computational complexity in some cases where large training set are involved

Discriminative vector machine

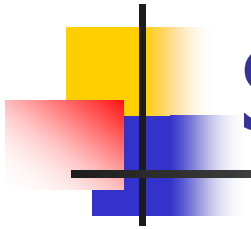
the robust M-estimator

A_k: k-nearest neighbors of y

$$\min_{\alpha_k} \sum_{i=1}^{d} \phi\left(\left(y - A_k\alpha_k\right)_i\right) + \beta\varphi\left(\alpha_k\right) + \gamma\sum_{p=1}^{k}\sum_{q=1}^{k} w_{pq}\left(\alpha_k^p - \alpha_k^q\right)^2$$

manifold regularization

the vector norm such as l_1-norm and l_2-norm



Statistical analysis for DVM

- First, we provide a generalization-error-like bound for the DVM algorithm by using the distribution-free inequalities obtained for k-local rules.
- Then, we prove that DVM algorithm is a PAC-learning algorithm for classification.

Generalization-error-like bound for DVM

Theorem 1: For DVM algorithm with $k \leq n-1$, we have

$$P\{|R - R_n^R| \geq \epsilon\} \leq 2\exp\left(-\frac{n\epsilon^2}{18}\right) + 6\exp\left(-\frac{n\epsilon^3}{108k(2+\gamma_d)}\right),$$

where γ_d is the maximum number of distinct points in R^d (a -d dimensional Euclidean space) which can share the same nearest neighbor and $\gamma_d \leq 3^d - 1$.

Main results

- **Theorem 2:** Under assumption 1, DVM algorithm is a PAC-learning algorithm for classification.
- **Lemma 1:** For DVM algorithm with $k \leq n-1$, we have

$$\begin{aligned} & E(R_n^D - R)^2 \\ & \leq \frac{1}{n} + 6 \max_i P\{c_n(x, Z^n) \neq c_n(x, Z^{n,i})\} \\ & \leq \frac{1}{n} + \frac{6k}{n}. \end{aligned}$$

- **Remark 1:** Deveroye and Wagner proved a faster convergence rate for $c = 2$.

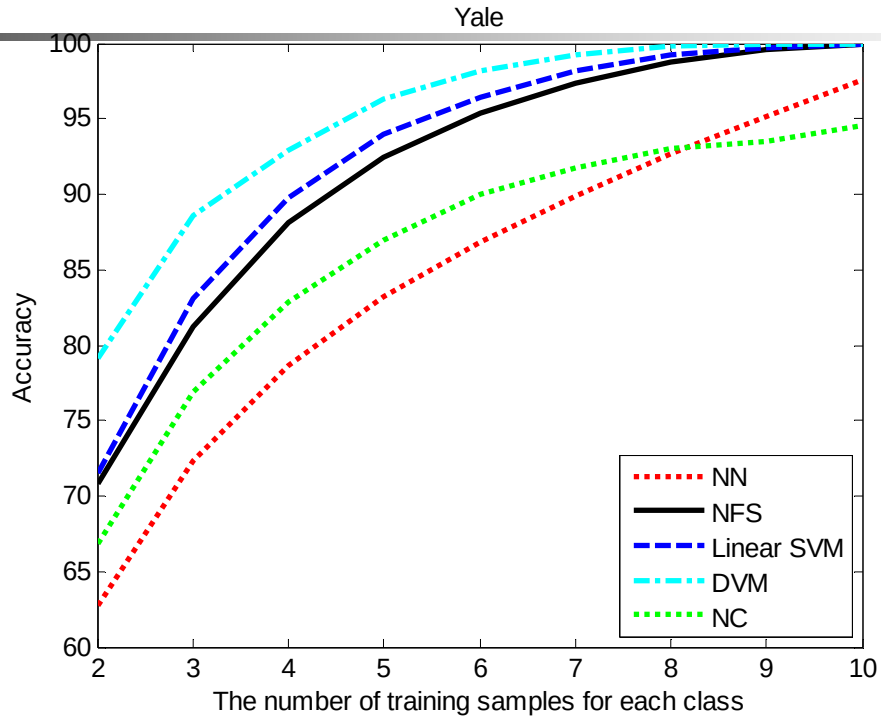
Experimental results using the Yale database

Method	2 Train	3 Train	4 Train	5 Train
NN	62.79±22.80	72.36±19.92	78.67±17.94	83.23±16.64
NC	66.79±20.83	76.89±17.34	82.91±14.55	86.98±11.82
NFL	70.67±19.36	80.81±15.40	86.93±12.98	91.66±10.30
NFP	-	81.54±15.26	88.38±11.47	93.10±8.44
NFS	70.79±19.09	81.25±15.31	88.10±11.56	92.41±8.96
SRC	78.79±15.45	87.27±11.54	91.92±8.66	94.57±6.59
Linear SVM	71.52±18.88	83.15±13.80	89.80±10.80	93.93±8.06
DVM	**79.15±14.63**	**88.57±10.99**	**92.87±8.83**	**96.33±6.15**

Method	6 Train	7 Train	8 Train	9 Train	10 Train
NN	86.87±15.44	89.94±14.10	92.65±12.55	95.15±10.62	97.58±8.04
NC	90.00±9.73	91.72±7.82	93.09±6.46	93.45±4.71	94.55±2.70
NFL	95.01±7.85	97.31±5.54	98.79±3.40	99.64±1.53	**100±0**
NFP	96.32±6.01	98.36±3.80	99.43±2.00	99.88±0.90	**100±0**
NFS	95.37±6.83	97.33±4.84	98.75±3.00	99.64±1.53	**100±0**
SRC	96.36±5.13	97.47±4.15	98.42±3.11	98.79±2.60	99.39±2.01
Linear SVM	96.41±6.01	98.22±4.07	99.19±2.42	99.76±1.26	**100±0**
DVM	**98.15±4.17**	**99.21±2.34**	**99.80±1.15**	**100± 0**	**100± 0**

Average recognition rates (percent) across all possible partitions on Yale

Experimental results using the Yale database



Average recognition rates (percent) as functions of the number of training examples per class on Yale

1. **DVM outperforms all other methods in all cases**
2. **NN method has the poorest performance except '9 Train' and '10 Train'.**

Experimental results on a large-scale database FRGC

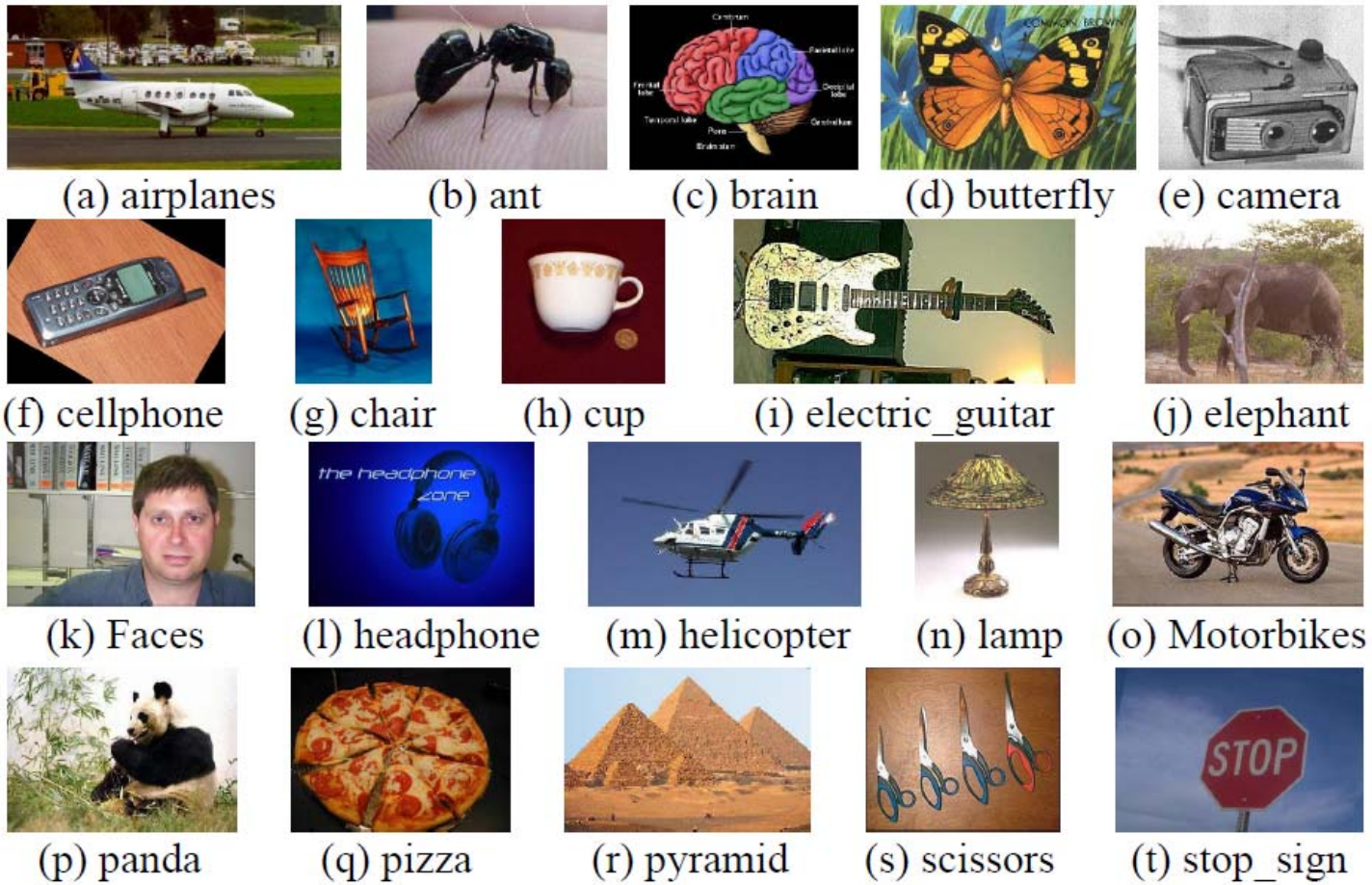
Method	NN	NC	NFL	NFP	NFS	SRC	SVM	DVM
OR	78.98± 1.08	55.51± 1.31	85.56± 1.08	88.31± 0.99	89.94± 0.92	**95.49± 0.72**	91.00± 0.83	88.41± 0.98
LBP	88.52± 1.12	78.33± 0.91	93.37± 1.01	93.38± 1.06	93.42± 0.99	**97.56± 0.46**	95.27± 0.91	97.28± 0.61
LDA	93.61± 0.76	93.74± 0.79	94.47± 0.83	94.56± 0.86	94.42± 0.84	93.90± 0.70	92.65± 0.86	**95.33± 0.64**
LBPLDA	96.00± 0.66	95.94± 0.54	95.99± 0.64	95.94± 0.69	95.30± 0.71	93.99± 0.72	95.91± 0.66	**96.16± 0.55**

Average recognition rate (percent) comparison on the FRGC dataset

1. DVM performs the best using LDA and LBPLDA

2. SRC performs the best using original representation (OR) and LBP.

Experimental results on the image dataset Caltech-101



Sample images of Caltech-101 (randomly selected 20 classes)

Comparison of accuracies on the Caltech-101

Method	15Train	30Train
LCC+SPM	65.43	73.44
Boureau et al.	-	77.1±0.7
Jia et al.	-	75.3±0.70
ScSPM +SVM	67.0±0.45	73.2±0.54
ScSPM +NN	49.95±0.92	56.53±0.96
ScSPM +NC	61.27±0.69	65.96±0.63
ScSPM +NFL	63.54±0.68	70.17±0.45
ScSPM +NFP	67.09±0.66	74.04±0.30
ScSPM +NFS	68.63±0.63	76.69±0.34
ScSPM +SRC	71.09±0.57	**78.28±0.52**
ScSPM +DVM	**71.69±0.49**	77.74±0.46

Comparison of average recognition rate (percent) on the Caltech-101 dataset

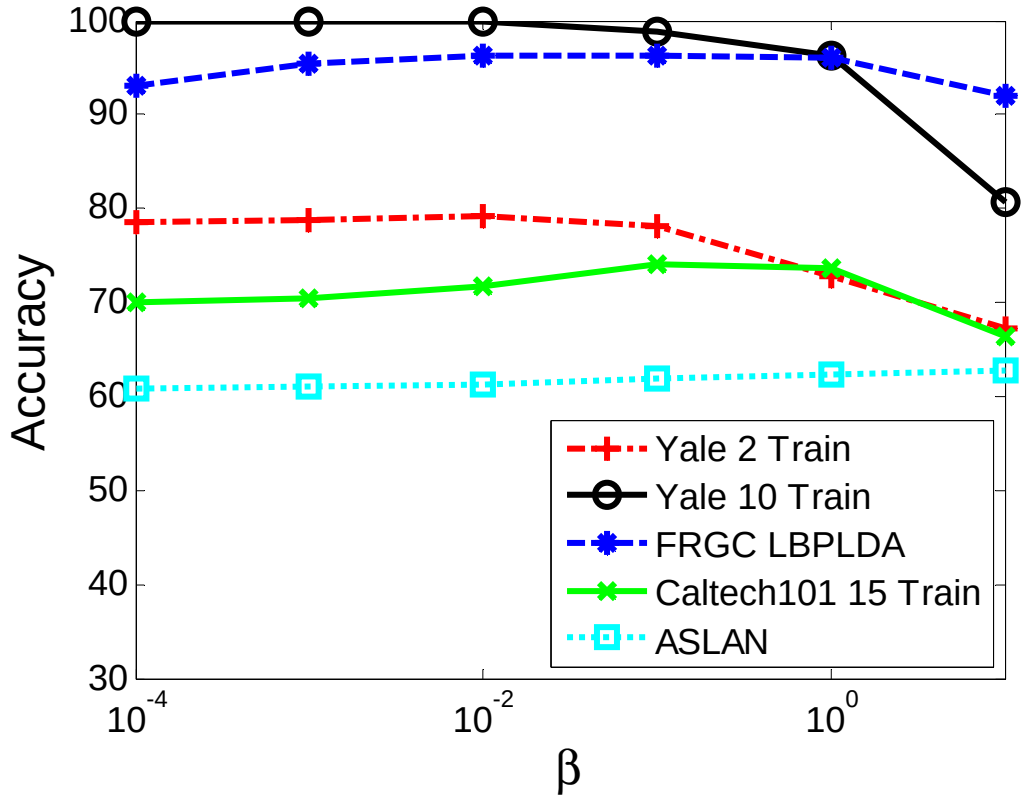
Experimental results on ASLAN

Methods	Performance
NN	53.95±0.76
NC	57.38±0.74
NFL	54.25±0.94
NFP	54.42±0.72
NFS	49.98±0.02
SRC	56.40±2.76
SVM	60.88±0.77
DVM	**61.37±0.68**

Comparison of average recognition rate (percent) on the ASLAN dataset

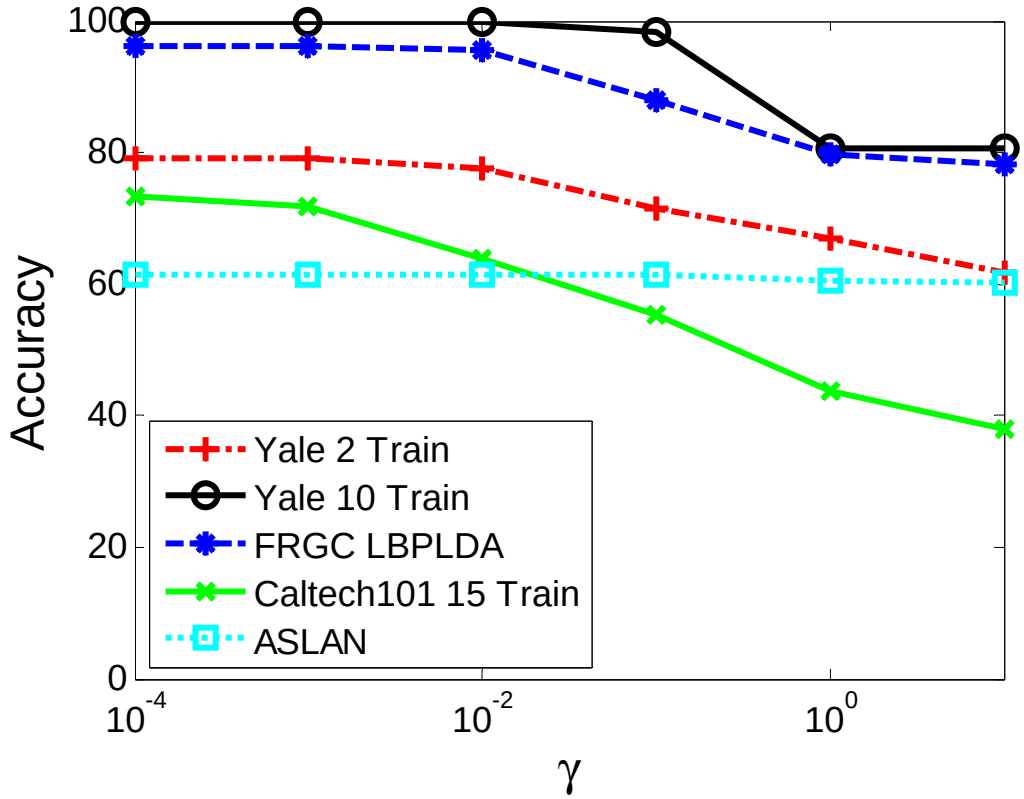
1. DVM outperforms all the other methods.

Parameter Selection for DVM



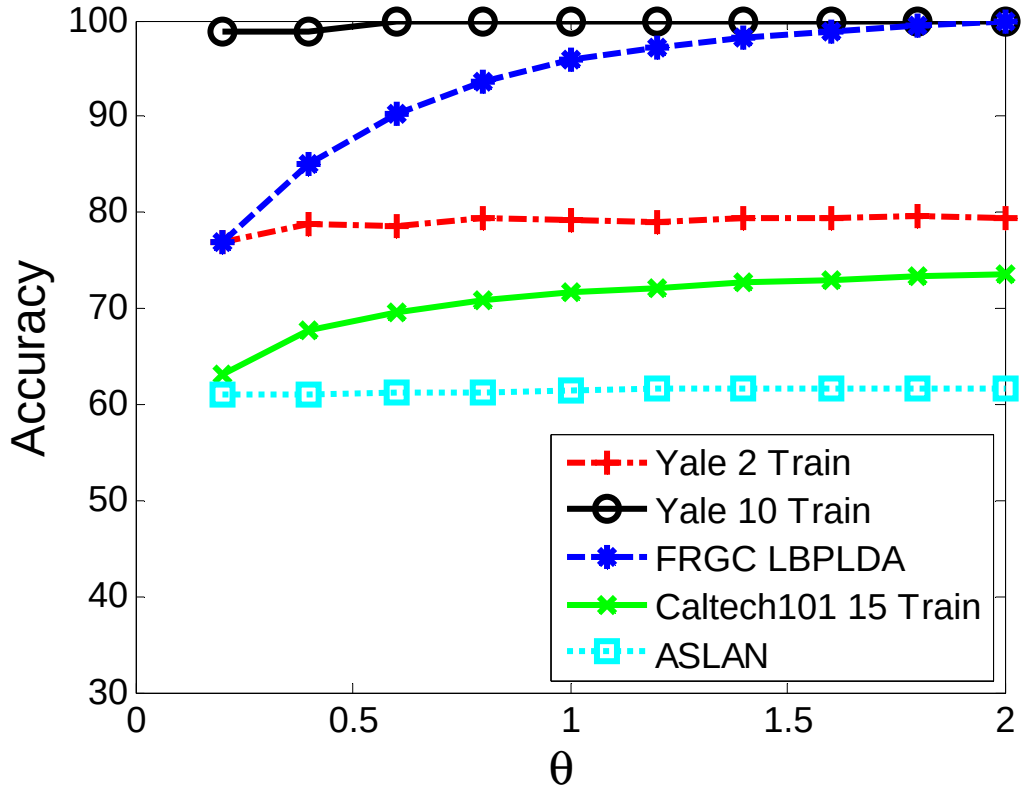
Accuracy versus β with γ and θ fixed on Yale, FRGC, Caltech 101 and ASLAN. The proposed DVM model is stable with varying β within $(10^{-4}, 10^{-1})$.

Parameter Selection for DVM

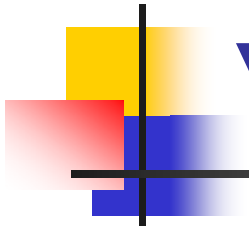


Accuracy versus γ with β and θ fixed on Yale, FRGC, Caltech 101 and ASLAN. The proposed DVM model is stable with varying γ within $(10^{-4}, 10^{-3})$.

Parameter Selection for DVM

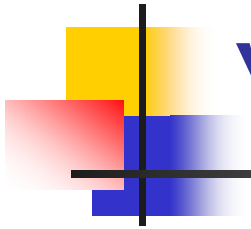


Accuracy versus θ with β and γ fixed on Yale, FRGC, Caltech 101 and ASLAN.



“Concerns” on our framework

- **C1:** Can this framework unify all classification algorithms?
  - **No.** Some classical classifiers, such as naive Bayes, cannot be unified in the manner of “representative vector machines”.



"Concerns" on our framework

- **C2:** Applications.
- **C3:** Note that the representative vector framework is a flexible framework. We can use l_2 distance, l_1 distance, etc. The selection of an appropriate similarity measure for different applications is still an unsolved problem.

Representative vector machines (RVM)

- This work is published in IEEE Transactions on Cybernetics:

Jie Gui, Tongliang Liu, Dacheng Tao, Zhenan Sun, Tieniu Tan, "Representative Vector Machines: A unified framework for classical classifiers", IEEE Transactions on Cybernetics, vol. 46, no. 8, pp. 1877-1888, 2016.

Representative vector machines (RVM)

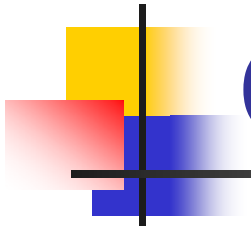
Although the motivations of the aforementioned classifiers vary, they can be unified in the form of "representative vector machines (RVM)" as follows:

representative vector to represent the ith class for *y*

$$k = \arg \min_i \|y - a_i\|$$

current test example

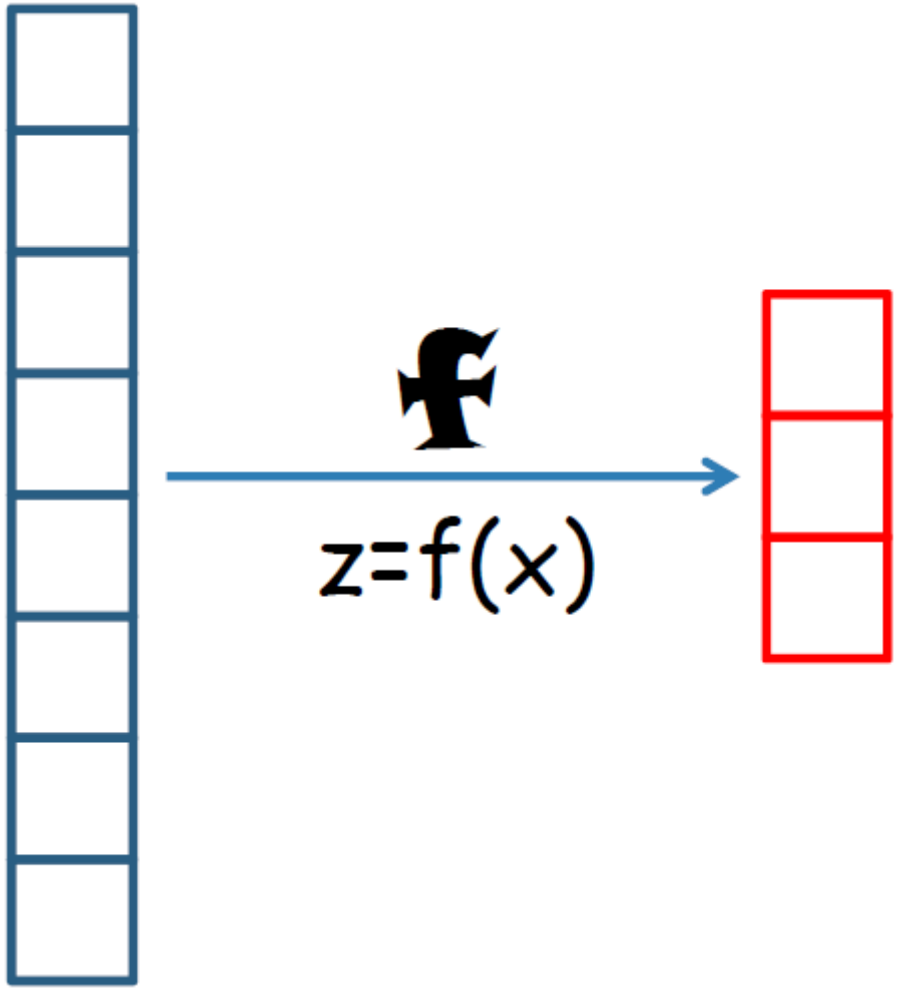
predicted class label for *y*



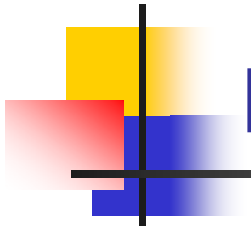
Outline

- Part I: Classification
- Part II: Dimensionality reduction
  - Feature selection
  - Feature extraction

What is dimensionality reduction?

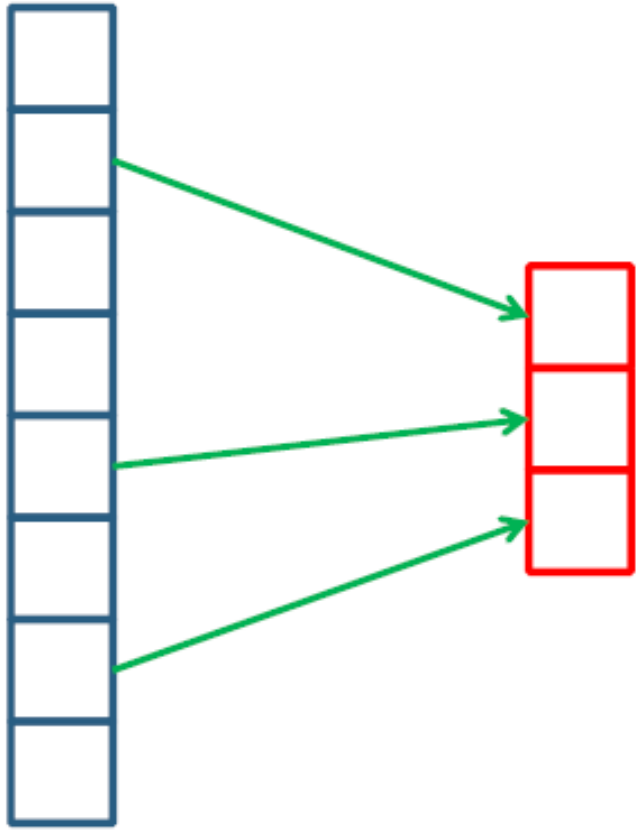


What is dimensionality reduction?

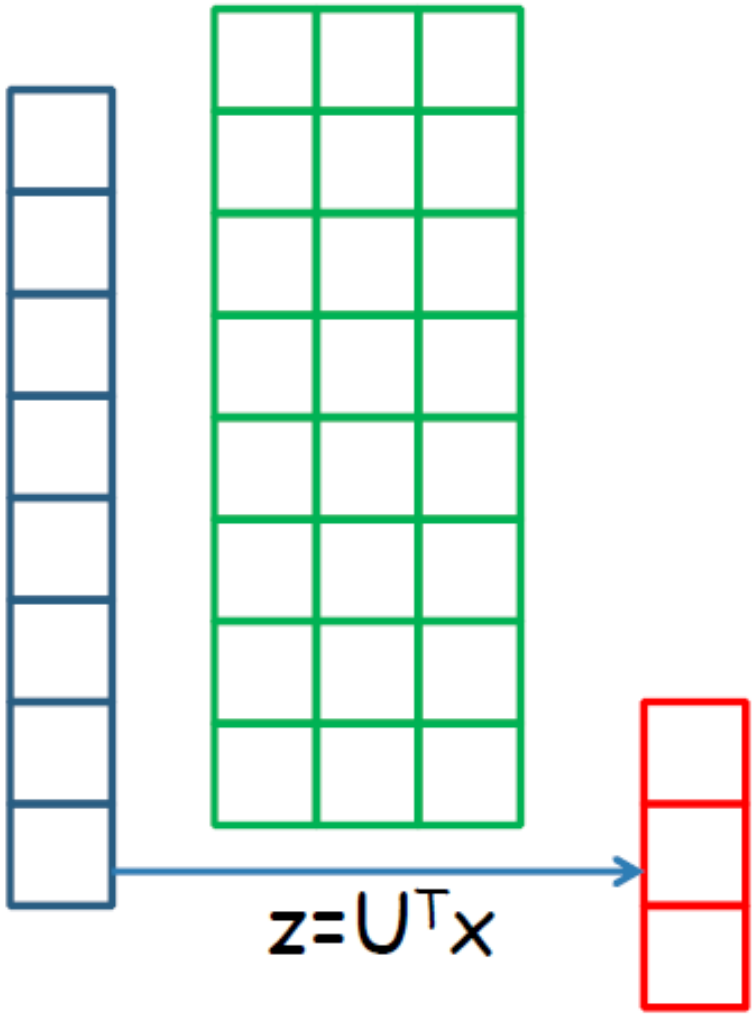


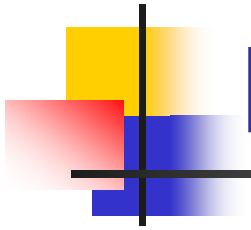
- Generally speaking, dimensionality reduction techniques can be classified into two categories:
  - **Feature selection**: to select a subset of most representative or discriminative features from the input feature set;
  - **Feature extraction**: to transform the original input features to a lower dimensional subspace through a projection matrix.

Feature selection



Feature extraction

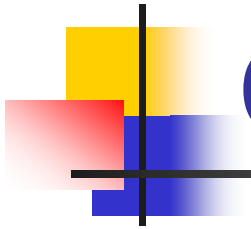




Feature extraction

- Linear (PCA, LDA, etc.)
- Kernel-based (KPCA, KLDA, etc.)
- Manifold learning (LLE, ISOMAP, etc.)
- Tensor (2DPCA, 2DLDA , etc.)
- ...

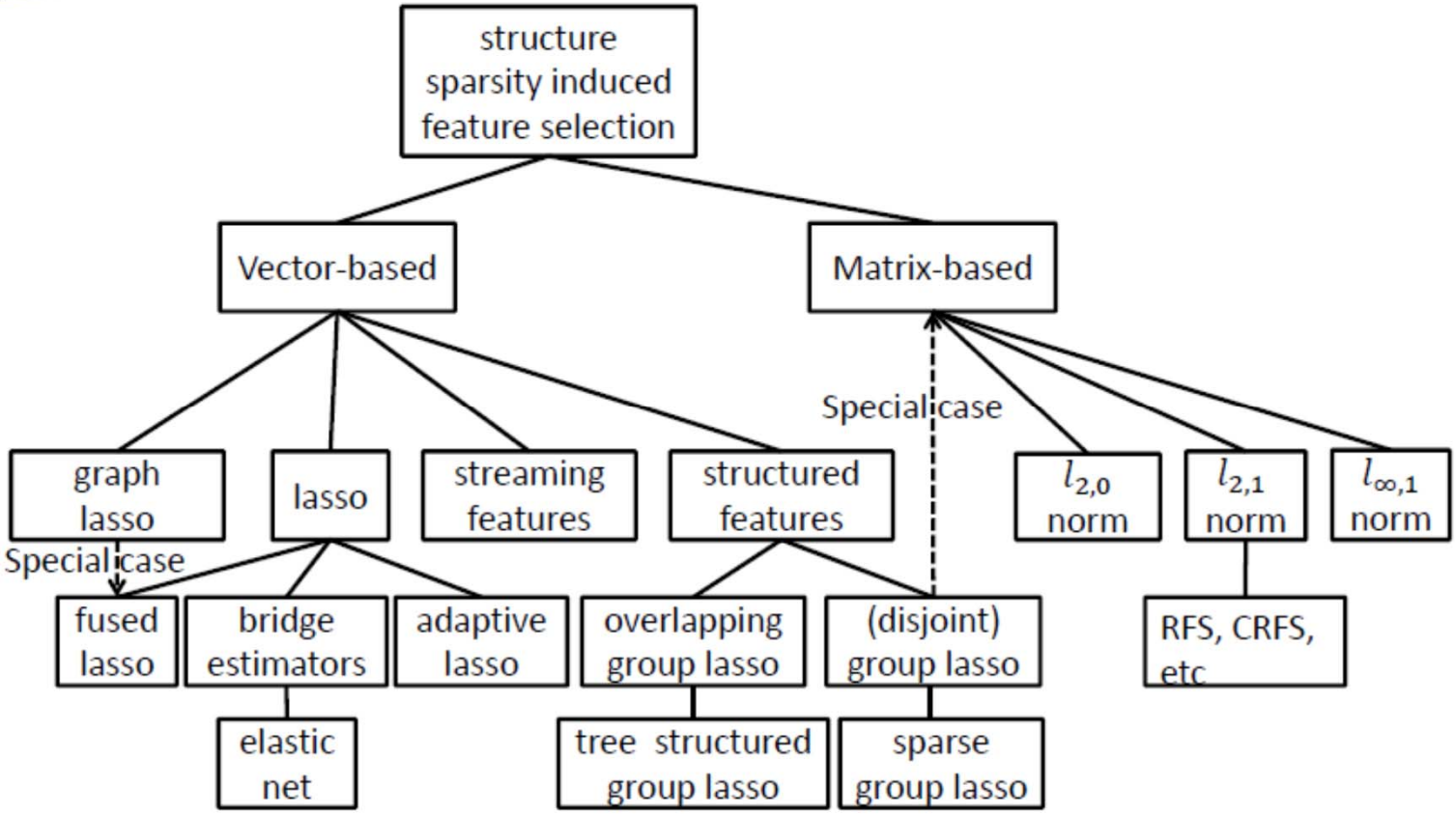
Please see the Introduction of the following reference:
Jie Gui, Zhenan Sun, Wei Jia, Rongxiang Hu, Yingke Lei and Shuiwang Ji, "Discriminant Sparse Neighborhood Preserving Embedding for Face Recognition", Pattern Recognition, vol. 45, no.8, pp. 2884–2893, 2012



Outline

- Part I: Classification
- Part II: Dimensionality reduction
  - Feature selection
  - Feature extraction

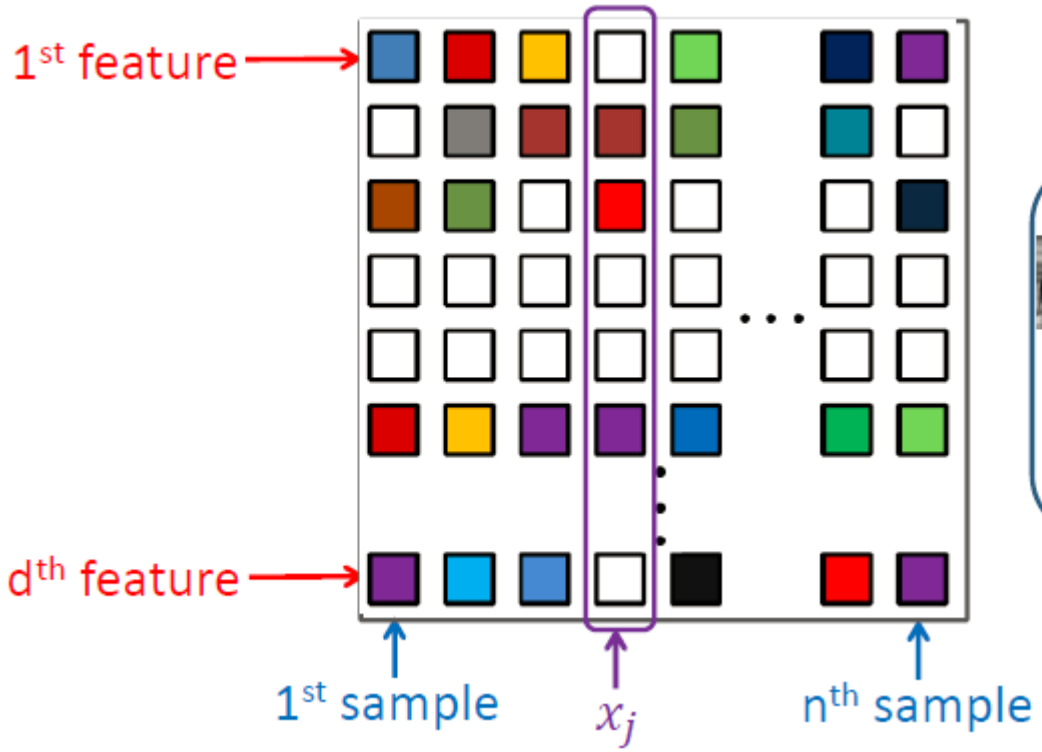
Summary

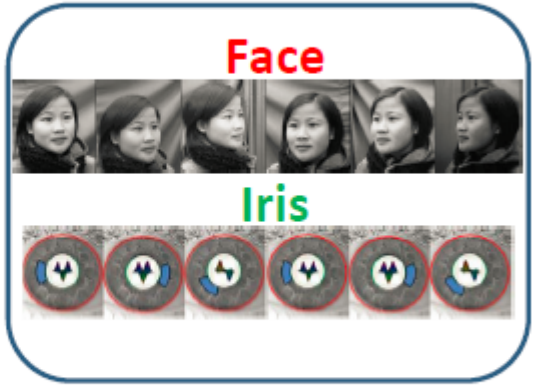


A taxonomy of structure sparsity induced feature selection

Notations

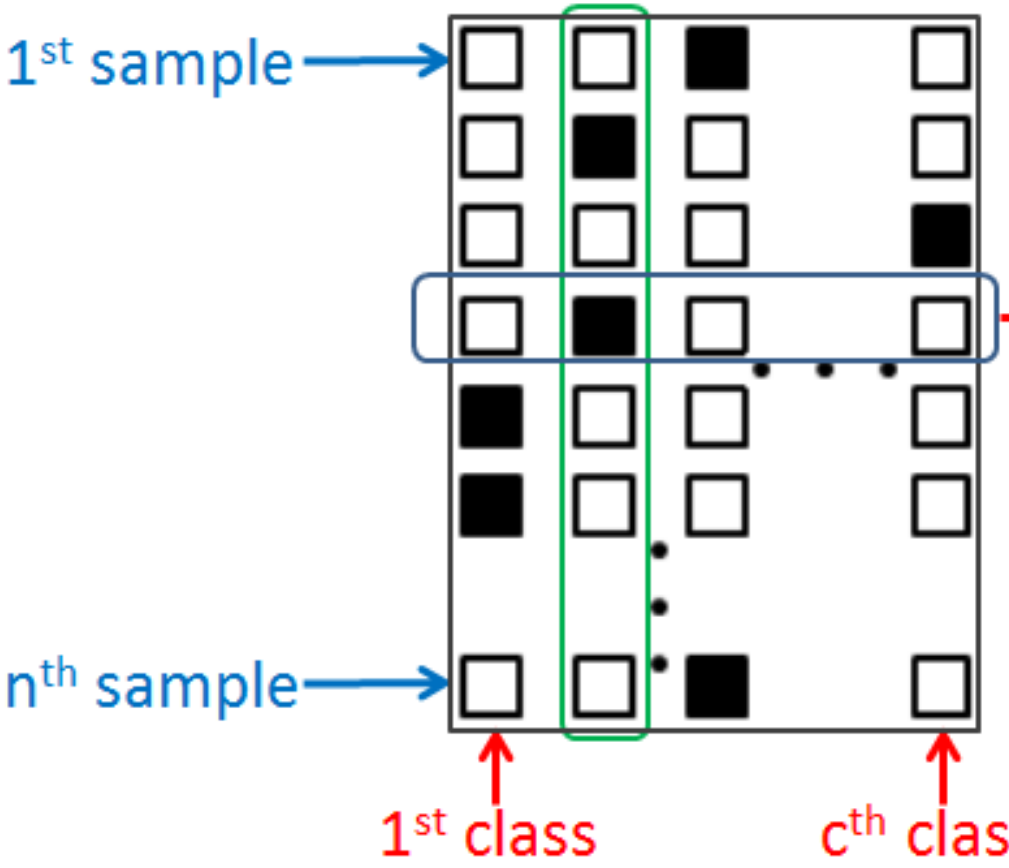
- Data matrix $X = \{x_1, \cdots, x_n\} \in R^{d \times n}$





Notations

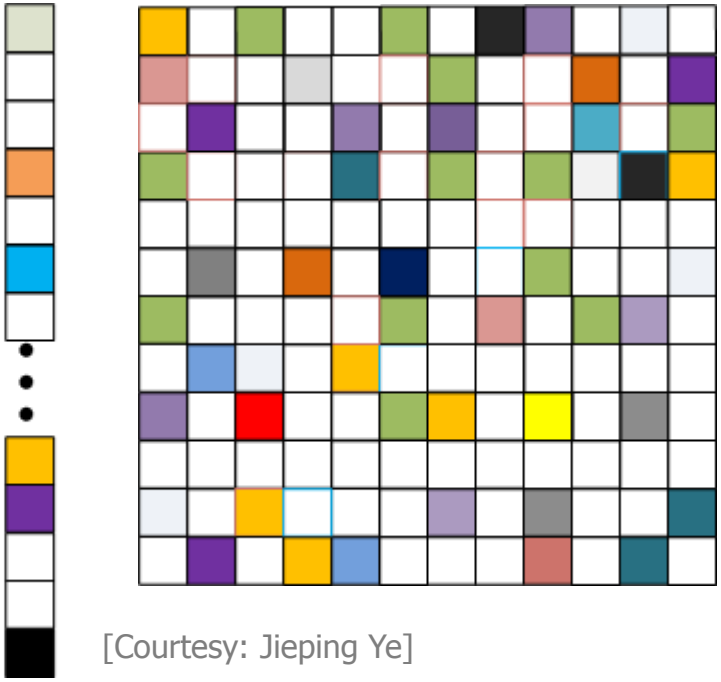
- Label matrix $Y = [y_1, \cdots, y_n]^T \in R^{n \times c}$



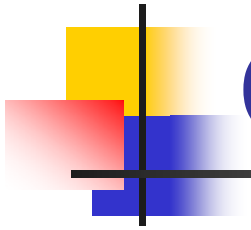
Define Y with $Y_{ij} = 1$ if x_i is from the jth class and $Y_{ij} = 0$ otherwise.

What is sparsity?

- Many machine learning and data mining tasks can be represented using a vector or a matrix.
- "Sparsity" implies many zeros in a vector or a matrix.

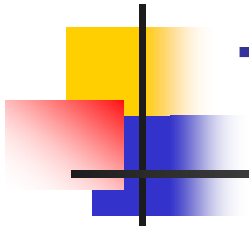


[Courtesy: Jieping Ye]



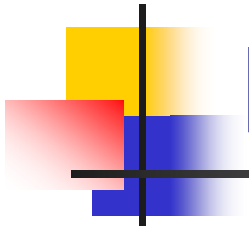
Contents

- Vector-based feature selection
  - ✓ Lasso
  - ✓ Various variants of lasso
  - ✓ Disjoint group lasso
  - ✓ Overlapping group lasso
- Matrix-based feature selection
  - ✓ $l_{2,0}$ –norm, $l_{2,1}$-norm, $l_{\infty,1}$-norm, etc



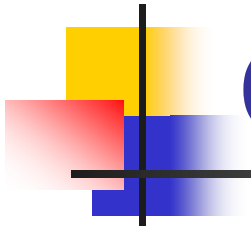
Task-driven feature selection

- Multi-task feature selection
- Multi-label feature selection
- Multi-view feature selection
- Joint feature selection and classification
- Joint feature selection and clustering
- …



Difference from previous work

- Review of sparsity.
  - eg. Wright et al. [Proceedings of the IEEE, 2010]
  - Cheng et al. [Signal Processing, 2013], etc.
- Review of feature selection.
  - Anne-Claire Haury et al. [PLoS ONE, 2011]
  - Verónica Bolón-Canedo et al. [KAIS, 2013], etc.

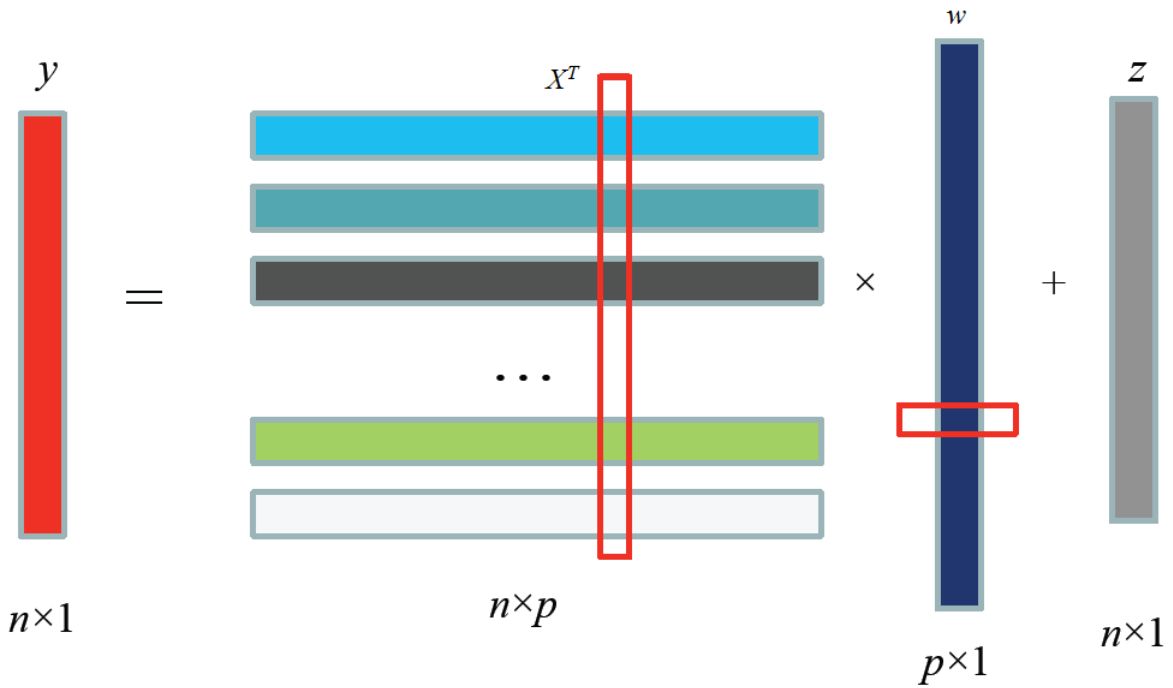


Contributions

- Providing a survey on structure sparsity induced feature selection (SSFS).
- Exploiting the relationships among different kinds of SSFS.
- Evaluating several representative SSFS methods.
- Summarizing main challenges and problems of current studies, and point out some future research directions.

Lasso

(Tibshirani, 1996, Chen, Donoho, and Saunders, 1999)

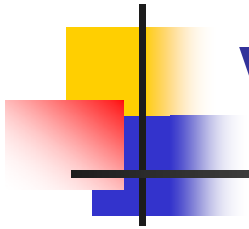


minimize

$$\|y - X^T w\|_2^2 + \gamma \|w\|_1$$

$$penalty(w) = \|w\|_1$$

[Courtesy: Jieping Ye]



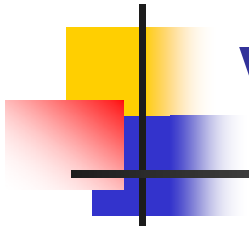
Various variants of lasso

- Adaptive lasso:

$$penalty(w) = \sum_{i=1}^{d} a_i |w_i|$$

- Fused lasso:

$$penalty(w) = \|w\|_1 + \alpha \sum_{i=2}^{d} |w_i - w_{i-1}|$$



Various variants of Lasso

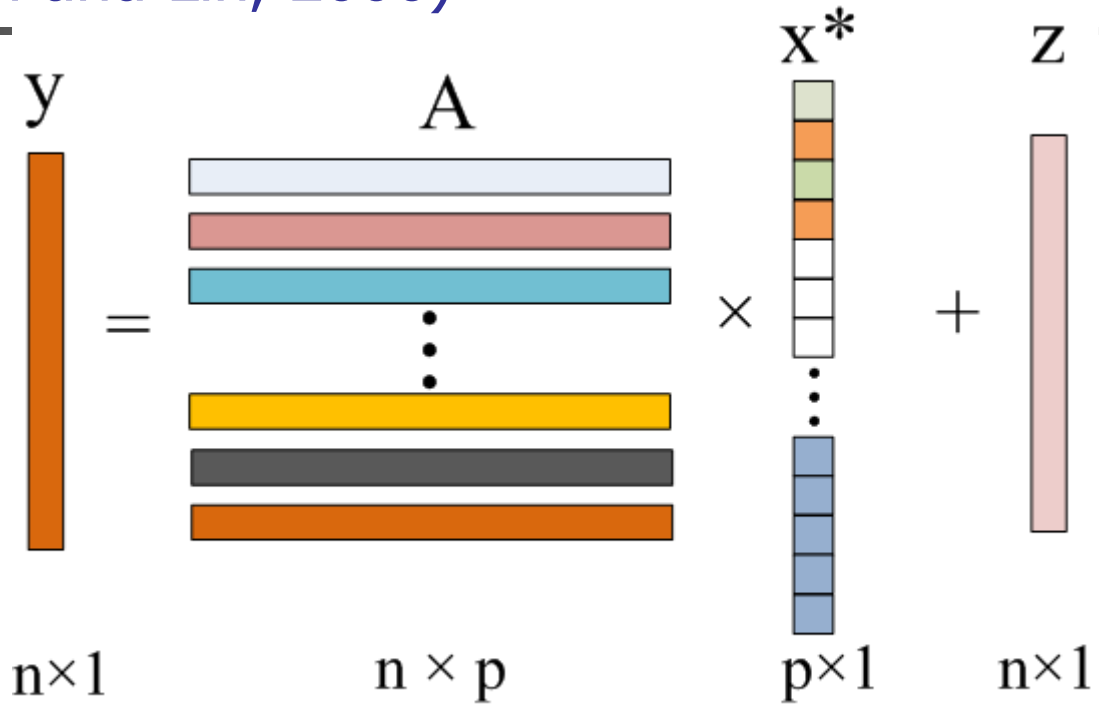
- Bridge estimator:

$$penalty(w) = \sum_{i=1}^{d} |w_i|^{\gamma}$$

- Elastic net:

$$penalty(w) = \alpha \sum_{i=1}^{d} |w_i| + (1-\alpha) \sum_{i=1}^{d} w_i^2$$

Disjoint group lasso
(Yuan and Lin, 2006)



p >>n

$$\min \frac{1}{2}\|Ax - y\|_2^2 + \lambda \sum_{i=1}^{g} d_i \|x_{G_i}\|_2$$

[Courtesy: Jieping Ye]

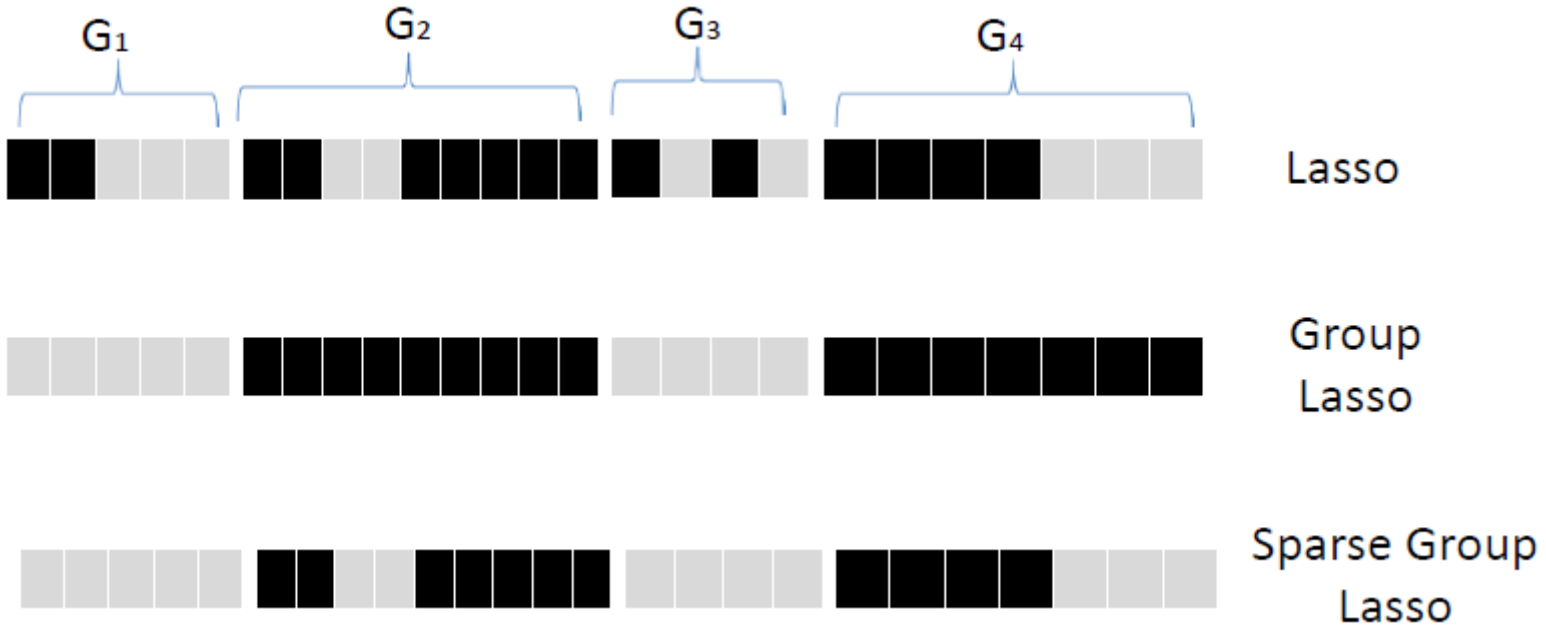
Sparse group lasso

- Sparse group lasso combines both lasso and group lasso

$$penalty(w) = (1-\alpha)\|w\|_1 + \alpha \sum_{i=1}^{k} \beta_i \|w_{G_i}\|_q$$

- Lasso and group lasso are special cases of sparse group lasso

Lasso, group lasso and sparse group lasso

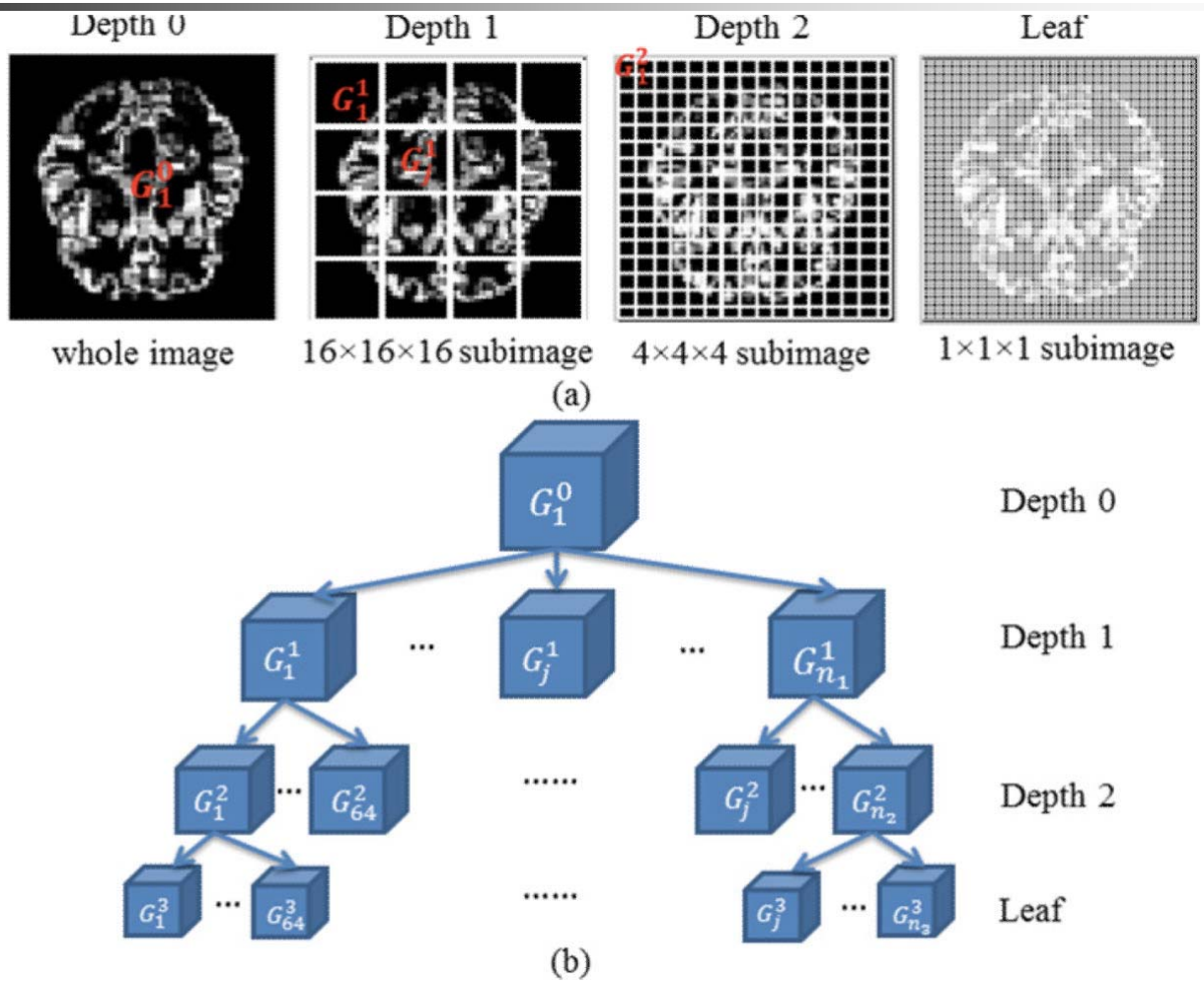


Features can be grouped into 4 disjoint groups {G1,G2,G3,G4}. Each cell denotes a feature and light color represents the corresponding cell with coefficient zero.

[Courtesy: Jiliang Tang]

Overlapping group lasso

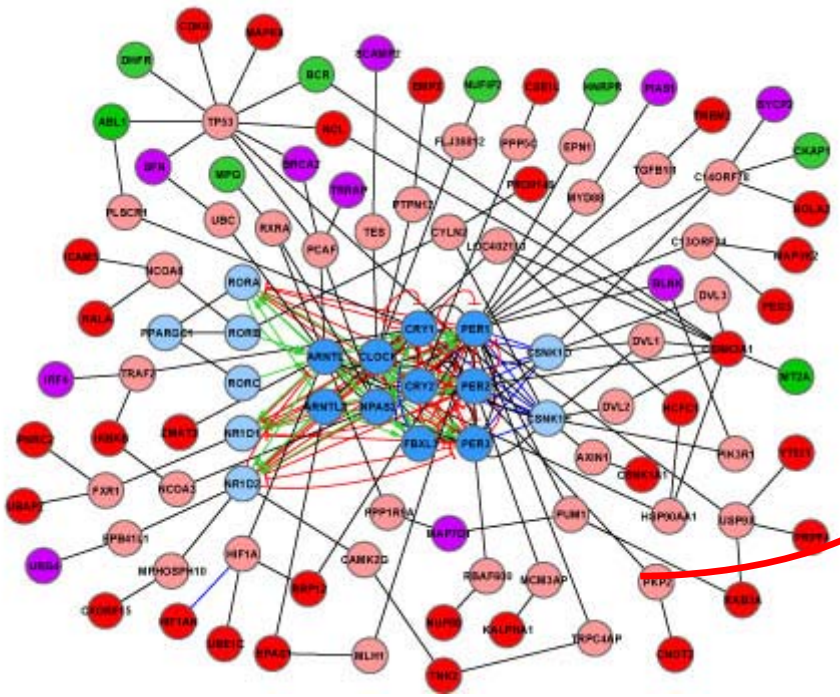
(Zhao, Rocha and Yu, 2009; Kim and Xing, 2010; Jenatton et al., 2010; Liu and Ye, 2010)



[Courtesy: Jieping Ye]

Graph lasso

(Slawski et al, 2009; Li and Li, 2010; Li and Zhang 2010)



$$\|w\|_1 + \alpha \sum_{(i,j)\in E} B_{ij}\left(w_i - w_j\right)^2$$

[Courtesy: Jieping Ye]

Matrix-based feature selection

- The $l_{r,p}$-norm of a matrix
- The physical meaning of $l_{r,p}$ -norm of a matrix
- $l_{2,1}$-norm based feature selection
- $l_{\infty,1}$ –norm based feature selection
- $l_{2,0}$-norm based feature selection

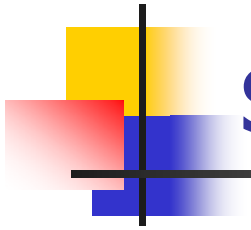
The $l_{r,p}$-norm of a matrix

- $\|A\|_{r,p} = \|\left(\|A^1\|_r, \cdots, \|A^i\|_r, \cdots, \|A^u\|_r\right)\|_p$
- $l_{2,1}$-norm
- $l_{2,0}$-norm
- $l_{2,p}$-norm
- $l_{\infty,1}$-norm
- ...

The physical meaning of $l_{r,p}$ - norm

- If we require most rows of A to be zero, we have $0 \leq p \leq 1$.
- The choice of r depends on what kind of correlation assumption among classes.
  - **Positive** correlation: $1 < r \leq \infty$
  - **Negative** correlation: $0 \leq r \leq 1$

$l_{2,1}$-norm based feature selection



- Efficient and robust feature selection via joint $l_{2,1}$ -norms minimization (RFS)
- Correntropy induced robust feature selection
- Feature selection via joint embedding learning and sparse regression
- Joint feature selection and subspace learning
- …

Efficient and robust feature selection

(Nie et al., 2010)

$$\min_W \sum_{i=1}^{n} \|Y^i - X_i^T W\|_2 + \alpha \|W\|_{2,1}$$

$$= \min_W \|Y - X^T W\|_{2,1} + \alpha \|W\|_{2,1}$$

Feature selection

Least squares regression

Correntropy induced robust feature selection

(He et al., 2012)

$$\min_{W} \sum_{i=1}^{n} \phi\left(\left(X^{T}W - Y\right)^{i}\right) + \lambda \left\|W\right\|_{2,1}$$

the robust M-estimator

Feature selection

FS via joint embedding learning and sparse regression

(Hou et al., 2011; Hou et al., 2014)

Regression to low dimensional representation

$$\min_{W, ZZ^T = I_{m\times m}} tr\left(ZLZ^T\right) + \beta \left\| W^T X - Z \right\|_2^2 + \alpha \left\| W \right\|_{r,p}^p$$

Laplacian matrix

Feature selection

Joint feature selection and subspace learning
(Gu et al., 2011)

$$\min_W \|W\|_{2,1} + \alpha tr(W^T X L X^T W)$$
$$s.t.\ \ W^T X D X^T W = I$$

- First term : Feature selection
- Second term:
  - $\min_W tr(W^T X L X^T W)$ $s.t.\ \ W^T X D X^T W = I$
  - the objective function of graph embedding (Yan et al., 2007)

$l_{\infty,1}$-norm based feature selection

(Masaeli et al., 2010)

- $\min_W tr\left((W^T S_w W)^{-1} W^T S_b W\right) + \alpha \|W\|_{\infty,1}$

Linear discriminant analysis

Feature selection

$l_{2,0}$-norm based feature selection

(Cai et al, 2013)

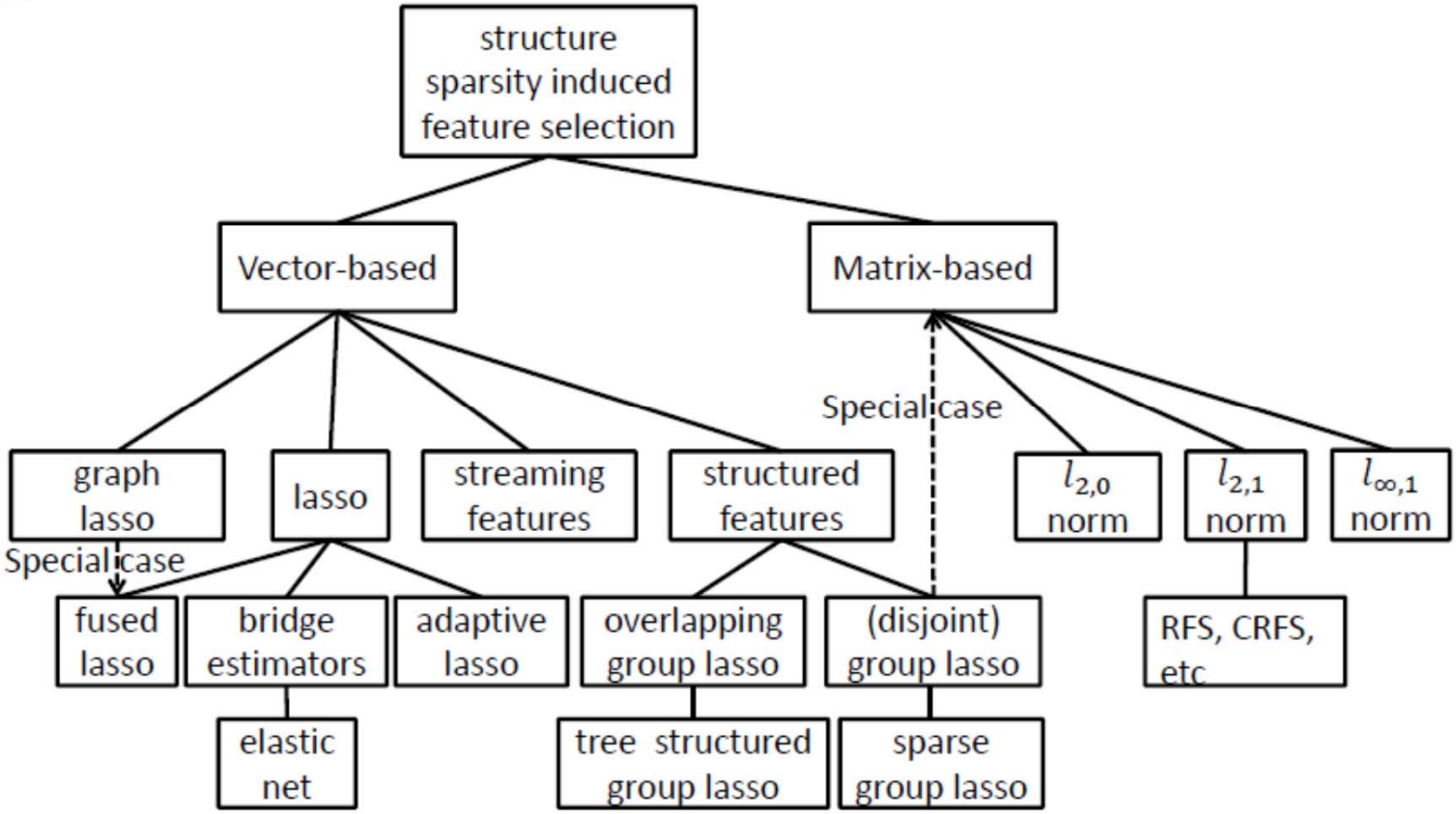
- $$\min_{W,b} \|Y - X^T W - e_n b\|_{2,1}$$

  $$s.t. \quad \|W\|_{2,0} = k$$

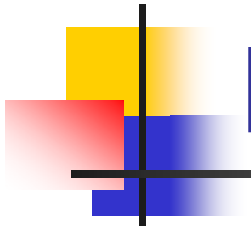
  the bias vector

- Since the regularization parameter of this method has the explicit meaning, i.e., the number of selected features, it alleviates the problem of tuning the parameter exhaustively.

Summary



A taxonomy of structure sparsity induced feature selection



Experiments

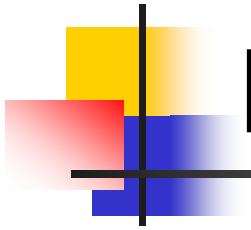
- Compared methods – 9 traditional methods
  - **Chi square**
  - **Data variance**
  - **Fisher score**
  - **Gini index**
  - **Information Gain**
  - **mRMR**
  - **ReliefF**
  - **T-test**
  - **Wilcoxon rank-sum test**

Software package

Huan Liu

（刘欢）

http://featureselection.asu.edu/software.php

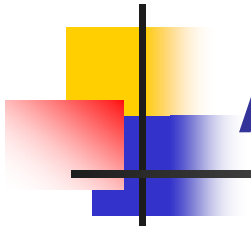


Experiments

- Compared methods – 5 structured sparsity based
  - **CRFS (He, 2012)**
  - **DLSR-FS (Xiang, 2012)**
  - **l_1 (Destrero, 2007)**
  - **$l_{2,0}$ (Cai, 2013)**
  - **RFS (Nie,2010)**
  - **UDFS (Yang, 2011)**

Data set

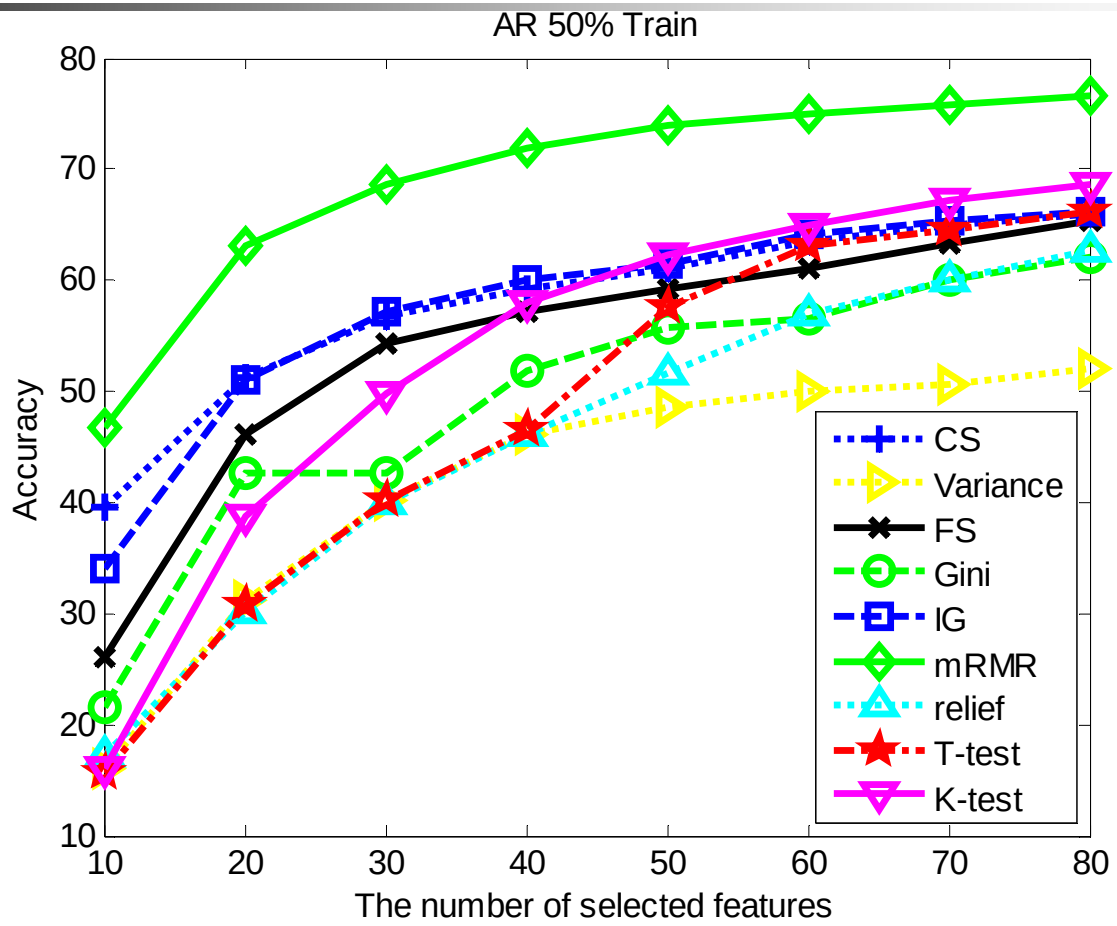
Data set	Category	Total number	Classes	Dimension
AR	face	400	40	644
Umist	face	575	20	2576
Coil20	image	1440	20	256
vehicle	UCI	846	4	18
Lung	Microarray	203	5	3312
TOX-171	Microarray	171	4	5748
MLL	Microarray	72	3	5848
CAR	Microarray	174	11	**9182**



AR data set

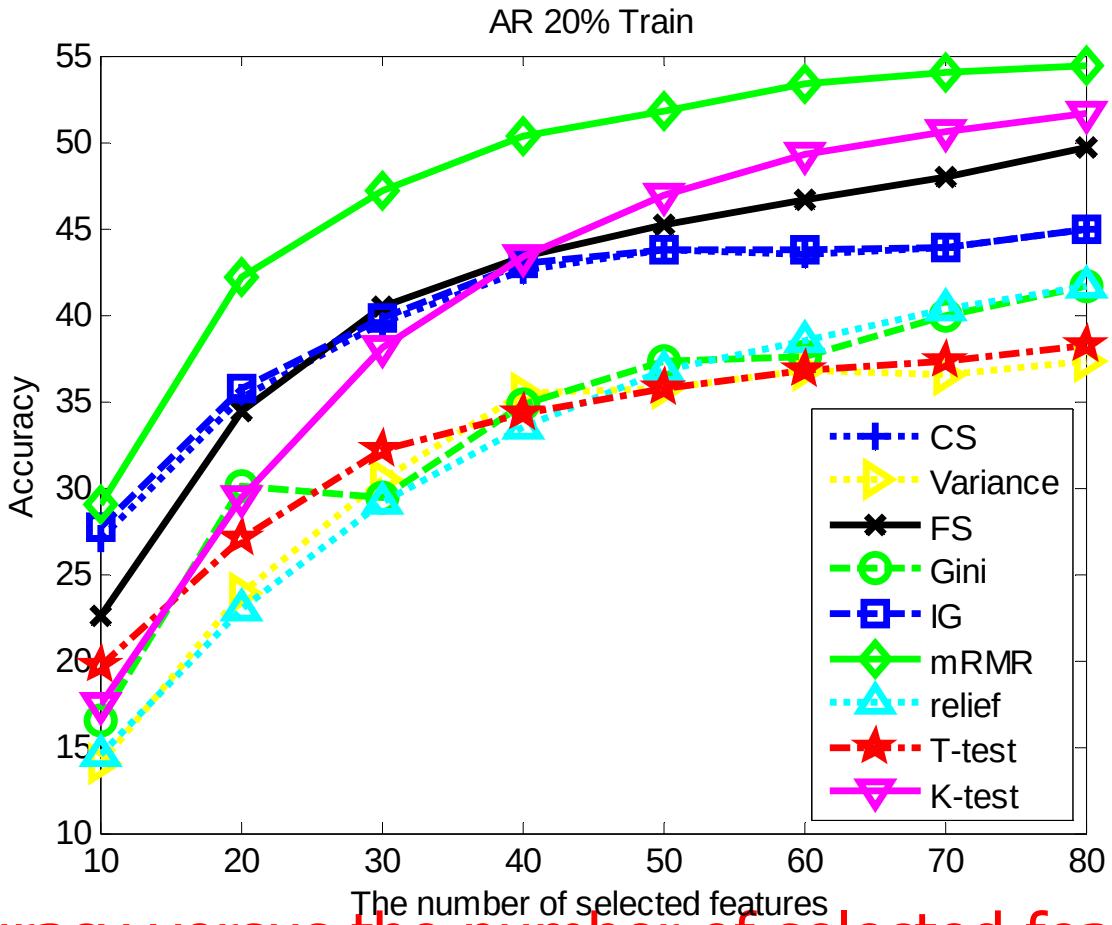
- **120** classes, **7** examples for each classes, **3** examples per class for training
- **20** random splitting
- In each random splitting, **cross validation** was used to tune the parameter of linear SVM and feature selection algorithms

Results of AR face data set



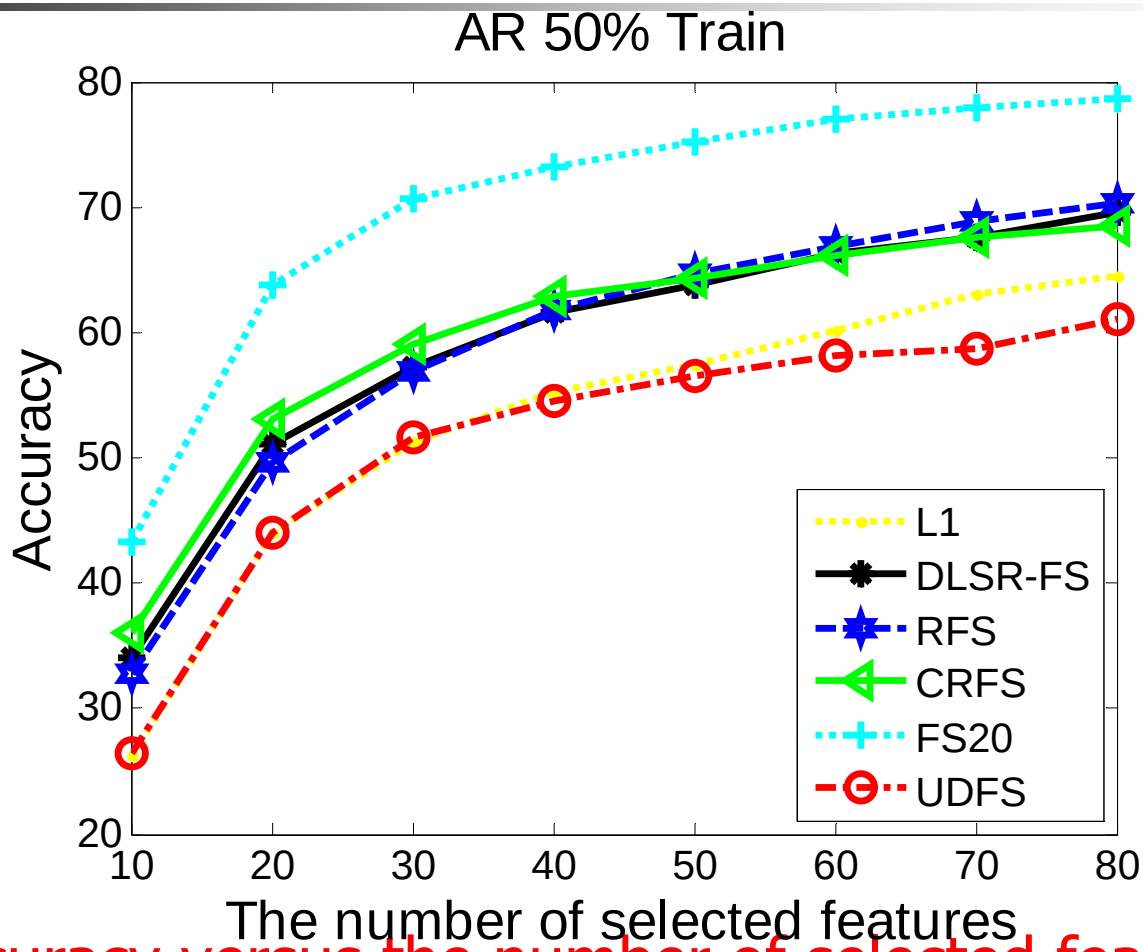
Accuracy versus the number of selected features.

Results of AR face data set



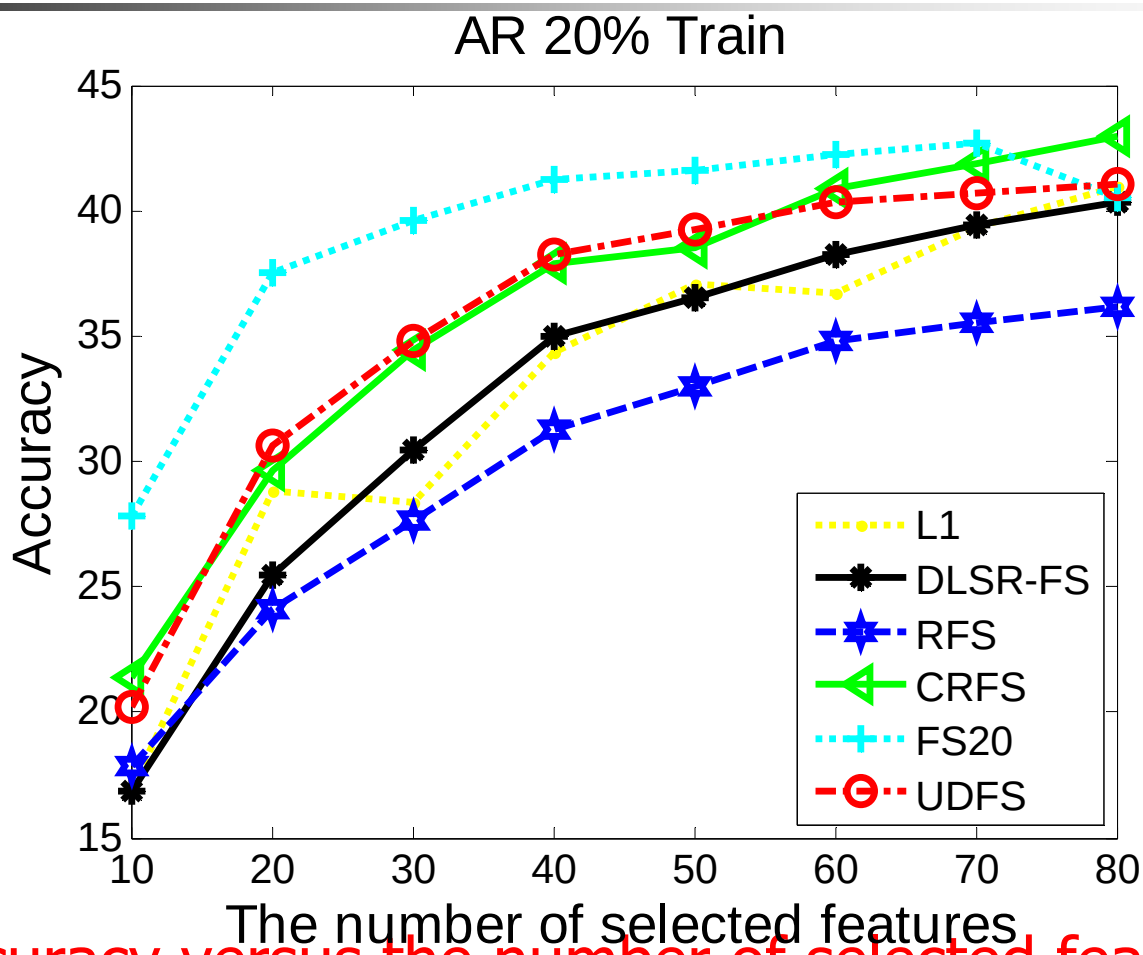
Accuracy versus the number of selected features.

Results of AR face data set

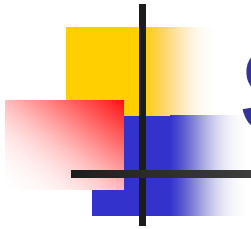


Accuracy versus the number of selected features.

Results of AR face data set

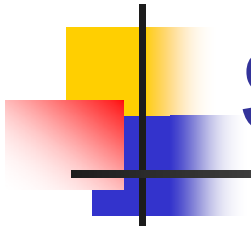


Accuracy versus the number of selected features.



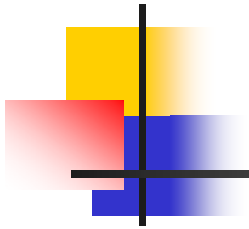
Some preliminary analyses

- Generally speaking, mRMR performs better than other traditional feature selection methods.
- No single method can always beat other methods.
- Traditional vs Sparse
  - Sparse wins 15 times in all 22 experiments.



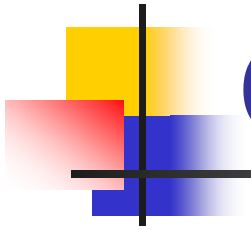
Some preliminary analyses

- However, the improvement of the structure sparsity induced feature selection methods over the traditional methods is marginal.
- Future research directions?



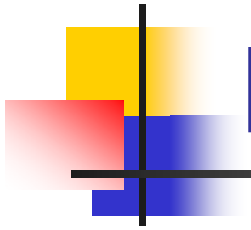
- This work is accepted in IEEE Transactions on Neural Networks and Learning Systems:

Jie Gui, Zhenan Sun, Shuiwang Ji, DachengTao, Tieniu Tan, "Feature Selection Based on Structured Sparsity: A Comprehensive Study", IEEE Transactions on Neural Networks and Learning Systems，DOI:10.1109/TNNLS.2016.2551724.



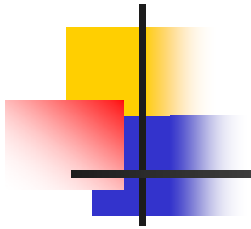
Outline

- Part I: Classification
- Part II: Dimensionality reduction
  - Feature selection
  - Feature extraction

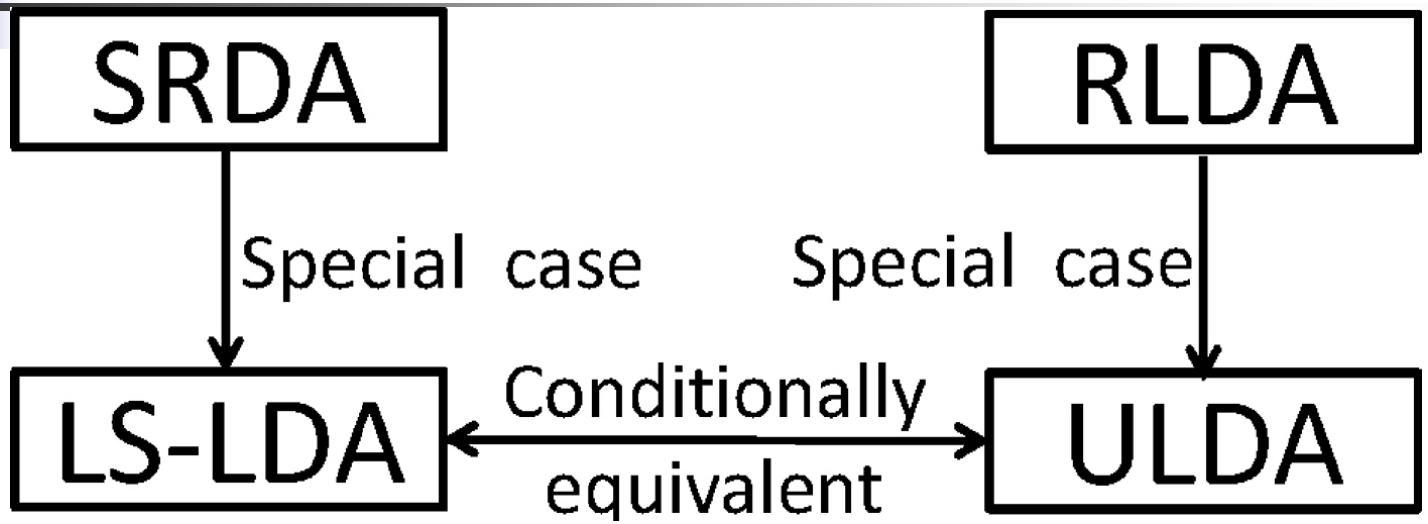


Feature extraction

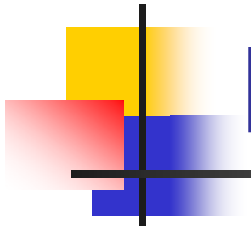
- **How to estimate the regularization parameter for spectral regression discriminant analysis and its kernel version?**
- An optimal set of code words and correntropy for rotated least squares regression



- Spectral regression discriminant analysis (SRDA) has recently been proposed as an efficient solution to large-scale subspace learning problems.
- There is a tunable regularization parameter in SRDA, which is critical to algorithm performance. However, how to automatically set this parameter has not been well solved until now.



Jie Gui, et al., "How to estimate the regularization parameter for spectral regression discriminant analysis and its kernel version?", IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 2, pp. 211-223, 2014



Feature extraction

- How to estimate the regularization parameter for spectral regression discriminant analysis and its kernel version?
- **An optimal set of code words and correntropy for rotated least squares regression**

Least squares regression (LSR)

- LSR solves the following problem to obtain the projection matrix $W \in R^{d\times c}$ and bias $b \in R^{c\times 1}$

$$\min_{W,b} \sum_{i=1}^{n} \left\| W^T x_i + b - y_i \right\|_2^2 + \lambda \left\| W \right\|_F^2$$

- The above equation can be equivalently rewritten as follows:

$$\min_{W,b} \left\| X^T W + e_n b^T - Y \right\|_2^2 + \lambda \left\| W \right\|_F^2$$

- LSR is sensitive to outliers.

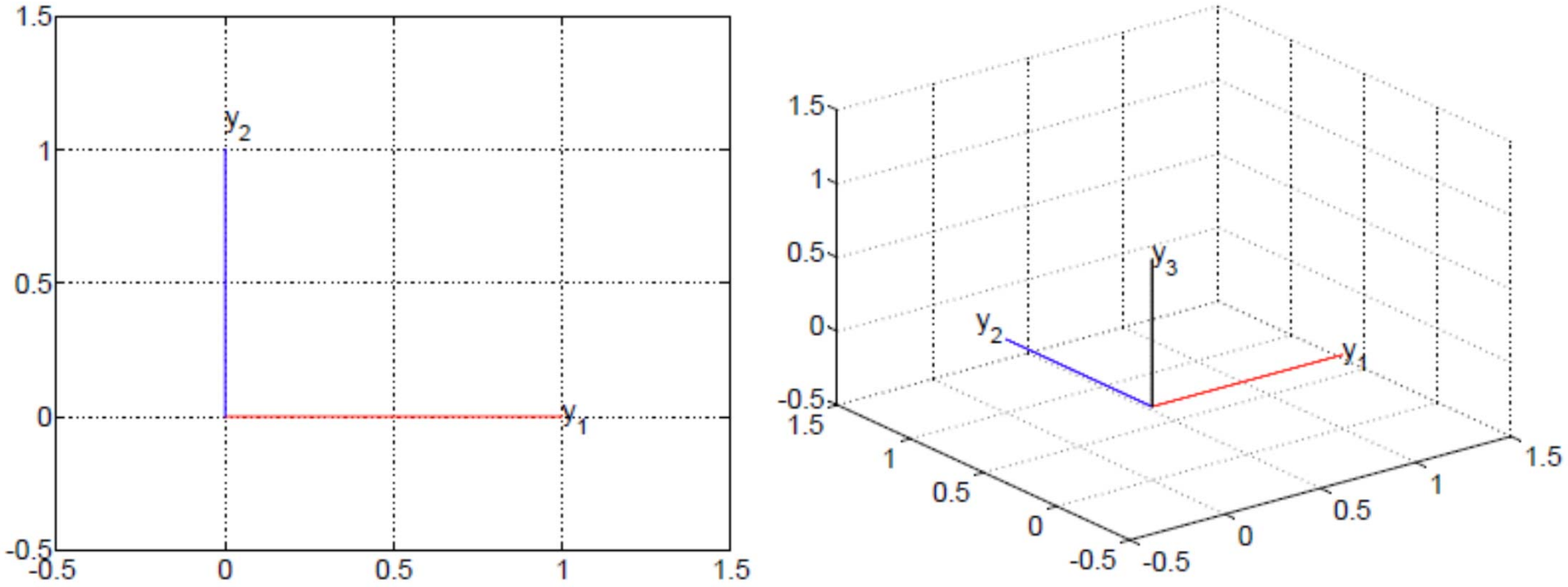
Traditional set of code words

- In traditional LSR, the ith row and jth column element of Y, i.e., Y_{ij}, is defined as

$$Y_{ij} = \begin{cases} 1, & \text{if } x_i \text{ is in the } j\text{th class} \\ 0, & \text{otherwise} \end{cases}$$

- For example, the traditional set of code words for two classes and three classes are

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$



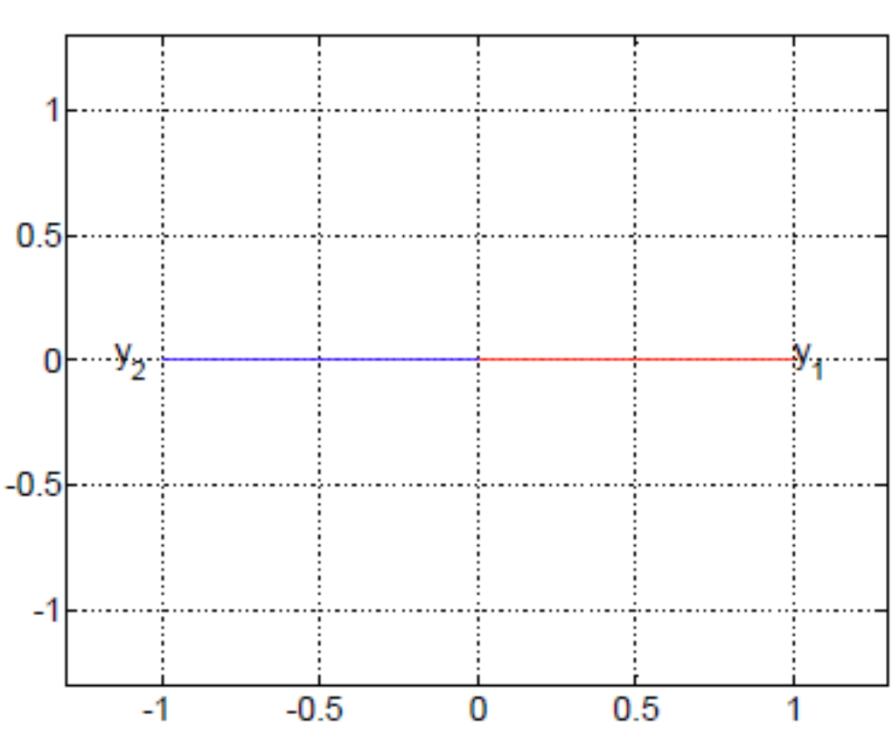
(a) two classes (b) three classes

Fig.1. The traditional set of code words

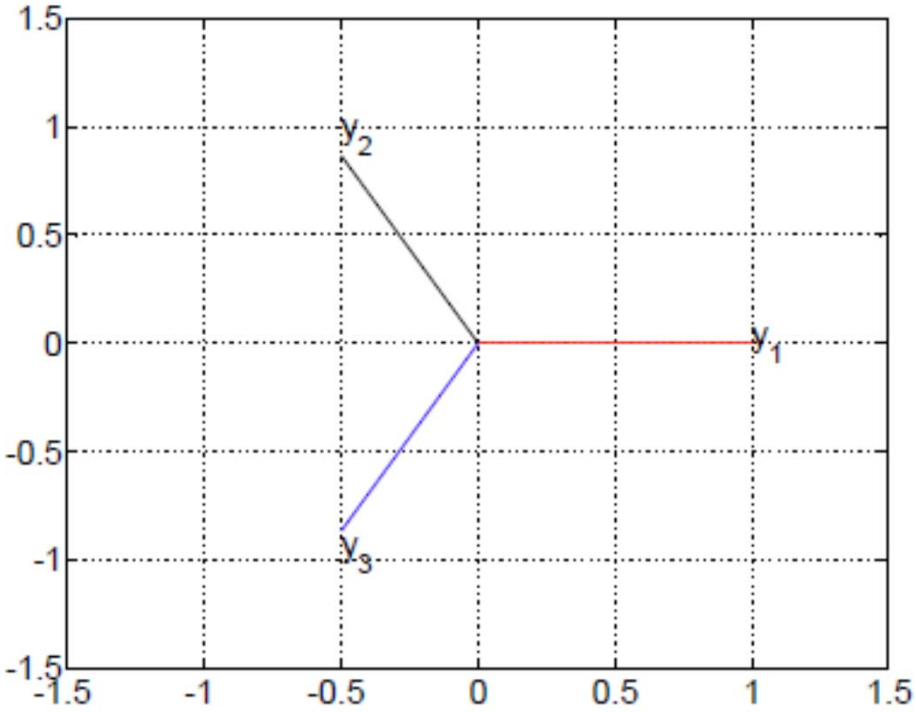
Deficiencies of traditional set of code words

- The distance between y_1 and y_2 is not the maximum in the two-dimensional space. The unit point pair $[-1 \quad 0]^T$ and $[1 \quad 0]^T$ is one of the farthest unit point pairs in the two-dimensional space. Obviously, 0 is **redundant**, -1 and 1 can be used instead.
- Here, we introduced an optimal set of code words, which was proposed in :

  Mohammad J. Saberian and Nuno Vasconcelos. "Multiclass Boosting: Theory and Algorithms," in ***Neural Information Processing Systems***, 2011



(a) two classes



(b) three classes

Fig. The optimal set of code words

Example 1

- The traditional set of code words for two classes and the new set of code words for two classes are

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \end{bmatrix},$$

  respectively.
- Length: 2, 1
- Distance: $\sqrt{2}, 2$

Example 2

- The traditional set of code words for three classes and the new set of code words for three classes are

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \\ 1 & 0 \end{bmatrix},$$

  respectively.
- Length: 3, 2
- Distance: $\sqrt{2}, \sqrt{3}$

Advantages of optimal set of code words

- The length of this new set of code words is **less**;
- The distance between different classes is **larger**.

Correntropy

- LSR is sensitive to outliers. For better robustness, correntropy is introduced and thus the objective function is defined as follows:

$$\min_{W,b,M} \sum_{i=1}^{n} \phi\left(\left(X^T W + e_n b^T - Y - G \odot M\right)^i\right) + \lambda \|W\|_F^2$$

where $\odot$ is a Hadamard product operator of matrices. The term G is defined as

$$G_{ij} = \begin{cases} 1, & \text{if } x_i \text{ is in the } j\text{th class} \\ -1, & \text{otherwise} \end{cases}$$

Rotation transformation invariant constraint

- Since the commonly utilized distance metrics in the subspace, such as Cosine and Euclidean, are invariant to rotation transformation, additional freedom in rotation can be introduced to promote sparsity without sacrificing accuracy.
- With an additional rotation transformation matrix R, our new formulation is defined as:

$$\min_{W,b,M,R} \sum_{i=1}^{n} \phi\left(\left(X^T W + e_n b^T - YR - G \odot M\right)^i\right) + \lambda \left\|W\right\|_F^2$$

$$s.t. \quad R^T R = I$$

Reference

- **Jie Gui,** Tongliang Liu, Dacheng Tao, Zhenan Sun, Tieniu Tan, "Representative Vector Machines: A unified framework for classical classifiers", IEEE Transactions on Cybernetics, vol. 46, no. 8, pp. 1877-1888, 2016
- **Jie Gui**, Zhenan Sun, Shuiwang Ji, DachengTao, Tieniu Tan, "Feature Selection Based on Structured Sparsity: A Comprehensive Study", IEEE Transactions on Neural Networks and Learning Systems，DOI:10.1109/TNNLS.2016.2551724.

- **Jie Gui**, et al., "How to estimate the regularization parameter for spectral regression discriminant analysis and its kernel version?", IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 2, pp. 211-223, 2014
- **Jie Gui**, Zhenan Sun, Wei Jia, Rongxiang Hu, Yingke Lei and Shuiwang Ji, "Discriminant Sparse Neighborhood Preserving Embedding for Face Recognition", Pattern Recognition, vol. 45, no.8, pp. 2884–2893, 2012
- **Jie Gui**, Zhenan Sun, Guangqi Hou, Tieniu Tan, "An optimal set of code words and correntropy for rotated least squares regression", International Joint Conference on Biometrics, pp. 1-6, 2014

Code

- http://www.escience.cn/people/guijie/index.html