Ensemble-Based Tracking: Aggregating Crowdsourced Structured Time Series Data

Presented by Naiyan Wang
Hongkong University of Science and Technology

June 24, 2014

In this paper, we start from a real application in computer vision: single object visual tracking (a.k.a object tracking).

(Zhang *et al.* 2012)

Motivating Example

1. Different tracker focuses on different aspects of this problem, and always yields diverse results.
2. A nature question is: Could we combine the results of different trackers to get a better result?

Motivating Example

1. Different tracker focuses on different aspects of this problem, and always yields diverse results.
2. A nature question is: Could we combine the results of different trackers to get a better result?

Taking a closer look...

The key properties of this problem:

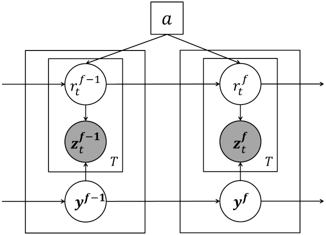
1. They are time series data over frames.
2. The output of each contributor is structured data (i.e. bounding box).

Taking a closer look...

The key properties of this problem:

1. They are time series data over frames.
2. The output of each contributor is structured data (i.e. bounding box).

Model Design



The joint probability is given by

$$\prod_{f=2}^{F}\prod_{t=1}^{T} p(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f)\, p(\mathbf{y}^f \mid \mathbf{y}^{f-1})\, p(r_t^f \mid r_t^{f-1})\, p(r_t^f \mid a),$$

Model Design

Observation Likelihood: To model the structured property of bounding box, we further decompose it using the normalized central-pixel error metric $\mathbb{D}(\cdot,\cdot)$ and the overlap rate metric $\mathcal{O}(\cdot,\cdot)$:

$$p(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) = p_c(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f)\, p_o(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f),$$

where

$$\begin{aligned} p_c(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) &\sim \mathcal{N}(\alpha\mathbb{D}(\mathbf{z}_t^f, \mathbf{y}^f) \mid 0, r_t^f) \\ p_o(\mathbf{z}_t^f \mid \mathbf{y}^f, r_t^f) &\sim \mathrm{TrunExp}(1 - \mathcal{O}(\mathbf{z}_t^f, \mathbf{y}^f) \mid r_t^f), \end{aligned}$$

Model Design

Transition Probability: Since the reliability r_t^f should be non-negative, we model it using a Gamma distribution.

$$p(y_d^f \mid y_d^{f-1}) \sim \mathcal{N}(y_d^f \mid y_d^{f-1}, \sigma_d^2),$$
$$p(r_t^f \mid r_t^{f-1}) \sim \mathcal{G}\left(r_t^f \mid k, \frac{r_t^{f-1}}{k}\right).$$

Moreover, the parameters are chosen that $\mathbb{E}[r_t^f] = r_t^{f-1}$.

Model Design

Reliability Prior: To avoid the model overfits some individual trackers, we use a time-invariant exponential distribution to penalize high reliability:

$$p(r_t^f \mid a) \sim \mathrm{Exp}(r_t^f \mid a).$$

Model Inference

1. The nature of visual tracking problem calls for online inference algorithm for the model.
2. However, a naïve particle filter algorithm incurs high computational cost of $O(NM^T)$.

Model Inference

1. The nature of visual tracking problem calls for online inference algorithm for the model.
2. However, a naïve particle filter algorithm incurs high computational cost of $O(NM^T)$.

Model Inference

1. Fortunately, the posterior distribution could be decomposed as:

$$p(\mathbf{y}^{f-1}, \mathbf{r}^{f-1} \mid \mathbf{Z}^{1:f-1}) \\ = p(\mathbf{y}^{f-1} \mid \mathbf{Z}^{1:f-1}) \prod_{t=1}^{T} p(r_t^{f-1} \mid \mathbf{y}^{f-1}, \mathbf{z}_t^{1:f-1}).$$

2. This observation induces an efficient *conditional* particle filter algorithm: The posterior in each time step could be approximated by $\left\{ \left(w_{(n)}^{f-1}, \mathbf{y}_{(n)}^{f-1}, \pi_{t,(m,n)}^{f-1}, r_{t,(m,n)}^{f-1} \right) \right\}$.
3. This approach only needs $O(MNT)$ particles compared to $O(NM^T)$ of the naïve approach.

Failure Detection and Implementation

1. We could easily detect the failure of individual tracker by the reliability r_t.
2. If r_t is lower than a threshold, we would like to reinitialize the tracker using the current ensemble result.
3. This requires a two-way communication between trackers and ensemble algorithm.
4. We developed a series of web service interface to support trackers from different platform, different programming language.

Failure Detection and Implementation

1. We could easily detect the failure of individual tracker by the reliability r_t.
2. If r_t is lower than a threshold, we would like to reinitialize the tracker using the current ensemble result.
3. This requires a two-way communication between trackers and ensemble algorithm.
4. We developed a series of web service interface to support trackers from different platform, different programming language.

Failure Detection and Implementation

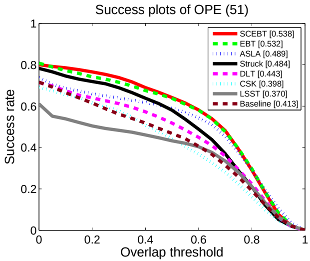
1. We could easily detect the failure of individual tracker by the reliability r_t.
2. If r_t is lower than a threshold, we would like to reinitialize the tracker using the current ensemble result.
3. This requires a two-way communication between trackers and ensemble algorithm.
4. We developed a series of web service interface to support trackers from different platform, different programming language.

Failure Detection and Implementation

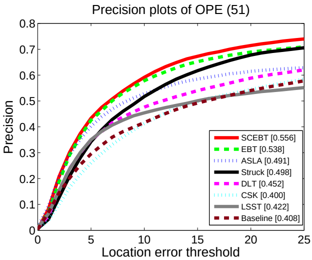
1. We could easily detect the failure of individual tracker by the reliability r_t.
2. If r_t is lower than a threshold, we would like to reinitialize the tracker using the current ensemble result.
3. This requires a two-way communication between trackers and ensemble algorithm.
4. We developed a series of web service interface to support trackers from different platform, different programming language.

Experiments

1. We choose 5 complementary trackers from top performers.
2. We test our algorithm on the largest open benchmark so far. It consists of 50 videos under uncontrolled environment.

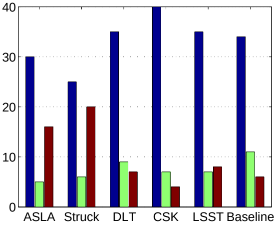


(a) Overlap rate (OR)

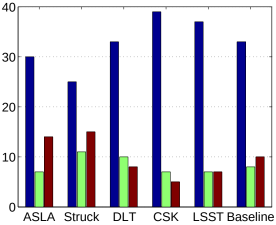


(b) Central-pixel error (CPE)

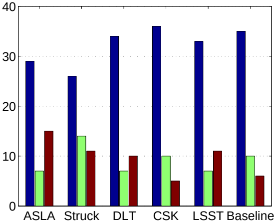
Experiments



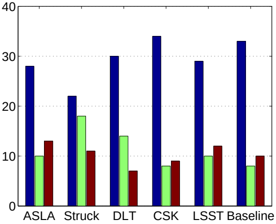
(c) SC-EBT (OR)



(d) EBT (OR)



(e) SC-EBT (CPE)



(f) EBT (CPE)

Conclusion

1. In contrast to the traditional crowdsourcing algorithms which focus on either classification or regression problem, we propose a Bayesian model for aggregating *structured time series* data.
2. We exploit the problem structure to develop an efficient conditional particle filter algorithm for model inference.
3. We apply the proposed algorithm on ensemble of the output of different trackers in object tracking problem. The results could beat state-of-the-art single tracker by a large margin.

Conclusion

1. In contrast to the traditional crowdsourcing algorithms which focus on either classification or regression problem, we propose a Bayesian model for aggregating *structured time series* data.
2. We exploit the problem structure to develop an efficient conditional particle filter algorithm for model inference.
3. We apply the proposed algorithm on ensemble of the output of different trackers in object tracking problem. The results could beat state-of-the-art single tracker by a large margin.

Conclusion

1. In contrast to the traditional crowdsourcing algorithms which focus on either classification or regression problem, we propose a Bayesian model for aggregating *structured time series* data.
2. We exploit the problem structure to develop an efficient conditional particle filter algorithm for model inference.
3. We apply the proposed algorithm on ensemble of the output of different trackers in object tracking problem. The results could beat state-of-the-art single tracker by a large margin.

End

Project Page (including paper, supplemental material, results and codes): `http://winsty.net/ebt.html`