

Research on weighted naive Bayesian classifier in discriminative tracking

Yan Wang

Graduate School
Shenyang Ligong University
Shenyang, China
e-mail: wy2228@sohu.com

Guangsan Li

Bank Insurance Department
Cinosoft Company Limited
Beijing, China
lixiaoye2011@163.com

Dawei Yang

School of Information Science and Engineering
Shenyang Ligong University
Shenyang, China
e-mail: ydwl128@126.com

Abstract—Aiming at the question that each positive sample in trained sample set has the same weight when training the naive Bayesian classifier in discriminative tracking algorithm, an algorithm is proposed to put different weight on different positive sample according to the distances between the positive samples and the object, the distance is smaller, the similarity between them is bigger, the weight on this positive sample should be bigger, otherwise, the weight on this sample should be smaller. Taking the distribution of positive samples into consideration, and using the distance between the positive sample and the object to measure the weight of this positive sample, and put different weights on different positive samples. The results of experiments show that this change can prompt the performance of the classifier, and further improve the robustness of the tracking algorithm.

Keywords- *discriminative tracking; naive bayesian classifier; Gaussian function; object tracking; trained sample*

I. INTRODUCTION

During visual object tracking, when falling across some things, such as strong illumination change, partial occlusion, which may cause object appearance change obviously, tracking will account drift problem, and tracking may fail [1]. In order to solve this problem, researchers propose adaptive appearance model, which may change adaptively as tracking going. Black et al. [2] have proposed to learn a subspace model to represent object offline. IVT [3] tracking uses an incremental subspace model to automatically adapt the change of object appearance. L1 [4] tracker uses a sparse combination of object template and patch template to represent object appearance. These methods have made some success on solving the problem of change of object appearance, but have three shortcuts: Firstly, these methods suppose object appearance would not change obviously during tracking period; Secondly, when sampling some samples around current object location, the appearance model need to adapt misaligned samples, this may cause drift problem; Third, these methods have not used background information.

In order to use background information in visual object tracking, there has been proposed a new class of visual object tracking algorithm called discriminative tracking algorithm, which treats tracking problem as a binary classification problem. Avidan [5] used support vector machine to classify samples in optical flow based discriminative tracking algorithm and have good gains. Grabner et al. [6] use online boosting algorithm to classify samples in discriminative tracking. These algorithms use current object location as positive samples, and use one positive sample and some negative samples to update the classifier, when appearance model updated adaptively, noise and misaligned instance may occur, this type of method using one positive sample may account drift problem. Babenko et al. [7] use multiple instances learning (MIL) algorithm to track the object in discriminative tracking. Zhang et al. [8] use weighted multiple instances learning (WMIL) algorithm to track the object, and later Zhang et al. [9] propose compressive tracking (CT) algorithm. The classifier in MIL algorithm, WMIL algorithm, and CT algorithm is naive Bayesian classifier, WMIL algorithm makes some improvement on MIL algorithm, and CT algorithm uses just the same classifier. These discriminative tracking algorithms put same weight on the positive sample when training naive Bayesian classifier, while not taking the distance between the positive sample and the object into consideration. The distance between the positive sample and the object is bigger, the similarity between them is smaller, the weight on this positive sample should be smaller. Otherwise, the weight on this positive sample should be bigger. In this paper, a weighted positive sample naive Bayesian Classifier is proposed.

II. THE IMPROVEMENT OF THE WEIGHT ON POSITIVE SAMPLE IN NAIVE BAYESIAN CLASSIFIER

The naive Bayesian classifier bases on Bayesian theory, and the term naive means that this Bayesian classifier supposes the feature value of samples are reciprocally independent. This suppose usually is not right in real

applications, but in 2004 Harry Zhang [10] had proved that the naive Bayesian classifier has good performance of sorting, even in the case of violating this suppose.

In CT algorithm, positive samples and negative samples need to be sampled to train the naive Bayesian classifier, showed as Fig. 1. Samples some positive samples in a circular region around the object, and the sampling formula just as:

$$D^{\alpha} = \{z \mid \|L(z) - L_t\| < \alpha\} \quad (1)$$

where, α means the radius of this circular, D^{α} means the sampled set of positive samples, which includes some positive samples, L_t means the location of the object in t-th frame, and $L(z)$ means the locations of positive samples. Samples some negative samples in a cirque region around the object, and the sampling formula just as:

$$D^{\zeta, \beta} = \{z \mid \zeta < \|L(z) - L_t\| < \beta\} \quad (2)$$

Where, ζ means the inner radius of this cirque, β means the outer radius of this cirque, $D^{\zeta, \beta}$ means the set of sampled negative samples, which includes some negative samples, L_t means the location of the object in t-th frame, and $L(z)$ means the locations of negative samples.

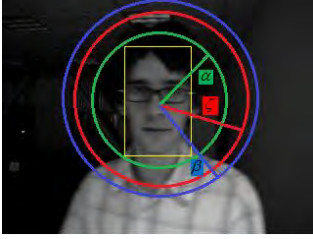


Figure 1. the sampling process of positive samples and negative samples

When getting positive samples and negative samples, uses a random sparse matrix according with restricted isometric property to distill generalized Haar-like feature from these samples, then uses integral histogram to calculate the value of generalized Haar-like feature, treats this value as the feature value of this sample, and last calculate the parameters of Gaussian distribution: means and standard deviation.

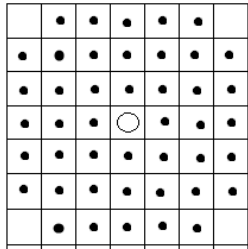


Figure 2. the distribution of positive samples

Supposing the number of positive samples in the set of positive samples is p , we use dense sampling method to get samples. Fig. 2 shows some positive samples on David video in one experiment. The rectangular including one black point means the pixel of left-top corner in the positive sample, while the rectangular including hollow

circular stands for the pixel of left-top corner in the current object, and treating this object as one positive sample when training the naive Bayesian classifier. Without taking the distance between the positive sample and the object into consideration, supposing the generalized Haar-like feature value of i -th positive samples in positive sample set D^{α} is $haar_i$, and $1 \leq i \leq p$, then the formula of the means of positive sample just as:

$$\mu_{pos} = \frac{1}{p} \sum_{i=1}^p haar_i \quad (3)$$

Where, μ_{pos} means the means of positive sample set D^{α} in naive Bayesian classifier, p is the number of positive samples in positive sample set, $haar_i$ is generalized Haar-like feature value of i -th positive samples. The standard deviation formula of the positive sample just as:

$$\sigma_{pos} = \sqrt{\frac{1}{p} \sum_{i=1}^p (haar_i - \mu_{pos})^2} \quad (4)$$

Where, σ_{pos} means the standard deviation of positive sample set D^{α} in naive Bayesian classifier. The 3-th formula and the 4-th formula are also the formula of calculating the means and standard deviation of Gaussian function on the positive sample in CT algorithm.

Supposing the location of the object is (x_0, y_0) , and the location of the i -th positive sample in positive sample set D^{α} is (x_i, y_i) . Using chessboard distance formula to calculate the distance between the positive sample and the object.

$$d_i = \max(\|x_i - x_0\|, \|y_i - y_0\|) \quad (5)$$

Where, d_i means the chessboard distance between the object and the i -th positive sample in positive sample set D^{α} , x_0 is the horizontal coordinate of the pixel of left-top corner in object location, y_0 is the vertical coordinate of the pixel of left-top corner in object location, x_i is the horizontal coordinate of the pixel of left-top corner of the i -th positive sample in positive sample set D^{α} , y_i is the pixel's vertical coordinate of left-top corner of the i -th positive sample in positive sample set D^{α} .

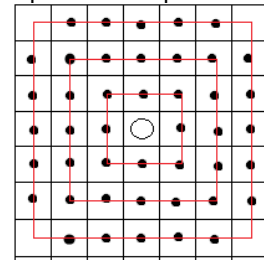


Figure 3. the chessboard

We can get the distribution of the positive samples after getting the distance between the positive sample and the object, just as Fig. 3. There are three squares around the object, the first square contains eight positive samples, the second square contains sixteen positive samples, the third square contains twenty positive samples. The distances between the positive samples and the object is unequal, the distance of the positive samples on the first square is least, the distance of the positive samples on the third square is most. The distance between the positive sample and the object smaller, the similarity between them is bigger. The distance between the positive sample and the object bigger, the similarity between them is smaller. Taking the distance between the positive sample and the object into consideration, translates the distance element into weight element, and uses the weight element to calculate the means. Uses (6) to translate the chessboard distance d_i , which between the i -th positive sample in positive sample set D^α and the object, into w_i , which means the weight of this positive sample:

$$w_i = \frac{1}{c} e^{-(d_i+1)} \quad \text{s.t.} \quad \sum_{i=1}^p w_i = 1 \quad (6)$$

Where, w_i is the weight of the i -th positive sample in positive sample set D^α , p is the number of positive samples in positive sample set D^α , c is normalized element. The formula of weighted positive sample means just as:

$$\mu_{wpos} = \sum_{i=1}^p haar_i w_i \quad (7)$$

Where, μ_{wpos} is the means of weighted positive sample set. The formula of weighted positive sample standard deviation just as:

$$\sigma_{wpos} = \sqrt{\frac{1}{p} \sum_{i=1}^p (haar_i - \mu_{wpos})^2} \quad (8)$$

Where, σ_{wpos} is the standard deviation of weighted positive sample set. The (7) and (8) are the formulas to calculate the means and standard deviation of Gaussian function on positive sample.

When training the naive Bayesian classifier, negative samples are sampled from background, so it is not need to take the distance between the negative sample and the object into consideration. The means formula of negative samples just as:

$$\mu_{neg} = \frac{1}{n} \sum_{i=1}^n haar_i \quad (9)$$

Where, μ_{neg} is the means of negative samples, n is the number of negative samples, $haar_i$ is the generalized Haar-like feature value of i -th negative samples, $1 \leq i \leq n$. The standard deviation formula of negative samples just as:

$$\sigma_{neg} = \sqrt{\frac{1}{n} \sum_{i=1}^n (haar_i - \mu_{neg})^2} \quad (10)$$

Where, σ_{neg} means the standard deviation of negative samples. The (9) and (10) are same as those formulas in CT algorithm, which are used to calculate the means and the standard deviation of Gaussian function on negative samples.

We can train the naive Bayesian classifier after getting the means and the standard deviation on positive samples and negative samples. Uses s to stand for one sample, $haar_s$ is the feature value of this sample. To decide this sample is positive or negative, the positive probability and the negative probability are needed to be calculated separately. The positive probability of this sample just as:

$$p(haar_s | pos) = \frac{1}{\sqrt{2\pi\sigma_{wpos}^2}} \exp\left(-\frac{(haar_s - \mu_{wpos})^2}{2\sigma_{wpos}^2}\right) \quad (11)$$

Where, $p(haar_s | pos)$ is the probability which stands for this sample is positive. The negative probability of this sample just as:

$$p(haar_s | neg) = \frac{1}{\sqrt{2\pi\sigma_{neg}^2}} \exp\left(-\frac{(haar_s - \mu_{neg})^2}{2\sigma_{neg}^2}\right) \quad (12)$$

Where, $p(haar_s | neg)$ is the probability which stands for this sample is negative. The naive Bayesian classifier uses (13) to decide this sample is negative or positive in a sense of probability:

$$p(haar_s) = \frac{p(haar_s | pos)}{p(haar_s | neg)} \quad (13)$$

Where, if $p(haar_s)$ is bigger than 1, and then this sample is likely to be positive. If $p(haar_s)$ is smaller than 1, and then this sample is likely to be negative.

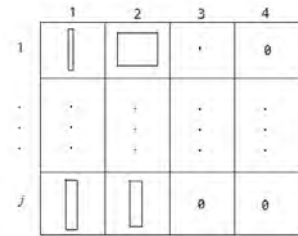


Figure 4. The generalized Haar-like feature of one sample

When getting the generalized Haar-like feature from positive samples and negative samples, we pick up j times, not only once. Fig. 4 shows that we pick up some generalized Haar-like features from one sample. The horizontal numeric one, two, three, four stands for the number of generalized Haar-like feature every time, the rectangular in the grid according with numeric one stands for the shape of the first generalized Haar-like feature, just as numeric one, the numeric zero in the grid show that we have not pick up generalized Haar-like feature from samples. The vertical numeric, from zero to j , stands for how many times we pick up generalized Haar-like feature,

the numeric one stands for we firstly pick up the generalized Haar-like feature from one sample, apostrophe stands for we go on picking up the generalized Haar-like feature from this sample, the numeric j stands for we pick up the generalized Haar-like feature the j -th times. Such and such, because pick up features for j times, after summing up the feature of every time, the number of feature is still j . We use a random sparse matrix according with Gaussian distribution when picking up these features, so these features distribute independently. We can calculate the density function of every feature and then get their product. If the number of used features is j , the probability formula that decides one sample is positive just as:

$$p(s | pos) = \prod_{i=1}^j p(haar_{s,i} | pos) \quad (14)$$

where, $p(s | pos)$ is the probability that s is positive, $haar_{s,i}$ is the generalized Haar-like feature from sample s at i -th times, $1 \leq i \leq j$, $p(haar_{s,i} | pos)$ is the probability that samples s is positive according with the features at i -th time. The probability formula that decides sample s is negative just as:

$$p(s | neg) = \prod_{i=1}^j p(haar_{s,i} | neg) \quad (15)$$

Where, $p(s | neg)$ is the probability that sample s is negative, $p(haar_{s,i} | neg)$ is the probability that samples s is negative according with the features at i -th time. The naive Bayesian Classifier use (16) to decide sample s is positive or negative in a dense of probability.

$$p(s) = \log \left(\frac{\prod_{i=1}^j p(haar_{s,i} | pos)}{\prod_{i=1}^j p(haar_{s,i} | neg)} \right) \quad (16)$$

$$= \log \left(\frac{p(haar_{s,1} | pos)}{p(haar_{s,1} | neg)} \right) + L + \log \left(\frac{p(haar_{s,j} | pos)}{p(haar_{s,j} | neg)} \right)$$

Where, $p(s)$ shows that sample s is positive to what extent. If $p(s)$ is bigger than one, it shows that sample s is likely to be positive, otherwise, likely to be negative.

When sampling detection samples from a circular region around the location of object in $t+1$ th frame, uses following formula:

$$D^\gamma = \{z | \|L(z) - L_t\| < \gamma\} \quad (17)$$

Where, γ stands for the radius of circular region, D^γ is the detection sample set, L_t is the location of object in t -th frame, $L(z)$ is the location of the detection samples, just as Fig. 5.

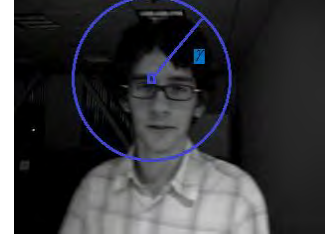


Figure 5. the sampled detection sample

When sampling detection samples, the number of detection samples is d , calculates the i -th detection sample's probability $p(sample_i)$, finds the biggest probability, and treats this detection sample as the location of new object, the formula just as:

$$L_{t+1} = L_i \text{ s.t. } \max_i p(sample_i) \quad (18)$$

Where, L_{t+1} is the location of the object in $t+1$ th frame, L_i is the location of the i -th detection sample in $t+1$ th frame, $1 \leq i \leq d$, d is the number of detection sample, $p(sample_i)$ is the probability of the i -th detection sample, so the location of the object in $t+1$ th frame is decided.

The algorithm in this paper just as:

Input: the t -th frame in the video sequence, the location of the object L_t

Output: The location of the object L_{t+1} , the score of the classifier.

Step 1: We have get the location of the object L_t in the t -th frame in the video sequence. L_t contains four elements: horizontal coordinate of pixel of left-top corner in the window, vertical coordinate of pixel of left-top corner in the window, the width of the window, the height of the window.

Step 2: Samples some positive samples based on (1), calculates the means and the standard deviation of weighted positive samples based on (7) and (8).

Step 3: Samples some negative samples based on (2), calculates the means and the standard deviation of negative samples based on (9) and (10).

Step 4: In $t+1$ th frame of video sequences, samples some detection samples based on (17).

Step 5: Uses the means and the standard deviation of positive samples and negative samples from step 2 and step 3, constructs Gaussian function based on (11) and (12).

Step 6: According to the naive Bayesian rule, calculates the probability of each detection samples based on (16).

Step 7: Treats the detection sample whose score values is biggest as the location of the object in $t+1$ -th frame based on (18).

Step 8: In the $t+1$ th frame of video sequences, repeats step 2 and step 3, updates parameters of the classifier based on (19).

$$\begin{aligned}\mu_{t+1} &= \phi\mu_{t+1} + (1-\phi)\mu_t \\ \sigma_{t+1} &= \sqrt{\phi(\sigma_{t+1})^2 + (1-\phi)(\sigma_t)^2 + \phi(1-\phi)(\mu_{t+1} - \mu_t)^2}\end{aligned}\quad (19)$$

Where, ϕ is a learning element, stands for the updating method of the appearance model.

Step 9: Repeats from step 1 to step 8.

III. EXPERIMENT RESULTS AND ANALYSIS

The hardware environment in these experiments just as: The CPU is i5-3210M, the memory is 4G. The software environment in these experiments just as: The operating system is windows8, the IDE is matlabR010b. In these experiments we make some comparison between CT tracker and our algorithms, the part results on David video sequence and sylv video sequence based on these two algorithms in Fig. 6 and Fig. 7 respectively. Solid rectangular is the result of CT algorithm, dashed rectangular is the result of our algorithm. The tracking error curve on David video sequence and sylv video sequence in Fig. 8, the solid stands for CT algorithm, dotted line stands for our algorithm.

When training the naive Bayesian classifier in CT algorithm, without taking the case that the distances between different positive samples and the object are different into consideration, and putting the same weight on different positive samples, so all positive samples have played the same role on determine one sample is positive or negative. The algorithm in this paper takes the fact that the distances between different positive samples and the object are different into consideration, and according with this inequality, putting different weight on different positive samples, makes these positive samples play different role on determining one detection sample is positive or negative. From the 49-th frame, the 83-th frame, the 159-th frame, the 203-th frame, the 275-th frame, the 324-th frame in David video sequence, and the 60-th frame, the 147-th frame, the 230-th frame, the 310-th frame, the 375-th frame, the 445-th frame in sylv video sequences, we can find that when putting the weight on the positive samples according with the distance between the positive sample and the object, the performance of the classifier is better, the robustness of the algorithm is improved.

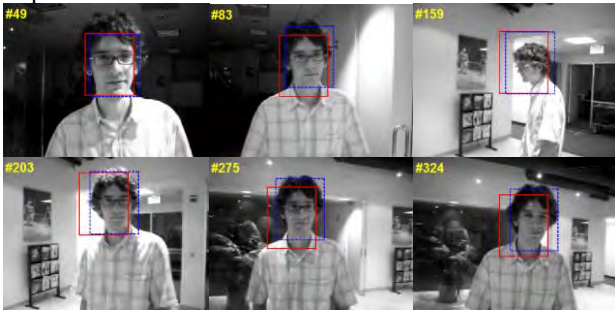


Figure 6. the tracking results on David video sequences



Figure 7. the tracking results on sylv video sequences

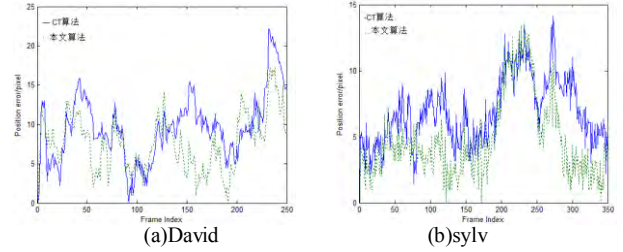


Figure 8. the tracking error curve

IV. CONCLUSIONS

Naive Bayesian Classifier is simple but effective, and it has been used in discriminative tracking algorithms. Some excellent discriminative tracking algorithms, such as MIL, WMIL, CT, which use naive Bayesian classifier, give the same weight to every positive sample. According to probability theory, the distance between positive sample and object larger, the similarity is smaller, so this sample should be put less weight. The distance is smaller, the similarity is bigger, and the weight is larger. Thinking of this, in this paper, when training naive Bayesian classifier, the distance between samples and object is considered, and according to this, different weight is given to relevant trained samples. Experiments show that this improvement can boost performance of naive Bayesian classifier and improve robustness of visual object tracking algorithm.

ACKNOWLEDGMENT

This research was supported in part by Shenyang Technological Innovation Special Fund (F13-316-1-73).

REFERENCES

- [1] Tsagkatakis G., Savakis A. Online Distance Metric Learning for Object Tracking. IEEE Transaction on Circuits and Systems for Video Technology, 2011, 21(12), 1810~1821.
- [2] Black, M., Jepson, A.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision 38 (1998) 63-84
- [3] Ross, D., Lim, J., Lin, R., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision 77 (2008) 125-141
- [4] MEI, X., LING, H. 2009. Robust visual tracking using L1 minimization. IEEE International Conference on Computer Vision, 1436-1443.

- [5] Avidan, S.: Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004) 1064-1072
- [6] Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via online boosting. In *British Machine Vision Conference* (2006) 47-56
- [7] Babenko, B., Yang, M. H., Belongie, S.: Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011) 1619-1632
- [8] ZHANG, K., SONG, H. 2012. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*.
- [9] ZHANG, K., ZHANG, L., YANG, M. 2012. Real-time compressive tracking. *European Conference on Computer Vision*.
- [10] Harry Zhang. The optimality of naive bayes. 2004, Florida Artificial Intelligence Research Society Conference