FAST GAUSSIAN MIXTURE CLUSTERING FOR SKIN DETECTION

Zhiwen Yu and Hau-San Wong

Department of Computer Science
City University of Hong Kong

ABSTRACT

Support vector machine (SVM) is a hot topic in many areas, such as machine learning, computer vision, data mining, and so on, due to its powerful ability to perform classification. Though there exist a lot of approaches to improve the accuracy and the efficiency of the models of SVM, few of them address how to eliminate the redundant data from the input training vectors. As it is known, most of support vectors distributes in the boundary of the class, which means the vectors in the center of the class are useless. In the paper, we propose a new approach based on Gaussian model to preserve the training vectors in the boundary of the class and eliminate the training vectors in the center of the class. The experiments show that our approach can reduce most of the input training vectors and preserve the support vectors at the same time, which leads to a significant reduction in the computational cost and maintains the accuracy.

***Index Terms*—** Support vector machine, Image segmentation

1. INTRODUCTION

Recently, the researchers gain more and more attention to support vector machine (SVM) due to its useful applications in many areas [1]-[9], such as machine learning, neural network, data mining, multimedia, and so on. Given a two-class linearly separable task, basic SVM approach [1] finds a hyperplane which maximizes the geometric margin and minimizes the classification error. Though there exist a lot of SVM approaches, they can be divided into two categories based on the algebraic view [1]-[5] and the geometric view [6]-[9]: (i) the approaches from the algebraic view includes sequential minimal optimization (SMO) [3], SVM with soft margin [2], ν-SVM [5], kernel SVM, support vector regression machine, and so on. These approaches explore how to minimize the classification error and reduce the computational cost of SVM by the algebraic algorithms. (ii) the approaches from the geometric view includes SVM with dual representation [6], the iterative nearest point algorithm, SVM based on convex hull [8], SVM based on reduced convex hull (RCH) [7], and so on. These approaches make use of the geometric properties of SVM to solve the classification task.

Though the approaches of SVM in both categories consider all kinds of problems about SVM, most of them still ignore one problem: how to eliminate the redundant training vectors to make SVM more efficient and maintain the accuracy at the same time. As it is known, the most useful training vectors are support vectors, which form the support vector classifier and determine the hyperplane with the maximum margin, while the contribution of the other training vectors is limited. As a result, we design a new approach called fast support vector machine approach (SVM) based on the Gaussian model and the projection process to remove the redundant training vectors and preserve the support vectors.

The remainder of the paper is organized as follows. Section 2 introduces fast support vector machine approach (FSVM) and its performance analysis. Section 3 presents how to estimate k value which is an important factor in FSVM. Section 4 describes how to extend FSVM to solve multi-class problem. Section 5 applies our proposed approach on real-time image segmentation. Section 6 is the conclusion and future work.

2. FAST SUPPORT VECTOR MACHINE APPROACH

Given a set of training vectors $V_{train} = \{v_1, v_2, ..., v_n\}$ with the labels $Y_{train} = \{y_1, y_2, ..., y_n\}$ $(y_i \in \{1, 2\})$, the objecitive of fast support vector machine approach (FSVM) is to (i) eliminate the redundant training vectors and (ii) train the classifier by the remaining training vectors. The difference between FSVM and the existing SVM approaches is that FSVM focuses on reducing the redundant training vectors. There are two assumptions which relate to FSVM: (i) there exist a convex hull for the input training vectors in each class; (ii) the problem is separable. Figure 2 (a) illustrates an example which satisfies the assumptions and the classifier obtained by the traditional SVM is shown in Figure 2 (b).

Figure 1 shows the overview of FSVM. FSVM first eliminate the training vectors which are close to the center of the class by the Gaussian models. Then, it removes the training vectors by a projection process. Finally, FSVM performs SMO on the remaining training vectors to obtain the classifier.

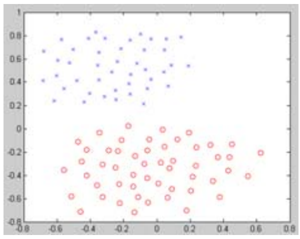
Algorithm **FSVM** (a set of training vectors V_{train})
1. Eliminate the training vectors by the Gaussian models;
2. Eliminate the training vectors by the projection process;
3. Perform SMO to obtain the binary SVM classifier;

Fig. 1. The overview of FSVM

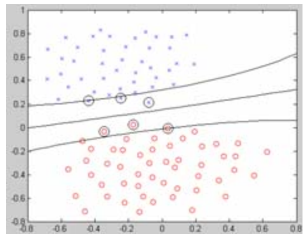
2.1. Eliminating by the Gaussian model

FSVM first estimate the multivariate Gaussian distribution of the input training vectors in each class:

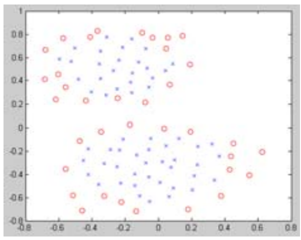
$$G = (\mu, \Sigma) \tag{1}$$



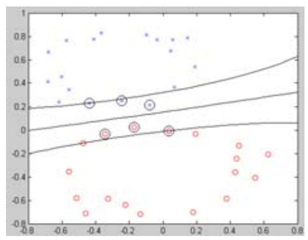
(a) The original training vectors



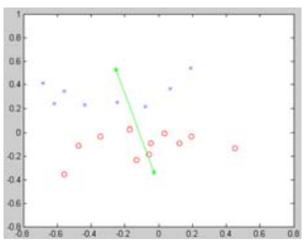
(b) The classifier obtained by FSVM(O)



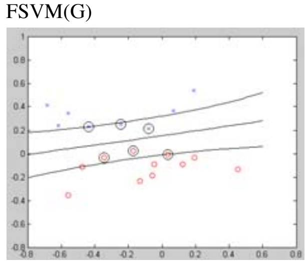
(c) The candidates generated by the Gaussian model



(d) The classifier obtained by FSVM(G)



(e) The candidates generated by the projection process



(f) The classifier obtained by FSVM(P)

Fig. 2. The example of fast support vector machine approach

$$\mu = \frac{\sum_{i=1}^{n} v_i}{n}, \quad \sigma^2 = \frac{\sum_{i=1}^{n} (v_i - \mu)^2}{n} \tag{2}$$

where μ is the mean of the Gaussian model G, Σ is the $d \times d$ diagonal covariance matrix with σ^2 on its diagonal, and d is the number of dimensions. The value of the input training vector v with respect to the multivariate Gaussian probability distribution function can be calculated by

$$P(v) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{|\Sigma|}} e^{(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu))} \tag{3}$$

One interesting observation for these probability values with respect to G is that the vectors which are close to the center of the Gaussian distribution have the large probability values, while the vectors which are close to the boundary have the small probability values. FSVM selects k training vectors in each class with the smallest probability values in the second step. Figure 2 (c) demonstrates an example of selecting $k = 18$ training vectors (red circles) from each class. Most of the selected vectors locate in the boundary of the class. The input vectors which are not selected will be removed from the training set as shown in Figure 2 (d).

2.2. Eliminating by the projection process

Another interesting observation for support vectors is that the support vectors of one class always locate in the place which are close to the other class as illustrated in Figure 2 (b) and Figure 2 (d). So, in the third step, FSVM further eliminates the redundant training vectors by a projection process as shown in Figure 2 (e) (f).

We formulate the process of the projection in the following. The training vectors can be divided into two classes I_{train} and J_{train}:

$$\begin{aligned} V_{train} &= I_{train} \cup J_{train} \\ I_{train} &= \{v_1, v_2, ..., v_I\} \\ J_{train} &= \{v_1, v_2, ..., v_J\} \end{aligned} \tag{4}$$

FSVM first considers the class I_{train}. It translates the origin of the coordinate system to the center μ_I of the class I_{train}:

$$v'_i = v_i - \mu_I, \; i \in [1, I] \tag{5}$$

Then, the vector $\mu_J \mu_I$ is obtained by the following equation:

$$\mu'_J = \mu_J - \mu_I \tag{6}$$

In the third step, FSVM project the vector μ'_J to all the vectors v'_i ($i \in [1, I]$) respectively as follows:

$$|\mu'_J| cos\theta_i = \frac{v'_i \cdot \mu'_J}{|v'_i|} \tag{7}$$

where θ_i is the angle between the input vector v'_i and the vector μ'_J. The following equation is obtained by substitute $(v_i - \mu_I)$ and $(\mu_J - \mu_I)$ for v'_i and μ'_J respectively:

$$|(\mu_J - \mu_I)| cos\theta_i = \frac{(v_i - \mu_I) \cdot (\mu_J - \mu_I)}{|v_i - \mu_I|} \tag{8}$$

$$\delta(|(\mu_J - \mu_I)| cos\theta_i) = \begin{cases} 1 & \text{if } |(\mu_J - \mu_I)| cos\theta_i \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{9}$$

If $\delta(|(v_i - \mu_I)| cos\theta_i) = 1$, the training vector will be preserved. Figure 3 (a) illustrate an example of the projection. The training vector v_2 in Figure 3 (a) will be preserved, since $cos\theta_2 > 0$.

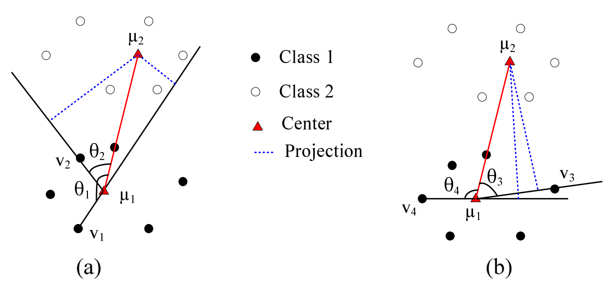


Fig. 3. The projection process

FSVM also preserves two training vectors $v_{i_1^*}$ and $v_{i_2^*}$ which satisfy one of the following conditions:

$$\begin{aligned} i_1^* &= arg\ min_{i \in [1,I] \&\& cos\ \theta_i \geq 0} \ \ cos\ \theta_i \\ i_2^* &= arg\ min_{i \in [1,I] \&\& cos\ \theta_i < 0} \ \ - cos\ \theta_i \end{aligned} \tag{10}$$

The training vector v_3 in Figure 3 (b) will be preserved since v_3 satisfies the first condition, while The training vector v_4 in Figure 3 (b) is preserved due to satisfy the second condition.

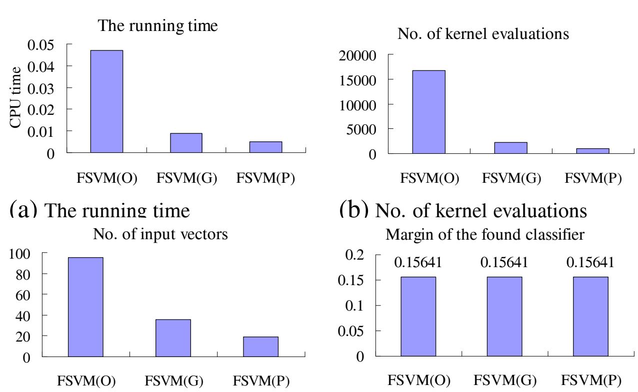
In the fourth step, FSVM eliminates all the training vectors which satisfy $\delta(|(v_i - \mu_I)|cos\theta_i) = 0$ and do not satisfy one of the about conditions. The training vector v_1 in Figure 3 (b) is removed from the training set.

By the same approach, part of the training vectors in J_{train} can be eliminated from the training set. **Note that**, the projection process can be used to remove the redundant training vectors for non-separable problem.

2.3. Training the classifier

Finally, FSVM performs sequential minimal optimization (SMO) to compute the discriminant function ($f(v) = \omega^T v + b$ for linear case and $f(v) = \langle \alpha \cdot K(v, s)\rangle + b$ for kernel case [10]) of binary SVM classifier on the remaining training vectors V_{remain} in the fourth step (where $K(v, s) = [k(v, s_1), k(v, s_2), ..., k(v, s_m)]^T$, $s \in S = \{s_1, s_2, ..., s_m\}$ is the support vector which is a subset of the input training vectors, m is the number of support vectors, and $k(v, s_j)(j \in [1, m])$ is the evaluation of kernel function centered at s_j).

Figure 2 illustrates an example of the FSVM algorithm on the synthetic dataset. Figure 2 (a) shows the original input training vectors, Figure 2 (c) illustrates the training vectors after pruning by the Gaussian model, and Figure 2 (e) demonstrates the training vectors after pruning by the projection process. The corresponding binary classifiers obtained by FSVM are shown in Figure 2 (b) (d) (f) respectively. Though the classifiers are obtained from different training sets with different number of the training vectors, the classifiers in Figure 2 (b) (d) (f) are same due to the same support vectors and the same margin as shown in Figure 4 (d). The pruning processes by the Gaussian model and the projection remove most of the training vectors as illustrated in Figure 4 (c), which lead to significantly reducing in the computational cost as demonstrated in Figure 4 (a) and the number of kernel evaluations as shown in Figure 4 (b) (where $FSVM(O)$, $FSVM(G)$ and $FSVM(P)$ correspond to the above three classifiers).



(a) The running time

(b) No. of kernel evaluations

(c) No. of input vectors

(d) The margins of the found classifiers

Fig. 4. The performance of the approaches

3. K VALUE ESTIMATION

The parameter k is the most important factor for FSVM. In order to estimate k value, FSVM has to evaluate the number of the training vectors on the boundary of the class. The algorithm first calculates the average probability value $\overline{P}$ for the training vectors in the first class I_{train} according to the following equation :

$$\overline{P} = \frac{\sum_{i=1}^{|I_{train}|} P(v_i)}{|I_{train}|} \tag{11}$$

where I_{train} is the cardinality of the first class.

Then, it computes the minimum average probability value $\overline{P}_{min}$ and the maximum average probability value $\overline{P}_{max}$:

$$\overline{P}_{min} = \frac{\sum_{j=1}^{|I_{min}|} P(v_j)}{|I_{min}|}, \ \overline{P}_{max} = \frac{\sum_{h=1}^{|I_{max}|} P(v_h)}{|I_{max}|} \tag{12}$$

$$\begin{aligned} v_j \in I_{min} &= \{v_i | P(v_i) < \overline{P}, v_i \in I_{train}\} \\ v_h \in I_{max} &= \{v_i | P(v_i) \geq \overline{P}, v_i \in I_{train}\} \end{aligned} \tag{13}$$

We further defines the distribution ratio (R) as follows:

$$R = \frac{\overline{P}_{min}}{\overline{P}_{max}} \tag{14}$$

If most of training vectors locates on the boundary of the class, $\overline{P}_{min}$ is close to $\overline{P}_{max}$ and R is close to 1. If R is small, most of the training vectors are close to the center of the class.

In the third step, the estimation algorithm evaluates k value based on the distribution ratio (R). If $R \geq \tau$ (τ is a threshold and be set to 0.4 in the paper), k is equal to the number of the training vectors in the first class. Otherwise, k is estimated by the following equation:

$$k = \lceil \frac{\overline{P} - \overline{P}_{min}}{\overline{P}_{max} - \overline{P}_{min}} \cdot |I_{train}| \rceil \tag{15}$$

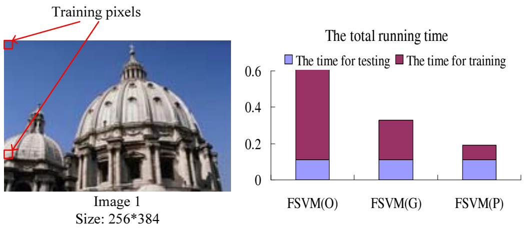
Where $|I_{train}|$ denotes the cardinality of the first class I_{train}. By the same approach, we can estimate k value for the second class.

4. EXPERIMENTS

In the experiment, we compare three approaches: FSVM(O) which performs SMO on the original training dataset, FSVM(G) which performs SMO on the training dataset after pruning by the Gaussian model and FSVM(P) which performs SMO on the training dataset after pruning by the projection process. The default setting of the threshold τ for k value estimation is 0.4.

The real training dataset consists of a set of training vectors with R, G, B value. The pixels in the red rectangle in Figure 5 (a) are the training vectors, which work as the prior knowledge for image segmentation. The remaining pixels are viewed as the test dataset. Our objective is performing image segmentation based on the SVM classifier training by the prior knowledge.

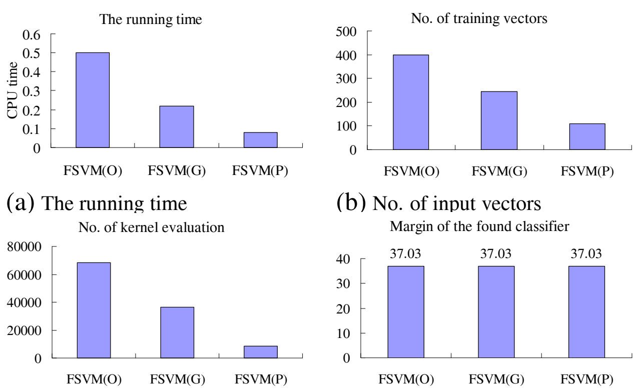
Figure 6 shows the performance of three approaches on image 1 and image 2 respectively. FSVM(P) outperforms its competitions as shown in Figure 6 (a), due to its powerful ability to reduce the number of the input training vectors as shown in Figure 6 (b). The corresponding number of kernel evaluations decreases too,



(a) The training pixels (b) The total running time

Fig. 5. The training pixels and the total running time

which is illustrated in Figure 6 (c). Fortunately, the classifier obtained by FSVM(P) has the same margin of the classifiers obtained by FSVM(O) and FSVM(G). They have the same support vectors (the number of support vectors is 9 and 6 for image 1 and image 2 respectively), too.



(a) The running time (b) No. of input vectors (c) No. of kernel evaluations (d) The margins of the found classifiers

Fig. 6. The performance of three approaches

Figure 7 illustrates the segmentation results by three classifiers which are obtained by FSVM(O), FSVM(G) and FSVM(P) on image 1 and image 2 respectively. The segmentation results on the test dataset shown in Figure 7 are indistinguishable, while FSVM(P) takes the lower computational cost by comparing with other two approaches, as shown in Figure 5 (b).

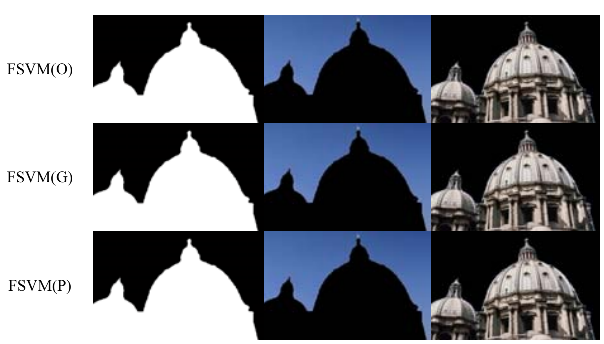


Fig. 7. Segmentation results by three approaches

5. CONCLUSION AND FUTURE WORK

This paper investigates the problem of eliminating the redundant data from the input training vectors. Though there exist a large number of algorithms to improve SVM, few of them consider how to reduce the number of the training vectors and maintain the accuracy at the same time. Our major contribution is a new approach based on the Gaussian model and the projection process to eliminate the redundant training vectors. We further propose an algorithm to estimate the number of training vectors (k value) on the boundary of the classes . Finally, the experiments on the synthetic dataset and the real dataset demonstrates that the new approach can reduce the computational cost greatly and maintains the accuracy of the classifier at the same time. In the future, we will explore how to extend our approach to solve nonseparable problems by FSVM.

Acknowledgments

The work described in this paper was fully supported by a grant from City University of Hong Kong [Project No. 7001596].

6. REFERENCES

[1] Vapnik V.N. Statistical Learning Theory. John Wiley & Sons, 1998.

[2] Corinna Cortes, Vladimir Vapnik: Support-Vector Networks. Machine Learning 20(3), 1995, pp. 273-297.

[3] John C. Platt. Using Analytic QP and Sparseness to Speed Training of Support Vector Machines. NIPS 1998, pp. 557-563.

[4] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, John C. Platt. Support Vector Method for Novelty Detection, NIPS 1999, pp. 582-588.

[5] Chang C.C., Lin C.J., Training ν-support vector classifiers: Theory and algorithms. Neural computation Vol. 13(9), 2001, pp. 2119-2147.

[6] Michael E. Mavroforakis, Sergios Theodoridis. A geometric approach to support vector machine (SVM) classification. IEEE Transactions on neural networks, Vol. 17, No. 3, 2006, pp. 671-682.

[7] Kristin P. Bennett, Erin J. Bredensteiner. Duality and Geometry in SVM Classifiers. ICML 2000, pp. 57-64

[8] D. J. Crisp and C. J. C. Burges, A geometric interpretation of ν-SVM classifiers, Adv. Neural Inform. Process. Syst. (NIPS), 1999, pp.244C250.

[9] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, Dept. CSA, IISc, Bangalore, Karnataka, India, Tech. Rep. TR-ISL-99-03, 1999.

[10] M.I. Schlesinger and V. Hlaváč. Ten lectures on statistical and structural pattern recognition. Kluwer Academic Publishers, 2002.