Soft SVM and Its Application to Video Object Extraction

Yi Liu and Yuan F. Zheng
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, Ohio 43210
Email:{liuyi, zheng}@ece.osu.edu

***Abstract*— Support Vector Machines (SVM) is a state-of-the-art learning machine and has found a great deal of success in a wide range of applications. In the framework of SVM, each sample belongs to either one class or the other. This requirement, however, makes it difficult to apply SVM to the applications where the data exhibit partial or unclear class memberships. To address this problem, this paper reformulates the standard SVM to be a new learning machine that is capable of dealing with binary (or hard) as well as real-valued (or soft) class memberships. The new machine, which is named as Soft SVM (S_SVM), has been integrated into a classification-based video object extraction approach, and the experimental results demonstrate the effectiveness of the new approach.**

***Index Terms*— Support Vector Machines (SVM), fuzzy class memberships, soft SVM, video object extraction.**

I. Introduction

Support Vector Machines (SVM) is a state-of-the-art learning machine based on the *structural risk minimization* induction principle [1]. In recent years, SVM has been extensively used as a classification tool and has found a great deal of success in a wide range of applications including pattern recognition [2] [3], communications [4] [5], and image/video analysis [6] [7].

Despite its superior performance in solving many classification problems, SVM is yet limited to crisp classification scenario where each sample falls into either one class or the other without ambiguity. However, there are situations where the collected samples exhibit partial or unclear membership as they may belong to different classes by different degrees. As a matter of fact, ambiguous membership is a typical problem for a large number of applications such as climatic prediction [8], soil classification [9], remote sensing [10], ecological modeling [11] and etc., where soft classification can capture the fuzzy nature of the data better than hard classification. Unfortunately, SVM lacks this ability. To address this problem, Fuzzy SVM (FSVM) has been developed [12], which associates each labeled training sample with a fuzzy membership s_i and employ s_i to weigh the corresponding penalty term in the objective function. FSVM extends the horizon of SVM, but the information embedded in the fuzzy membership is missing when the corresponding sample is correctly classified because the penalty term is non-zero only when misclassification occurs. In this paper, we propose Soft SVM (S_SVM) which takes account of the real-valued membership no matter whether the samples are classified correctly or not. To test the effectiveness of S_SVM, we employ S_SVM as the classifier in a classification-based video object extraction approach. S_SVM shows its advantage over both standard SVM and FSVM by achieving higher classification accuracy.

The rest of the paper is organized as follows. First a brief introduction of SVM is given in Section II. Then the training mechanism of the S_SVM is reformulated in Section III. Section IV presents a classification-based approach for video object extraction where S_SVM is employed as the classifier. It also explains how to define the real-valued membership in this specific application. Experimental results are given in Section V which is followed by conclusions in Section VI.

II. Support Vector Machines

To facilitate the discussion, we only give a brief review of SVM in this section and refer the details to [13] [14]. Consider N training samples: $\{x_1, y_1\}, \ldots, \{x_N, y_N\}$, where $x_i = [\, x_{i,1}\ x_{i,2}\ \ldots\ x_{i,k}\,]^T$ is a k-dimensional feature vector representing the ith training sample, and $y_i \in \{-1, 1\}$ is the class label of x_i. A hyperplane in the feature space can be described by the equation $w^T x + b = 0$, where $w \in R^k$ and b is a scalar. The signed distance d_i from a point x_i in the feature space to the hyperplane is $d_i = \frac{w^T x_i + b}{\|w\|}$. When the training samples are linearly separable, SVM yields the optimal hyperplane that separates two classes with no training error, and maximizes the minimum value of $|d_i|$. It is easy to find that the parameter pair (w, b) corresponding to the optimal hyperplane is the solution to the following optimization problem:

$$\begin{aligned} \text{minimize}: &\quad L(w) = \tfrac{1}{2}\|w\|^2 \\ \text{subject to}: &\quad y_i\left(w^T x_i + b\right) \geq 1,\ i = 1, ..., N. \end{aligned} \qquad (1)$$

For linearly nonseparable cases, there is no such a hyperplane that is able to classify every training sample correctly. However the optimization idea can be generalized by introducing the concept of *soft margin*. The optimization problem thus becomes:

$$\begin{aligned} \text{minimize}: &\quad L(w, \xi_i) = \tfrac{1}{2}\|w\|^2 + C\textstyle\sum_{i=1}^N \xi_i \\ \text{subject to}: &\quad y_i\big(w^T x_i + b\big) \geq 1 - \xi_i, i = 1, \ldots N, \end{aligned} \qquad (2)$$

where ξ_i are called slack variables that are related to the soft margin, and C is the tuning parameter used to balance the margin and the training error. Both optimization problems (1) and (2) can be solved by introducing the Lagrange multipliers α_i that transforms them to a quadratic programming problem:

$$\begin{aligned} &\text{maximize:} && \textstyle\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ &\text{subject to:} && \textstyle\sum_{i=1}^{N} y_i \alpha_i = 0, \quad 0 \le \alpha_i \le C. \end{aligned} \tag{3}$$

III. Soft SVM

A. Introduction of Real-Valued Memberships

Again suppose we are given N training samples $\{(x_1, y_1), \dots, (x_N, y_N)\}$, where $x_i \in R^k$ are the input vectors and $y_i \in [-1\ 1]$ are their corresponding *real-valued* memberships. y_i can be considered as a slider moving between -1 and 1. The more it slides toward 1 (-1) the more degrees x_i exhibits to be a member of class 1 (-1), or the more certain we are about the fact x_i belongs to class 1 (-1). When the membership stays in the middle ($y_i = 0$), we have absolutely no idea which class the sample x_i comes from. In order to fit into the notations of standard SVM, we decompose y_i into two parameters as $y_i = \tilde{y}_i \lambda_i$, where $\tilde{y}_i = \text{Sign}(y_i)$ is the binary class label and $\lambda_i = |y_i|$ is the certainty measure.

B. Formulation of Soft SVM

In this section, we derives the detailed formulation of S_SVM. We start with the simple case of linearly separable sets, and then extend it to the linearly nonseparable case which is more general in reality.

1) ***Linearly separable case:*** Let us first consider a very simple example. Assume we have only two training samples x_1 and x_2. Also assume that x_1 comes from class 1 with full membership ($y_1 = 1$) while x_2 from class -1 with much less certainty, say, $y_2 = -0.2$. Considering that the uncertainties of the class labels are caused by the overlapping of the data near the separating boundary, the optimal hyperplane is expected to move toward the point x_2 to reflect the unbalanced memberships rather than stand right in the middle between x_1 and x_2 as yielded by the standard SVM. To do so, we relax the constraints as

$$\tilde{y}_i \big(w^T x_i + b\big) \ge \lambda_i \tag{4}$$

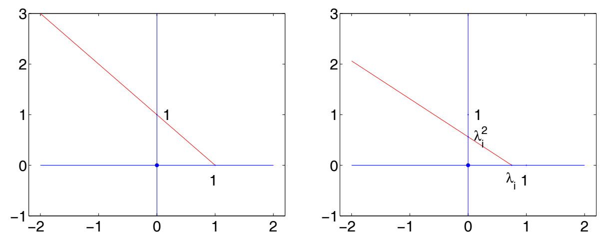
such that the samples with smaller λ_i stay closer to the decision hyperplane. As a result, the optimal hyperplane of S_SVM is the solution to the following optimization problem:

$$\begin{aligned} &\text{minimize :} && L(w) \;=\; \tfrac{1}{2}\,\|w\|^2 \\ &\text{subject to :} && \tilde{y}_i \left(w^T x_i \,+\, b \right) \ge\; \lambda_i,\; i = 1, ..., N. \end{aligned} \tag{5}$$

2) ***Linearly nonseparable case:*** In analogy to what SVM does to deal with the training error occurred in the linearly nonseparable case, we introduce the non-negative variables ξ_i, which satisfy

$$\tilde{y}_i \big(w^T x_i + b\big) \ge \lambda_i - \xi_i, \tag{6}$$

to penalize the objective function when the training samples x_i are misclassified. However, as mentioned before, the statements "x_i belongs to $\tilde{y}_i$" have different confidence levels measured by λ_i and we should worry more about the misclassification of the samples with higher λ_i. Thus in our S_SVM



(a) F_{SVM} function (b) $F_{\text{S\_SVM}}$ function

Fig. 1. The comparisons of F_{SVM} (SVM) and $F_{\text{S\_SVM}}$ (Soft SVM).

the error term ξ_i is further modified to $\lambda_i \xi_i$ to differentiate the penalty to the error, which yields the following formulation:

$$\begin{aligned} &\text{minimize :} && L(w, \xi_i) \;=\; \tfrac{1}{2}\,\|w\|^2 + C \textstyle\sum_{i=1}^{N} \lambda_i \xi_i, \\ &\text{subject to :} && \tilde{y}_i \big(w^T x_i + b\big) \ge \lambda_i - \xi_i. \end{aligned} \tag{7}$$

It is easy to find that the minimization problem of SVM which is described in (2) is equivalent to

$$\text{minimize :}\ \frac{1}{2}\|\omega\|^2 \;+\; C \sum_{i=1}^{N} F_{\text{SVM}}\big(y_i f(x_i)\big), \tag{8}$$

where $f(x_i) = w^T\, x_i + b$; $F_{\text{SVM}}(u) = 0$ if $u \ge 1$ and $1 - u$ if $u \le 1$, whose plot is displayed in Fig. 1(a). Similarly, the optimization function (7) is equivalent to:

$$\text{minimize :}\ \frac{1}{2}\|\omega\|^2 \;+\; C \sum_{i=1}^{N} F_{\text{S\_SVM}}\big(\lambda_i, \tilde{y}_i f(x_i)\big), \tag{9}$$

where $F_{\text{S\_SVM}}(u_1, u_2) = 0$ if $u_2 \ge u_1$ and $u_1(u_1 - u_2)$ if $u_2 < u_1$, as shown in Fig. 1(b).

As one can see from Fig. 1(a) and 1(b), all the training samples are treated equally when $\lambda_i = 1$ and the Soft SVM is reduced to the standard SVM. Another extreme is $\lambda_i = 0$. In that case, the penalty term $F_{\text{SSVM}}(0, u_2)$ would always be zero no matter it is classified as class 1 or -1. As a result the sample x_i would have no contribution to the decision boundary, which makes perfect sense since $\lambda_i = 0$ implies total uncertainty about which class x_i belongs to.

Similar to the standard SVM, the optimization problem of S_SVM can be transformed into the dual problem

$$\begin{aligned} &\text{maximize:} && \textstyle\sum_{i=1}^{N} \lambda_i \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j x_i^T x_j \\ &\text{subject to:} && \textstyle\sum_{i=1}^{N} \tilde{y}_i \alpha_i = 0, \quad 0 \le \alpha_i \le \lambda_i C. \end{aligned} \tag{10}$$

Then the optimal $\bar{w}$ of Eq. (7) is the linear combination of x_i as $\bar{w} = \sum_{i=1}^{N} \bar{\alpha}_i \tilde{y}_i x_i$, where $\bar{\alpha}_i$ denotes the optimal point of Eq. (10). As for the optimal b, it can be determined from the following Kuhn-Tucker conditions:

$$\bar{\alpha}_i \Big(\tilde{y}_i(\bar{w}^T x_i + \bar{b}) - \lambda_i + \bar{\xi}_i\Big) = 0, \quad i = 1, \dots, N, \tag{11}$$

$$(\lambda_i C - \bar{\alpha}_i)\bar{\xi}_i = 0, \quad i = 1, \dots, N. \tag{12}$$

From the derivation above, one can see that the dual problem of S_SVM is a quadratic problem similar to that of the standard SVM. The computational load of the new approach thus stays the same.

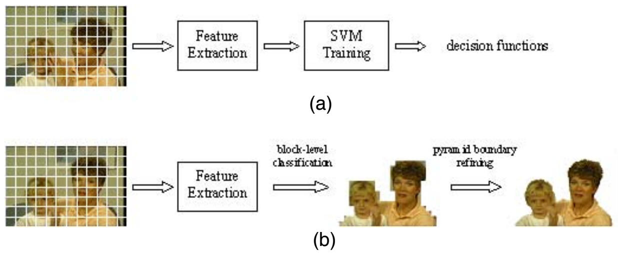


Fig. 2. An overview of the proposed approach. (a) The training phase. (b) The extraction phase.

IV. Video Object Extraction: an Application of Soft SVM

As a prerequisite of the emerging content-based video technologies, video object (VO) extraction is a very important yet challenging task. Recently the classification-based approaches which handle the VO extraction directly as a classification problem have been proposed [15]–[17]. For the remainder of this paper, we will integrate the proposed S_SVM into the VO extraction method described in [16] [17] and present the experimental results which demonstrate the effectiveness of this new learning machine.

A. Review of the VO Extraction Approach

The basic idea of [16] [17] is to decompose each frame into small blocks, and to use the learning machines such as SVM to classify them as foreground or background. The object of interest is then formed by all the foreground blocks. Fig. 2 presents an overview of the this scheme. As one can see, it consists of two phases: the training phase and the extraction phase. The training phase begins with dividing the first frame, chosen as the training frame, into blocks that are defined as *object blocks* or *background blocks* depending on which class the pixel at the center of the block belongs to. Every centering pixel as well as every block is represented by the local and neighboring features and through a learning procedure a decision function that separates the object and background is obtained. In the extraction phase, each subsequent frame is also divided into blocks, and for each block the decision function is evaluated to decide whether the centering pixel belongs to the object or not, which consequently determine the class label of the block. Then the tracking mask is formed by all the identified object blocks, at which point the resolution of object's boundary is as large as the size of the block. Finally, the pixel-wise accuracy is obtained by applying a *pyramid boundary refining algorithm* [16] [17] which refines the object boundary in an efficient and scalable manner.

B. Generating the real-valued membership

Recall that in the training phase the training frame is divided into blocks and each of them is labeled depending on which class the centering pixels belong to. It has been observed during the experiments that the labeling job is not as easy when the blocks lay around the object boundary which usually exhibits a gradual rather than a clean-cut transition. In other words, there are some uncertainties associated with these boundary blocks, i.e., they can not be fully assigned to either one of the two classes, where S_SVM comes to play an important role.

Now the question is how to generate the real-valued membership $y_i \in [-1\ 1]$ for a given block i. For this specific application, we utilize the normalized difference between the number of object and background pixels contained in the block as the measure. More specifically, for the $L \times M$ block size,

$$y_i = \frac{\#\text{ of object pixels} - \#\text{ of background pixels}}{L \times M}. \quad (13)$$

When the block locates fully inside (or outside) the object, we have $y_i = 1$ (or $y_i = -1$) showing no labeling ambiguity at all. When the block contains equal number of object and background pixels, $y_i = 0$ which indicates the maximum uncertainty. In other cases, y_i varies between -1 and 1.

V. Experimental Results

Experiments are conducted on some standard MPEG-4 test video sequences, and the performance is compared among SVM, the proposed S_SVM and the Fuzzy SVM proposed in [12] whose objective function is

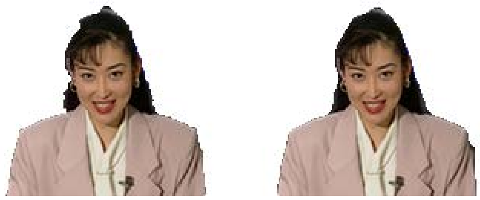
$$\begin{aligned} \text{minimize :} \quad & \tfrac{1}{2}\,\|w\|^2 + C\sum_{i=1}^{N}\lambda_i\xi_i, \\ \text{subject to :} \quad & \tilde{y}_i\big(w^T x_i + b\big) \geq 1 - \xi_i, \end{aligned} \quad (14)$$

where λ_i replaces the notation s_i used in [12].

Fig. 3 shows the tracking results of *Akiyo*, a typical head-and-shoulder type of sequence. Evidentally the object tracked by using S_SVM is more complete and accurate. The major difference lies around the boundary area of the object where SVM yields more misclassifications. Another test sequence is *Mom and Daughter*, which exhibits heterogenous spatial and motion characteristics in comparison with *Akiyo*. The Radial Basis Function (RBF) is employed as the kernel function, and again S_SVM yields more satisfactory results (Fig. 4).

To assess the performance quantitatively and objectively, we calculate the classification errors yielded by SVM, S_SVM and FSVM, which are plotted verses the number of frames as the dashed, dotted and solid lines respectively in Fig. 5. As one can see, the solid line is below the other two lines throughout the whole sequences showing that S_SVM achieves the highest classification accuracy. We want to point out that the absolute difference between the lines does not fully demonstrate how powerful S_SVM is considering the fact that SVM and FSVM have already delivered very low classification errors and not leaved much room for improvement. When we compute the relative difference, which is $\frac{E_{SVM}-E_{S\_SVM}}{E_{SVM}}$ and $\frac{E_{FSVM}-E_{S\_SVM}}{E_{FSVM}}$ where $E_{(\cdot)}$ denotes the classification errors yielded by the corresponding machine, S_SVM significantly outperforms SVM and FSVM by 31.3% and 11.0% in average with the biggest improvement 51.7% and 39.7%, respectively.

It also should be noted that for the *Mom and Daughter* sequence the dashed line and the dotted line coincide since FSVM and SVM yield the same decision function. This is not surprising because by using the RBF kernel, zero training error is attainable, and all the penalty terms ξ_i in Eq. (14) becomes

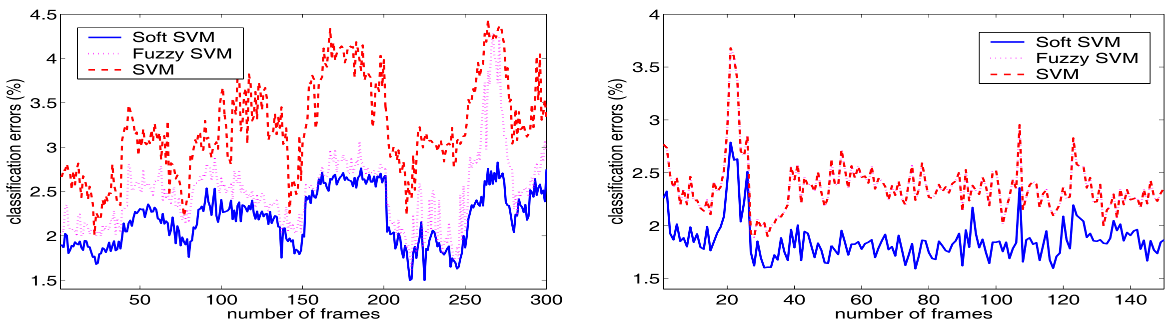


(a) (b)

Fig. 3. The tracking results of *Akiyo* using (a) SVM and (b) S_SVM.

(a) (b)

Fig. 4. The tracking results of *Mom and Daughter* using (a) SVM and (b) S_SVM.



(a) (b)

Fig. 5. The comparison of the classification accuracy among SVM, FSVM and S_SVM. (a) *Akiyo* sequence. (b) *Mom and Daughter* sequence.

zero which flattens the only effect of different λ_i. Nevertheless, FSVM delivers the second best performance among the three machines, which support our motivation that the fuzzy feature of data, if there is any, should be taken into consideration when the machine is trained.

VI. Conclusions

SVM is a powerful learning machine and has drawn a lot of attention in recent years. Under the framework of SVM, each training point belongs to one class or the other. However, in many classification problems, the collected samples do not exhibit a clean-cut membership. Instead they may belong to different classes by different degrees or different certainties. In this paper, we present S_SVM, a reformulated version of SVM which can deal with both binary and real-valued class memberships without increasing the computational cost. Classification-based video object extraction, a novel application, is also presented in this work as an example showing that S_SVM captures better the fuzzy nature of the gradual transition between two classes than the standard SVM.

References

[1] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support Vector Machines", *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18-28, July-Aug. 1998.

[2] M. Pontil and A. Verri, "Support Vector Machines for 3D Object Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637-646, June 1998.

[3] A. Tefas, C. Kotropoulos, and I. Pitas, "Using Support Vector Machines to Enhance the Performance of Elastic Graph Matching for Frontal Face Authentication", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 735-746, July 2001.

[4] D.J. Sebald and J.A. Bucklew, "Support Vector Machine Techniques for Nonlinear Equalization", *IEEE Tran. on Signal Processing*, vol. 48, no. 11, pp. 3217-3226, Nov. 2000.

[5] S. Chen, A.K. Samingan, and L. Hanzo, "Support Vector Mmachine Multiuser Receiver for DS-CDMA Signals in Multipath Channels", *IEEE Tran. on Neural Networks*, vol. 12, no. 3, pp. 604-611, May 2001.

[6] T.S. Huang, X.S. Zhou, M. Nakazato, Y. Wu, and I. Cohen "Learning in Content-Based Image Retrieval", *Proc. the 2nd International Conference on Development and Learning*, pp. 155-162, June 2002.

[7] G.D. Guo, A.K. Jain, W.Y. Ma, and H.J. Zhang, "Learning Similarity Measure for Natural Image Retrieval with Relevance Feedback", *IEEE Tran. on Neural Networks*, vol. 13, no. 4, pp. 811-820, July 2002.

[8] A.B. McBratney and A. W. Moore, "Application of Fuzzy Sets to Climatic Classification", *Agricultural and Forest Meteorology*, vol.35, pp. 85-165, 1985.

[9] A.B. McBratney and J.J. deGruijter, "A Continuum Approach to Soil Classification by Modified Fuzzy K-Means with Extragrades", *Journal of Soil Science*, vol. 43, pp. 79-159, 1992.

[10] A. del Amo, J. Montero, A. Fernandez, M. Lopez, J.M. Tordesillas and G. Biging, "Spectral Fuzzy Classification: An Application", *IEEE Tran on Systems, Man and Cybernetics, Part C*, vol. 32, no. 1, pp. 42-48, Feb. 2002.

[11] A. Salski, O. Franzle and P. Kandzia, "Fuzzy Logic in Ecological Modeling", *Ecological Moddeling*, vol. 85, 1995.

[12] C.F. Lin and S.D. Wang, "Fuzzy Support Vector Machines", *IEEE Tran. on Neural Networks*, vol. 13, no. 2, pp. 464-471, Mar. 2002.

[13] C. Cortes and V.N. Vapnik, "Support Vector Networks", *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[14] V.N. Vapnik, *The Natural of Statistical Learning Theory*, New York, NY: Springer-Verlag, 1995.

[15] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, "An Efficient Fully Unsupervised Video Object Segmentation Scheme Using an Adaptive Neural-Network Classifier Architecture", *IEEE Tran. on Neural Networks*, vol. 14, no. 3, pp. 616-630, May 2003.

[16] Y. Liu and Y.F. Zheng, "A GM-based Multi-Layer Method for Object Tracking in Video Sequences", *Proc. the 1st International Conference on Information Technology: Research and Education*, pp. 11-16, Aug. 2003.

[17] Y. Liu and Y.F. Zheng, "Video Object Segmentation and Tracking Using ψ_Learnin g Classification", accepted for publication in *IEEE Trans. on Circuits and Systems for Video Technology*.