Parallel and Sequential Support Vector Machines for Multi-label Classification

Liwei Wang, Ming Chang, and Jufu Feng

Center for Information Sciences, Peking University,
Beijing, 100871, China

{wanglw, changing, fjf}@cis.pku.edu.cn

Abstract

Multi-label classification is the problem that classes are not mutually exclusive, so that an example may belong to more than one category. This poses challenges to the traditional pattern recognition theory where class overlap means classification error. Multi-label classification arises typically in semantic scene classification, text categorization, medical diagnosis, and bioinformatics. However, only a small number of methods have been developed for multi-label classification. In this paper, we propose two algorithms, called Parallel Support Vector Machines (PSVMs) and Sequential Support Vector Machines (SSVMs), to handle multi-label classification problems. We applied them to scene classification. It is demonstrated that PSVM is comparable to, and SSVM outperforms the so-called cross-training C-criterion method.

Keyword: multi-label classification; scene classification; parallel SVM; sequential SVM; cross-training; c-criterion testing

I. Introduction

Multi-label classification has attracted attention in many areas [1-3]. It differs from traditional classification tasks in that the base classes are not mutually exclusive, so that an example may belong to more than one category. Such problems arise in semantic scene classification, text categorization, medical diagnosis, and bioinformatics. In scene classification, it is common that some images are relevant to multiple semantic categories. Fig. 1 shows an image that can be labeled by both *mountain* and *field*. Although in theory the multi-label problem may be reduced to single label problem by considering *mountain+field* as a new class, this method is impractical because the number of classes would increase tremendously and the data in the combined classes are often sparse. Recently, near infrared reflectance (NIR), Raman and Fourier transform infrared (FTIR) spectroscopy has been paid more attention to since they are fast, accurate and nondestructive techniques. For the raw spectral curves and their derivatives of herbal samples from different geographical origins including different provinces (in China) and different countries (Korea and China), through visible inspection it has been found that there exist some slight distinctions in the locations and relative intensities of peaks [1][2][3][4]. In order to identify geographical origins of Chinese medical herbs according to their spectral data effectively and automatically, some researchers have utilized several information processing techniques.

While numerous methods have been developed for conventional single label classification, only a few algorithms can handle multi-labeled data. In the document categorization domain, Schapire and Singer [2] proposed two extensions of AdaBoost for multi-label classification. McCallum [3] adopted a Bayesian approach in which the multiple classes that comprise a document are represented

by a mixture model. Parameters of the mixture model are then learned by the EM algorithm.Clare and King[4] develop new resampling methods to deal with the multi-label problem in gene analysis.

Figure 1. An example image:*mountain+field.*

More recently, Boutell et al [1] studied multi-label scene classification, and proposed training and testing algorithms. In training, they utilized the one-vs-rest approach, building classifiers for all base classes. Hence examples with multi-label are used more than once during training. Their main contribution is a new testing criterion, called C-criterion, by which one can obtain multiple labels for an example from the outputs of the base classifiers. Consider the 2-class problem, two base classifiers output confidence scores of the example's membership in the corresponding classes. Let s_1 and s_2 denote the two scores. Without loss of generality, we assume $s_1 \geq s_2$. To decide whether the example belongs to both classes or only to class I, they compute the difference of the scores. If $s_1 - s_2$ is less than a threshold (determined by the MAP principle [1]), both classes are considered as the labels, otherwise only class I is the label.

In this paper, we focus on 2-class multi-label classification. We present two variants of the support Vector Machines(SVM)[5,6] to handle the multi-label problem. The first algorithm is called Parallel Support Vector Machines (PSVMs), which outputs the labels in one step. The second is referred to as Sequential Support Vector Machines (SSVMs), which predicts labels in two steps. We apply the two methods to semantic scene classification. Experimental results demonstrated that PSVM is comparable to, and SSVM outperforms cross-training C-criterion testing algorithm.

II. Preliminaries

In this section, we describe the formal setting we use to study 2-class multi-label classification problem. Let A and B denote the two classes. For example, in scene classification, A may be the *mountain* scene, and B corresponds to the *field* scene. A and B are not mutually exclusive, that is $A \bigcap B \neq \phi$. An image may contain both mountain and field. In this paper we will always use the *closed world assumption*, meaning that each example belongs to at least one of the two classes.

Let X denote the domain of possible examples. For each instance $x \in X$, we denote its labels by sets such as $\{A\}$, $\{B\}$ and $\{A, B\}$. The whole world can be partitioned in three ways: $A \bigcup \overline{A}$, $B \bigcup \overline{B}$ and $\overline{B} \bigcup (A \bigcap B) \bigcup \overline{A}$ (see Fig. 2), where $\overline{A}$ is the complementary set of A. We often write $A \bigcap \overline{B}$ and $\overline{A} \bigcap B$ instead of $\overline{B}$ and $\overline{A}$ respectively. It is clear that they are equivalent.

Using these notions, we can restate the cross-training algorithm: It trains two SVMs discriminating A vs. $\bar{A}$ and B vs. $\bar{B}$, and then applies the C-criterion to predict the labels.

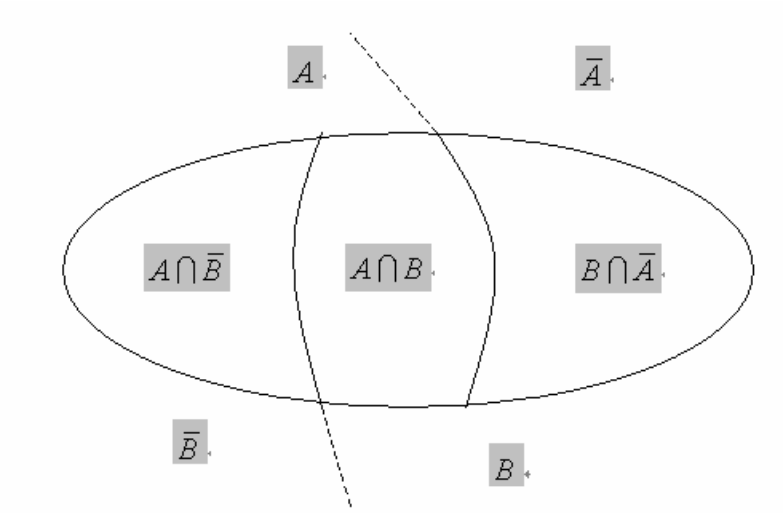


Figure 2. World partitions.

III. Parallel Support Vector Machines

In cross-training, two SVMs are trained separately. For a sample x, it is possible that two SVMs both output negative scores, meaning that $x \notin A$ and $x \notin B$, which violates the closed world assumption. (So they introduced the C-criterion, which depends only on the closeness between the scores, regardless of whether the outputs are positive or negative.) To overcome this problem, we develop a new classifier that can only output one of the three legal label sets: $\{A\}$, $\{B\}$ and $\{A, B\}$ corresponding to $A \cap \bar{B}$, $\bar{A} \cap B$ and $A \cap B$ respectively.

Our method is based on the geometric intuition that examples in $A \cap B$ are very likely to lie between the examples in $A \cap \bar{B}$ and $\bar{A} \cap B$ (see Fig. 3). We use two parallel hyperplanes to discriminate the examples. With two parallel hyperplanes, the classifier never output the illegal result $x \notin A$ and $x \notin B$.(Comparing to cross-training, in which the two independently trained hyperplanes are not parallel, so illegal outputs are possible.) We point out that parallel hyperplanes are only suitable for this multi-label problem. It is not appropriate for ordinary 3-class problem, because one cannot assume, in general, that examples in one class lie between those in the other two classes.

To train the two parallel hyperplanes, we adopt the technique used in standard SVM, which trades off between the training errors and the margins. We call this classifier Parallel Support Vector Machine (PSVM). The major difference in PSVM from SVM is that there are six kinds of classification errors (see also Fig. 3):

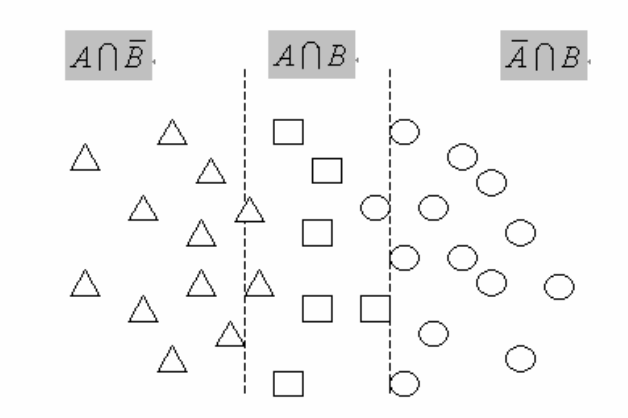


Figure 3. Illustration of the example distribution.

	correct		*wrong*	*correct* *wrong*
1.	$x \in A \cap \bar{B}$	$\rightarrow$	$x \in A \cap B,$	$(\{A\} \rightarrow \{A, B\})$
2.	$x \in A \cap \bar{B}$	$\rightarrow$	$x \in \bar{A} \cap B,$	$(\{A\} \rightarrow \{B\})$
3.	$x \in A \cap B$	$\rightarrow$	$x \in A \cap \bar{B},$	$(\{A, B\} \rightarrow \{A\})$,
4.	$x \in A \cap B$	$\rightarrow$	$x \in \bar{A} \cap B,$	$(\{A, B\} \rightarrow \{B\})$
5.	$x \in \bar{A} \cap B$	$\rightarrow$	$x \in A \cap B,$	$(\{B\} \rightarrow \{A, B\})$
6.	$x \in \bar{A} \cap B$	$\rightarrow$	$x \in A \cap \bar{B},$	$(\{B\} \rightarrow \{A\}).$

(1)

Let the two hyperplanes be $w^T x + b_1$ and $w^T x + b_2$. Let examples in $A \cap \bar{B}$, $A \cap B$ and $\bar{A} \cap B$ be denoted by x_i^1, x_i^2 and x_i^3 respectively. If the training error is zero, then

$$\begin{aligned} &w^T x_i^1 + b_1 > 0, \quad w^T x_i^1 + b_2 > 0; \\ &w^T x_i^2 + b_1 < 0, \quad w^T x_i^2 + b_2 > 0;, \\ &w^T x_i^3 + b_1 < 0, \quad w^T x_i^3 + b_2 < 0. \end{aligned} \tag{2}$$

Accordingly, the primal optimization problem is

$$\min \frac{1}{2}\|w\|^2 + C(\sum_i \xi_i^1 + \sum_i \xi_i^2 + \sum_i \xi_i^3 + \sum_i \xi_i^4 + \sum_i \xi_i^5 + \sum_i \xi_i^6),$$

$$\begin{aligned} s.t. \quad & w^T x_i^1 + b_1 \geq 1 - \xi_i^1 && (\{A\} \rightarrow \{A, B\}) \\ & w^T x_i^1 + b_2 \geq 1 - \xi_i^2 && (\{A\} \rightarrow \{B\}) \\ & -(w^T x_i^2 + b_1) \geq 1 - \xi_i^3 && (\{A, B\} \rightarrow \{A\}) \\ & w^T x_i^2 + b_2 \geq 1 - \xi_i^4 && (\{A, B\} \rightarrow \{B\}) \quad , \\ & -(w^T x_i^3 + b_2) \geq 1 - \xi_i^5 && (\{B\} \rightarrow \{A, B\}) \\ & -(w^T x_i^3 + b_1) \geq 1 - \xi_i^6 && (\{B\} \rightarrow \{A\}). \\ & \xi_i^1, \xi_i^2, \xi_i^3, \xi_i^4, \xi_i^5, \xi_i^6 \geq 0 \end{aligned} \tag{3}$$

where $\xi_i^1, \xi_i^2, \xi_i^3, \xi_i^4, \xi_i^5, \xi_i^6$ are positive slack variables. Its Lagrange function is written by

$$\begin{aligned}
&\frac{1}{2}\|w\|^2 + C(\sum_i \xi_i^1 + \sum_i \xi_i^2 + \sum_i \xi_i^3 + \sum_i \xi_i^4 + \sum_i \xi_i^5 + \sum_i \xi_i^6) \\
&-\sum_i \alpha_i^1 (w^T x_i^1 + b_1 - 1 + \xi_i^1) - \sum_i \alpha_i^2 (w^T x_i^1 + b_2 - 1 + \xi_i^2) \\
&-\sum_i \alpha_i^3 (-(w^T x_i^2 + b_1) - 1 + \xi_i^3) - \sum_i \alpha_i^4 (w^T x_i^2 + b_2 - 1 + \xi_i^4) \quad , \\
&-\sum_i \alpha_i^5 (-(w^T x_i^3 + b_2) - 1 + \xi_i^5) - \sum_i \alpha_i^6 (-(w^T x_i^3 + b_1) - 1 + \xi_i^6) \\
&-\sum_i \beta_i^1 \xi_i^1 - \sum_i \beta_i^2 \xi_i^2 - \sum_i \beta_i^3 \xi_i^3 - \sum_i \beta_i^4 \xi_i^4 - \sum_i \beta_i^5 \xi_i^5 - \sum_i \beta_i^6 \xi_i^6 .
\end{aligned} \tag{4}$$

where the positive numbers $\alpha_i^1, \alpha_i^2, \alpha_i^3, \alpha_i^4, \alpha_i^5, \alpha_i^6$ and $\beta_i^1, \beta_i^2, \beta_i^3, \beta_i^4, \beta_i^5, \beta_i^6$ are Lagrangian multipliers. Differentiate with respect to w, b_1, b_2 and $\xi_i^1, \xi_i^2, \xi_i^3, \xi_i^4, \xi_i^5, \xi_i^6$, we obtain

$$w = \sum_i \alpha_i^1 x_i^1 + \sum_i \alpha_i^2 x_i^1 - \sum_i \alpha_i^3 x_i^2 + \sum_i \alpha_i^4 x_i^2 - \sum_i \alpha_i^5 x_i^3 - \sum_i \alpha_i^6 x_i^3 ., \tag{5}$$

For notational convenience, let

$$\tilde{x}_i^1 = \tilde{x}_i^2 = x_i^1, \qquad \tilde{x}_i^3 = \tilde{x}_i^4 = x_i^2, \qquad \tilde{x}_i^5 = \tilde{x}_i^6 = x_i^3,$$

and

$$\begin{aligned} y^1 = y^2 = y^4 = 1, \\ y^3 = y^5 = y^6 = -1. \end{aligned}$$

We have the dual optimization problem of (3):

$$\begin{aligned}
&\max_{\alpha_i^1, \alpha_i^2, \alpha_i^3, \alpha_i^4, \alpha_i^5, \alpha_i^6} \sum_{p=1}^{6} \sum_i \alpha_i^p - \frac{1}{2} \sum_{p=1}^{6} \sum_{q=1}^{6} \sum_i \sum_j \alpha_i^p \alpha_j^q y^p y^q ((\tilde{x}_i^p)^T (\tilde{x}_j^q)). \\
&s.t. \qquad \sum_i \alpha_i^1 - \sum_i \alpha_i^3 - \sum_i \alpha_i^6 = 0, \\
&\qquad\quad \sum_i \alpha_i^2 + \sum_i \alpha_i^4 - \sum_i \alpha_i^5 = 0, \\
&\qquad\quad 0 \le \alpha_i^1, \alpha_i^2, \alpha_i^3, \alpha_i^4, \alpha_i^5, \alpha_i^6 \le C.
\end{aligned} \quad , \tag{6}$$

This is a quadratic programming (QP) problem similar to standard SVM except that in our PSVM there are two equality constrains while in SVM only one. The biases b_1 and b_2 are easily found by the KKT conditions.

The above derivations can be generalized to feature space using the kernel trick, replacing inner product by a positive definite kernel such as the Gaussian function.

IV. Sequential Support Vector Machines

In this Section, we present another algorithm for multi-label classification. To better describe our motivation, we first interpret the C-criterion testing method from an alternative point of view.

The C-criterion algorithm may be seen as a two-stage classification. In the first step, one decides whether an example x has multi-label or not. That is, if $x \in A \bigcap B$. The criterion is how close the outputs of the two base classifiers. If $x \notin A \bigcap B$, the next step is then to determine if $x \in A$ or $x \in B$. The decision depends again on the outputs of the base classifiers.

Our algorithm, called sequential SVM (SSVM) is analogous to the above two-stage classification method. However, in both steps, we use different classification criteria. We first study the second decision step, assuming that we have determined $x \notin A \bigcap B$. This means that we need to tell if $x \in A \bigcap \overline{B}$ or $x \in \overline{A} \bigcap B$. One should note that this is a 2-class single label classification problem. So we use a SVM trained by examples in $A \bigcap \overline{B}$ and $\overline{A} \bigcap B$ only. In most situations, this will not decrease the training sample size significantly, since examples in combined classes are usually sparse. In the first decision step, we also call for a SVM to decide if $x \in A \bigcap B$ or not. The training samples are as follows: examples in $A \bigcap \overline{B}$ are labeled as positive; examples in $\overline{A} \bigcap B$ are labeled as negative; each example in $A \bigcap B$ is duplicated, one labeled positive and the other negative. We argue that examples in $A \bigcap \overline{B}$ and $\overline{A} \bigcap B$ are usually far from the separating hyperplane, since SVM is a large margin classifier. And examples in $A \bigcap B$ should be close to the hyperplane, because duplication of these data ensures that any bias from the hyperplane would cause training error increase. Hence an example is classified as in $A \bigcap B$ if it is close enough to the hyperplane, i.e. the SVM output is around zero. We adopt the MAP principle to determine the threshold, as for the C-criterion [1].

V. Experimental Results

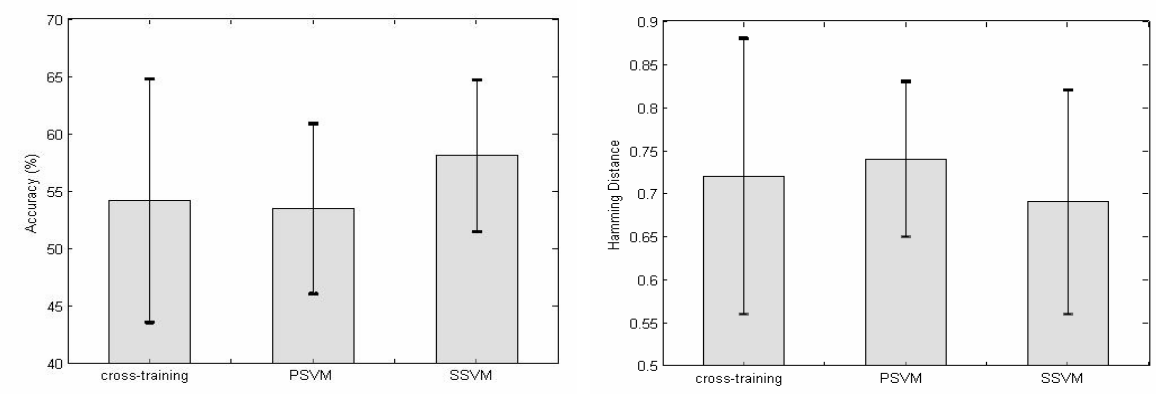
We apply the two algorithms PSVM and SSVM to semantic scene classification, and compare them to the cross-training C-criterion testing method. As in [1], we use spatial color moments as features. We collected 262 images, among which 121 contain *mountain* only, 114 are *field* only, and 27 are *mountain+field*. 10-fold cross validation is adopted to efficiently make use of the examples. For 2-class multi-label classification problem, many complicated evaluation metrics reduce to equivalent simple forms. Here, we use two measures to evaluate the classification results.

An example is said to be correctly classified if and only if the prediction agrees exactly with the ground truth labels. We define the accuracy as

$$accuracy = \frac{number\ \ of\ \ correctly\ \ classified\ \ examples}{number\ \ of\ \ all\ \ testing\ \ examples}.$$

Accuracy is a "strict" evaluation measure. Another measure is the commonly used Hamming distance. For example, the Hamming distance between $\{A\}$ and $\{A, B\}$ is 1, and that between $\{A\}$ and $\{B\}$ is 2.

We plotted classification results of PSVM, SSVM and cross-training in Fig. 4, evaluated by accuracy and Hamming distance respectively. It shows that PSVM is comparable to and SSVM outperforms cross-training C-criterion testing method in this scene classification experiment.



(a) (b)

Figure 4. Scene classification results: (a) accuracy; (b) Hamming distance.

VI. Conclusions

We propose two algorithms PSVM and SSVM for 2-class multi-label classification problems. PSVM makes decision in one step, while SSVM in two successive steps. When applied to scene classification, SSVM outperforms the cross-training method in our experiment. Although PSVM seems to be inferior to SSVM, it is partly due to the fact that the QP solver used to solve the optimization problem (6) is not specifically intended for this problem. There may be potential for an improvement of PSVM in the future.

Acknowledgment

This work was supported by NKBRPC (2004CB318000) and NSFC (60575002).

References

[1] M.R. Boutell, J. Juo, X. Shen, C.M. Brown, “Learning Multi-label Scene Classification,” *Pattern Recognition* 37(9) (2004).

[2] R. Schapire, Y. Singer, “BoosTexter: A Boosting-based System for Text Categorization,” *Machine Learning*, 39(2/3) (2000) 135-168.

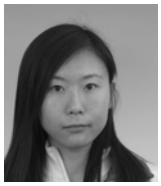
[3] A.K. McCallum, “Multi-label Text Classification with a Mixture Model Trained by EM,” *AAAI'99 Workshop on Text Learning*.

[4] A. Clare, R.D. King, “Knowledge Discovery in Multi-label Phenotype Data,” *Lecture Notes in Computer Science*, Vol. 2168, Springer, Berlin, (2001).

[5] Cortes, V. Vapnik, “Support Vector Networks,” *Machine Learning*, 20(9) (1995) 273-297.

[6] V. Vapnik, “Statistical Learning Theory,” John Wiely, New York, (1998).

Liwei Wang received the B.S., M.S. degree in Electronic Engineering from Tsinghua University in 1999 and 2002 respectively and the Ph.D. degree in Mathematics from Peking University in 2005. He is currently an assistant professor in the Center for Information Sciences and National Laboratory on Machine Perception, School of Electronics Engineering and Computer.



Ming Chang received the B.S. degree in 2003 in mathematics from Peking University, Beijing, P. R. China. Now she is a third-year postgraduate student in the Center for Information Sciences and National Laboratory on Machine Perception, School of Electronics Engineering and Computer Science, Peking University. Her interests include pattern recognition and machine.

Jufu Feng received the B.S. degree in 1989 and Ph.D. degree in 1997 both in mathematics from Peking University, Beijing, P. R. China. Since 1992, he has been with the Center for Information Science and National Laboratory on Machine Perception at Peking University. His current research interests include image processing, pattern recognition and biometrics. He is currently a professor in the Center for Information Science and National Laboratory on Machine Perception, School of Electronics Engineering and Computer Science, Peking University.