Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition Letters 28 (2007) 2088–2093

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Reducing the number of sub-classifiers for pairwise multi-category support vector machines

Wang Ye *, Huang Shang-Teng

Department of Computer Science and Engineering, Shanghai JiaoTong University, China

Received 4 January 2005; received in revised form 1 June 2007
Available online 25 July 2007

Communicated by L. Bottou

Abstract

Among the SVM-based methods for multi-category classification, "1-a-r", pairwise and DAGSVM are most widely used. The deficiency of "1-a-r" is long training time and unclassifiable region; the deficiency of pairwise and DAGSVM is the redundancy of sub-classifiers. We propose an uncertainty sampling-based multi-category SVM in this paper. In the new method, some necessary sub-classifiers instead of all $N \times (N-1)/2$ sub-classifiers are selected to be trained and the uncertainty sampling strategy is used to decide which samples should be selected in each training round. This uncertainty sampling-based method is proved to be accurate and efficient by experimental results on the benchmark data.

Keywords: SVM; Multi-category classification; Pairwise; Uncertainty sampling

1. Introduction

The Support Vector Machine (SVM) is a powerful technique for classification. It classifies positive and negative samples by searching a hyperplane with the largest margin between them, so that better generalization performance and fewer training errors can be obtained. In this paper, we will discuss SVM for multi-category classification, which means the number of the categories is more than two.

Generally, the binary (two-category) SVM can be extended to multi-category case in two ways. The first way is considering all categories in one optimization problem. According to this way, a multi-category problem is formulated into one optimization equation, but there are too many parameters to adjust, so it is inefficient. The second way is constructing several binary sub-classifiers. In this way, multi-category problems are treated as a series of binary sub-problems, and many methods are developed based on this idea. Compared with the first, the second way is more widely used.

Although many methods of the second way are available, when the number of the categories or the size of each category is quite large, these methods are faced with a common problem, that is, it takes a very long time for all binary SVM sub-classifiers being trained. Targeted on this, we propose an uncertainty sampling-based multi-category SVM (abbreviated as US_MSVM) in this paper. Faster than "1-a-r" and pairwise, the new method has similar average accuracy with them. In each round of US_MSVM, samples of the two most indistinguishable categories are selected for the next training round. After a training round, the probabilities of positive samples (PPS) are used to decide which two categories are most indistinguishable.

The remaining of this paper is organized as follows: In Section 2, we briefly review the current research situation of multi-category SVM. The main idea of uncertainty sampling strategy will be introduced in Section 3. The new method, US_MSVM, is presented and analyzed in Section

* Corresponding author. Tel./fax: +86 21 5448 5310.
E-mail address: wangye34@126.com (W. Ye).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.06.020

4. Experimental results of the performance comparison between the new method and pairwise classifier on the benchmark data are shown in Section 5. Conclusions are drawn in Section 6.

2. Multi-category SVM

The basic form of SVM is presented to solve the problem of two-category linearly separable cases (Vapnik, 1995). By using kernel functions and slack variables, SVM can be extended to solve problems of nonlinearly cases and non-separable cases. A multi-category problem can be converted into a series of two-category sub-problems. “1-a-r” (Bennett, 1999), pairwise (“1-a-1”) (Kreßel, 1999), Decision Directed Acyclic Graph (DDAG) (Platt et al., 2000) and Adaptive Directed Acyclic Graph (ADAG) (Kijsirikul and Ussivakul, 2002) are all based on this idea.

The “1-a-r” method is used to combine N binary sub-classifiers, where N is the number of the categories. In the ith round of the training phase, samples of the ith category are labeled positive, and all others are labeled negative. The advantage of “1-a-r” is simple architecture and high testing speed, but it costs long time for training and the unclassifiable region is quite large (Shigeo, 2003).

The pairwise method is used to combine $N\times(N-1)/2$ binary sub-classifiers and each sub-classifier is trained on samples of two out of N categories. In the testing phase, the Max Wins algorithm is adopted, that is, the final result is the category gets more supports. According to Shigeo (2003), Abe and Inoue (2002), The pairwise classifier costs less training time and has smaller unclassifiable region than “1-a-r”.

To solve the unclassifiable region problem in “1-a-r” and pairwise, Platt proposed the DDAG, which is a special pairwise classifier. The training phase of DDAG is the same as the pairwise method by solving $N\times(N-1)/2$ binary SVMs. In the testing phase, these SVMs are arranged in an N-layer DAG. Excluding impossible categories step by step, The DDAG labels a sample with its most possible category label at the bottom of the DAG, as is shown in Fig. 1.

Kijsirikul and Ussivakul proposed a tournament-based classifier: ADAG. The training phase of ADAG and DDAG are the same. In each testing round of the ADAG, the number of the candidate categories reduces by half. The final label is given after the last decision is made, as is shown in Fig. 1b. Pontil and Verri (1998) proposed another version of the ADAG.

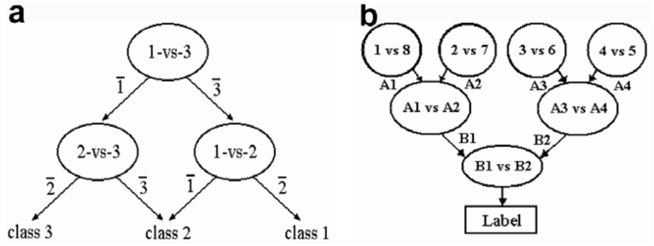

Fig. 1. The testing process of (a) DDAG and (b) ADAG.

Compared with the training time, the testing time of the Multi-category SVM can be ignored generally (Shigeo, 2003). As the training time as concerned, the complexity of pairwise is $2^{\gamma-1}cN^2-\gamma m^\gamma$ and the complexity of “1-a-r” is cNm^γ (Shigeo, 2003). Here, N is the number of categories and m is the number of all training samples and c is a constant. γ is equal to 2, when decomposition method is used to solve SVM (Shigeo, 2003). Clearly, the complexity of pairwise is lower than that of “1-a-r”.

3. Uncertainty sampling

Before introducing our new method, we will review the uncertainty sampling strategy (Lewis and Gale, 1994) firstly. The uncertainty sampling strategy is an important sampling selecting strategy used in active learning. Active learning (Simon and Lea, 1974; Winston, 1975) is an efficient supervised learning algorithm that actively selects “helpful” samples to learn, instead of learning from the original training set passively. The uncertainty sampling strategy is used to select the “helpful” samples by measuring their uncertainty to the current classifier.

A typical active learning framework is described in (Tong, 2001). In active learning, the whole data are divided into labeled samples X and unlabeled samples U. There is also a learner l and a deciding module q. The learner l is trained on the labeled samples X and the module q is used to decide which samples of U should be selected and labeled, and should be added into X. The updated X will be used to train l in the next step. According to the difference mechanism of deciding modules, active learning methods can be divided into two groups: uncertainty sampling and query by committee (QBC) Seung et al. (1992).

The main idea of uncertainty sampling is that a classifier will benefit more from being trained on samples, which it is more uncertain to current classifier. Uncertainty sampling requires a probabilistic classifier that assigns to unlabeled samples each possible label with a certain probability. The unlabeled samples with most uncertainty are selected and labeled, and then are added into X. Various methods for measuring uncertainty have been proposed Lewis and Gale (1994), Iyengar et al. (2000). Query by committee is another group of active learning methods. It is based on the disagreement among a committee of classifiers.

Active learning is effective on saving labeled data and has been applied to various fields, such as natural language parsing, spoken language understanding, feature selection and text classification. In our method, we will use uncertainty sampling as a sample selecting strategy to decide which two categories are most indistinguishable.

4. Uncertainty sampling-based MSVM

As reviewed in Section 2, the sub-classifiers of all $N\times(N-1)/2$ pairs should be trained. Are these sub-classifiers

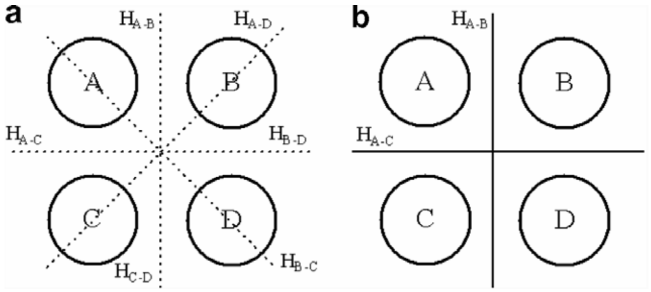

Fig. 2. (a) The hyperplanes of pairwise. (b) The necessary hyperplanes.

all necessary? Consider a multi-category classification problem as shown in Fig. 2. To distinguish A, B, C and D, six sub-classifiers should be trained, which are represented by the dash lines in Fig. 2. However, only two hyperplanes of them are really necessary, which are represented by the solid lines in Fig. 2b. Maybe this example is a little extreme, but it shows that some sub-classifiers of pairwise are redundant.

Our improving motivation is to train necessary sub-classifiers and to ignore those trivial ones. The main idea of the uncertainty sampling-based MSVM is constructing the "most helpful" sub-classifier gradually, according to the strategy of uncertainty sampling. In the training phase, samples of two categories are selected and labeled positive or negative, and the uncertainty sampling strategy is used to decide which two categories should be selected. In the testing phase, the final result is the integrative opinion of all sub-classifiers.

Definition 1. The probability of positive samples (PPS) for category i of SVM sub-classifier k is defined as follows:

$$\mathrm{PPS}_{k,i} = \frac{\mathrm{card}(f_k(x_j^i) > 0)}{\mathrm{card}(x_j^i)}. \tag{4.1}$$

Here, x_j^i is the jth sample of category i. Function card(.) is to get the cardinal number and $f_k(.)$ is the decision function of SVM sub-classifier k. When $f_k\,(x_j)$ is greater than 0, x_j is labeled positive, else negative. Obviously, the value of PPS varies from 0 to 1. If the PPS to two categories of a certain sub-classifier are quite similar; then these two categories are indiscriminatingly to this sub-classifier.

Definition 2. The decision matrix (DM) is defined as follows:

$$DM = \begin{bmatrix} \mathrm{PPS}_{1,1} & \mathrm{PPS}_{1,2} & \cdots & \mathrm{PPS}_{1,N} \\ \mathrm{PPS}_{2,1} & \mathrm{PPS}_{2,2} & \cdots & \mathrm{PPS}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{PPS}_{k,1} & \mathrm{PPS}_{k,2} & \cdots & \mathrm{PPS}_{k,N} \end{bmatrix}. \tag{4.2}$$

Here, k is the number of trained sub-classifiers and N is the number of categories. The DM can be used to measure the uncertainty of categories. Notice that the ith column vector comprises the PPS's for a certain category i of all trained sub-classifiers, so if two column vectors of DM are similar, that means the corresponding two categories are indiscriminatingly to all trained sub-classifiers. The samples of these two categories should be selected as the training subset on which a new sub-classifier should be trained.

Now, what we need is a similarity measure of two column vectors. The similarity of the column vectors in the DM can be measured using the distance of column vectors. The distance measuring formula is described in Eq. (4.3)

$$\mathrm{dist}_{i,j} = \left\| \begin{bmatrix} \mathrm{PPS}_{1,i} \\ \mathrm{PPS}_{2,i} \\ \vdots \\ \mathrm{PPS}_{k,i} \end{bmatrix} - \begin{bmatrix} \mathrm{PPS}_{1,j} \\ \mathrm{PPS}_{2,j} \\ \vdots \\ \mathrm{PPS}_{k,j} \end{bmatrix} \right\|. \tag{4.3}$$

The training algorithm of the US_MSVM is listed as follows. The input of the algorithm is the training set of several categories and its output is a trained learner that contains some sub-classifiers.

(1) Specify the maximum round of loop r and a distance threshold d^*. Let $k = 1$; select two categories C_{j1} and C_{j2} randomly; let Trained_Pair $= \{\langle C_{j1}, C_{j2}\rangle\}$.
(2) Select the samples whose category is C_{j1} or C_{j2} to make a new training subset; train a SVM sub-classifier l_k on this subset.
(3) Test a subset of samples using l_k, and calculate $\mathrm{PPS}_{k,i}$ for every category using Eq. (4.1), and add $\mathrm{PPS}_{k,i}$ to the DM as a new row vector.

Table 5.2
Topics of the selected categories in "20-NG"

Grp	Topics	Grp	Topics
1	comp.os.ms-windows.misc	6	sci.med
2	rec.sport.baseball	7	talk.politics.misc
3	talk.religion.misc	8	rec.autos
4	sci.space	9	misc.forsale
5	comp.sys.mac.hardware	10	talk.politics.mideast

Table 5.1
Topics and numbers of the most 8 categories in "Reuters"

	Earn	acq	Money-fx	Grain	Crude	Trade	Interest	Ship
# Training set	2703	1487	460	391	350	336	286	190
# Testing set	1057	718	219	326	212	174	134	102

Table 5.3
F-1 comparison of the first test ("Reuters", $r = 14$)

	Earn	acq	Money-fx	Grain	Crude	Trade	Interest	Ship
US_MSVM	98.0%	93.7%	70.9%	95.4%	83.1%	78.8%	72.9%	76.1%
Pairwise	98.3%	95.7%	74.5%	94.6%	82.9%	79.3%	73.2%	76.5%

Table 5.4
F-1 comparison of the second test ("20-NG", $r = 23$)

Group	Grp 1	Grp 2	Grp 3	Grp 4	Grp 5	Grp 6	Grp 7	Grp 8	Grp 9	Grp 10
US_MSVM	87.5%	93.8%	81.2%	86.9%	84.5%	86.1%	73.0%	91.4%	78.6%	87.2%
Pairwise	87.5%	95.4%	81.2%	90.7%	85.0%	88.5%	73.0%	93.3%	80.2%	87.2%

Table 5.5
F-1 comparison of the third test ("USPS", $r = 23$)

Number	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
US_MSVM	93.5%	95.8%	88.0%	89.1%	93.1%	86.9%	95.5%	91.7%	90.3%	94.2%
Pairwise	93.5%	96.2%	88.6%	90.0%	95.6%	89.3%	96.0%	92.4%	91.5%	95.5%

(4) Measure the distances of all pairs of categories according to Eq. (4.3), excluding the pairs have been added into Trained_Pair. Select two categories that are most indiscriminatingly as the new C_{j1} and C_{j2}, add the new $\langle C_{j1}, C_{j2}\rangle$ into the Trained_Pair. $k = k + 1$.

(5) Loop Step (2)–Step (4) until $k = r$ or the minimum distance of all category pairs is greater than d^*.

The time complexity of Step (2) is $O(m^2)$, where m is the number of all samples. The time complexity of Step (3) is $O(m' \times n\_sv)$, where m' is the size of the testing subset and n_sv is the number of support vectors. Since $m' \leqslant m$ and $n\_sv \leqslant m$, in the worst case, $O(m' \times n\_sv)$ is comparable to $O(m^2)$. Step (4) is not a time consuming step, for its time complexity is $O(N^2)$. Here, N is the number of the categories. Since $O(N^2)$ is lower than $O(m^2)$, the time complexity of the whole algorithm is $O(m^2)$.

The testing phase of US_MSVM is described as follows. Each sub-classifier predicts whether a testing sample is positive or negative, respectively. Let PY be a vector, whose elements are the predicted results of each sub-classifier, as is shown in Eq. (4.4)

$$PY = [py_1 \quad py_2 \quad \cdots \quad py_R], \quad py_i = \begin{cases} 1, & f(x) \geqslant 0, \\ 0, & f(x) < 0. \end{cases} \tag{4.4}$$

Here, the subscript R is the number of trained sub-classifiers, $R \leqslant r$. Each column vector of DM should be compared with PY^{T}. If a column vector and PY^{T} are similar, the corresponding category and the testing sample are indiscriminatingly to all trained sub-classifiers, that is, the testing sample belongs to this category with large possibility. The predicting formula is described as follows.

$$category = \arg\min_i \left\| PY^{\mathrm{T}} - [\mathrm{PPS}_{1,i} \quad \mathrm{PPS}_{2,i} \quad \cdots \quad \mathrm{PPS}_{R,i}]^{\mathrm{T}} \right\|. \tag{4.5}$$

Since the time complexity of Step (4) is lower than that of Step (2) and Step (3), the time complexity of US_MSVM can be calculated as follows. According to the complexity formula in (Shigeo, 2003) and the analysis above, the time complexity of US_MSVM should be $cR(2m/N)^{\gamma} + R(m' \times n\_sv)$, where the first item is the time complexity of Step (2) and the second item is the time complexity of Step (3). Here, R is the number of sub-classifiers being trained and $2m/N$ is the size of the training subset.

5. Experimental results

To evaluate the performance of the US_MSVM, experimental results of the US_MSVM on "Reuters-21578", "20-NG" and "USPS" are compared with those of pairwise. "Reuters-21578" and "20-NG" are real-world text datasets and "USPS" is a real-world text dataset of digital images.

Among all 135 categories of "Reuters-21578", we select the most eight categories[1] as the subset of the first experiment. This subset includes 6203 training samples and 2942 testing samples. The numbers and topics of texts of the selected categories are listed in Table 5.1.

Ten categories of "20-NG" are selected as the subset of the second experiment. For each category, 250 documents are selected as the training set and other 100 documents

[1] Most 10 categories of "Reuters-21578" are often used, which include the categories of "wheat" and "corn", but we find every sample of "wheat" and "corn" also belongs to the category of "grain", so we use most 8 categories.

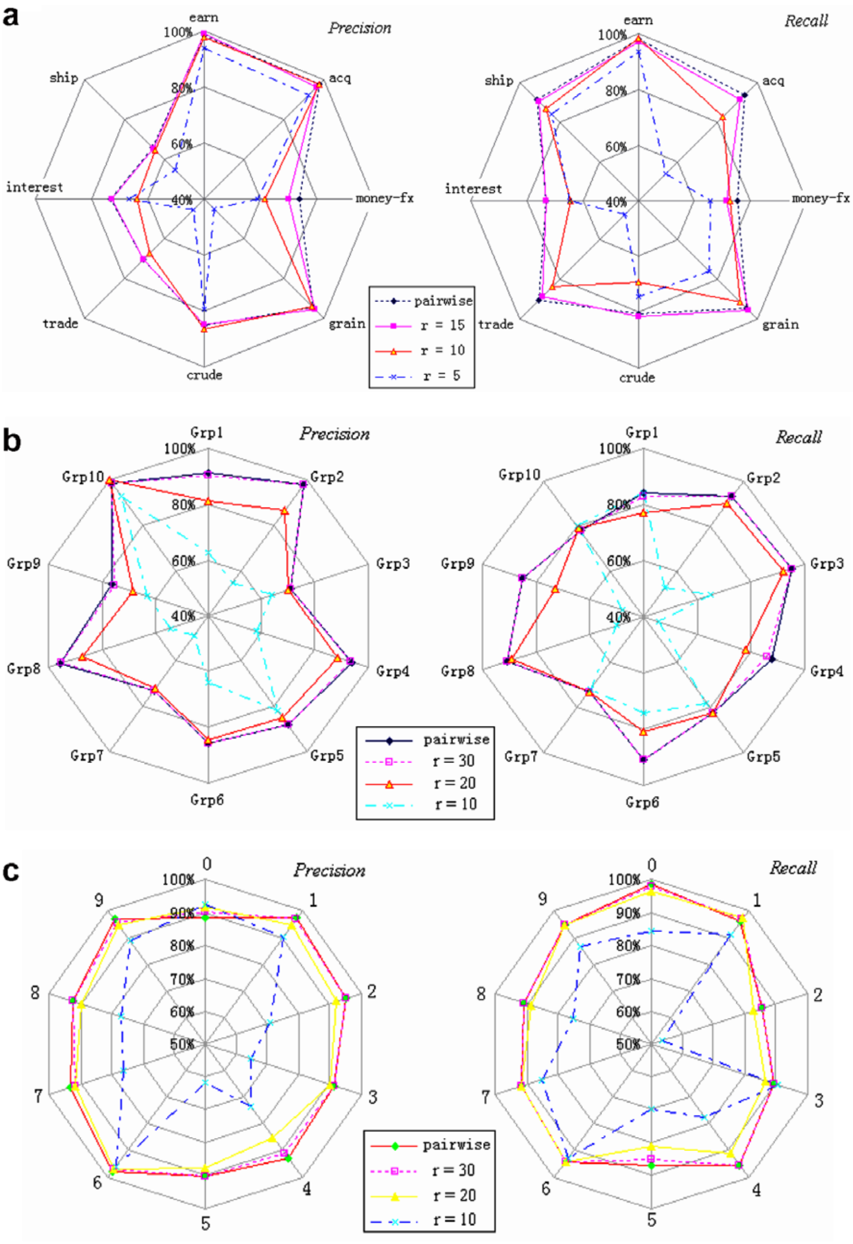

Fig. 3. The influence of training rounds to Precision and Recall in the first test (Reuters) (a), second test (20-NGs) (b) and third test (USPS) (c).

are selected as the testing set, randomly. Topics of the selected categories are listed in Table 5.2.

After stop words filtering and stemming, these documents are transformed into vectors in a high dimension space. The value of each item in a vector can be calculated using the *tf-idf* equation. All images of "USPS" are used as a dataset of the third experiment, including 7291 training samples and 2007 testing samples.

All experiments are performed on a personal computer, which has a 2.0 GHz Intel processor and 512 MB memory. Precision, Recall and F-1 are used as the measures for performance evaluating. The F-1 measure of the US_MSVM and pairwise in three experiments are listed in Tables 5.3–5.5. The algorithm of US_MSVM is implemented by using LIBSVM. In these experiments, r is set to $0.5 \times N(N-1)/2$ and d^{*} is set to 0.8; the size of the testing subset of each category in Step (3) is set to 30. When parameters tuning, we find using $\gamma = 0.1$ and $C = 100$, the classification results of "Reuters", "20-NG" and "USPS" are all satisfying, so we use $\gamma = 0.1$ and $C = 100$ as SVM parameters to train all sub-classifiers of US_MSVM and pairwise.

Table 5.6
Comparison of the training time

	US_MSVM			Pairwise
"Reuters"	13.58 (5)	20.35 (10)	27.49 (15)	37.57 (28)
"20-NG"	3.15 (10)	6.14 (20)	8.95 (30)	10.82 (45)
"USPS"	13.32 (10)	27.51 (20)	34.68 (30)	45.26 (45)

Table 5.7
The time of each step of the new algorithm

	"Reuters"			"20-NG"			"USPS"		
	$r = 15$	$r = 10$	$r = 5$	$r = 30$	$r = 20$	$r = 10$	$r = 30$	$r = 20$	$r = 10$
Total	27.49	20.35	13.58	8.95	6.14	3.15	34.68	27.51	13.32
Step (2)	25.08	18.52	12.45	7.34	5.05	2.64	29.50	23.46	11.17
Step (3)	2.39	1.81	1.12	1.61	1.09	0.51	5.13	4.02	2.14
Step (4)	0.02	0.02	0.01	0.00	0.00	0.00	0.05	0.03	0.01

From Tables 5.3–5.5, we can find that when $r = 0.5N_{\text{all}}$, the F-1 measures of US_MSVM are quite close to pairwise. Compared with pairwise, the maximum F-1 losses of US_MSVM of three datasets are 3.6%, 3.8% and 2.5%, respectively. In some cases, F-1 of US_MSVM is even a little higher.

These experiments are all performed in the case that the loop round r is a half of N_{all}. To evaluate the influence of r, some additional experiments are performed. Fig. 3a–c shows the influence of r to Precision and Recall. Precision and Recall of pairwise are also shown for comparison.

In Fig. 3a–c, we can find that when the value of r is small, Precision and Recall of US_MSVM is poor. When r increases, Precision and Recall of US_MSVM increase. In the case that r is greater than $0.5N_{\text{all}}$, Precision and Recall of US_MSVM are both very close to pairwise.

This phenomenon can be explained as follows. In the US_MSVM, sub-classifiers are trained in order of their significance, so the sub-classifiers formerly trained are more "helpful" than the sub-classifiers latterly trained. As a result, when the number of trained sub-classifiers is greater than a threshold (such as $0.5N_{\text{all}}$), all untrained sub-classifiers are almost "unhelpful". The comparison of the training time of the US_MSVM and pairwise is listed in Table 5.6.

In Table 5.6, the number in the parentheses is the training round (r) and the number out of the parentheses is the CPU time whose unit is "second". From Table 5.6, we can see the training time of US_MSVM increases with the increasing of r. When r is less than $2/3N_{\text{all}}$, the US_MSVM is faster than the pairwise remarkably. According to our experimental results, an appropriate value of r is $1/2N_{\text{all}}$ to $2/3N_{\text{all}}$. The time of each step of the new algorithm is listed in Table 5.7.

In Table 5.7, the unit of the CPU time is "second" and r is the training round. From Table 5.7, we can find that Step (2) is the main spending of the algorithm.

6. Conclusions

In this paper, we propose a novel uncertainty sampling-based method, US_MSVM, to solve multi-category classification problems. In the new method, sub-classifiers are trained in order of their significances and those unhelpful sub-classifiers are ignored. The uncertainty sampling strategy is used to decide which samples should be trained in the next round. When testing, the final result is the integrative opinion of all trained sub-classifiers.

Experimental results on real-world data set show that, Precision and Recall of US_MSVM are comparable to those of pairwise in the condition that the training round is much less than that of the pairwise. The US_MSVM can be used as a substituted version of pairwise.

References

Abe, S., Inoue, T., 2002. Fuzzy support vector machines for multiclass problems. In: Proc. 10th European Symp. on Artificial Neural Networks (ESANN2002), Bruges, Belgium, April, pp. 113–118.

Bennett, K.P., 1999. Combining support vector and mathematical programming methods for classification. In: Scholkopf, B., Burges, C.J.C., Smola, A.J. (Eds.), Advances in Kernel Methods: Support Vector Learning. The MIT Press, Cambridge, MA, pp. 307–326.

Iyengar, V.S., Apte, C., Zhang, T., 2000. Active learning using adaptive resampling. In: Sixth ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining, pp. 92–98.

Kijsirikul, B., Ussivakul, N., 2002. Multiclass support vector machines using adaptive directed acyclic graph. In: Proc. Internat. Joint Conf. on Neural Networks (IJCNN2002), pp. 980–985.

Kreßel, U.H.-G., 1999. Pairwise classification and support vector machines. In: Scholkopf, B., Burges, C.J.C., Smola, A.J. (Eds.), Advances in Kernel Methods: Support Vector Learning. The MIT Press, Cambridge, MA, pp. 255–268.

Lewis, D.D., Gale, W.A., 1994. A sequential algorithm for training text classifiers. In: Croft, W.B., van Rijsbergen, C.J. (Eds.), Proc. SIGIR-94, 17th ACM Internat. Conf. on Research and Development in Information Retrieval, Dublin, IE. Springer Verlag, Heidelberg, DE, pp. 3–12.

Platt, J.C., Cristianini, N., Shawe-Taylor, J., 2000. Large margin DAGs for multiclass classification. In: Solla, S.A., Leen, T.K., Muller, K.R. (Eds.), . In: Advances in Neural Information Processing Systems, vol. 12. The MIT Press, pp. 547–553.

Pontil, M., Verri, A., 1998. Support vector machines for 3-d object recognition. IEEE Trans. Pattern Anal. Machine Intell. 20 (6), 637–646.

Seung, H.S., Opper, M., Sompolinsky, H, 1992. Query by committee. In: Computational Learning Theory, pp. 287–294.

Shigeo, Abe, 2003. Analysis of multiclass support vector machines. In: Proc. Internat. Conf. on Computational Intelligence for Modelling Control and Automation (CIMCA'2003), Vienna, Austria, February, pp. 385–396.

Simon, Herbert A., Lea, Glenn, 1974. Problem solving and rule induction: A unified view. In: Gregg, L.W. (Ed.), Knowledge and Cognition. Erlbaum.

Tong, S., 2001. Active learning: Theory and applications.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag.

Winston, P.H., 1975. Learning structural descriptions from examples. In: Winston, P.H. (Ed.), The Psychology of Computer Vision. McGraw-Hill, New York.