

# Improved DAG SVM: A New Method for Multi-Class SVM Classification

Mostafa Sabzekar, Mohammad GhasemiGol, Mahmoud Naghibzadeh, Hadi Sadoghi Yazdi  
Department of computer Engineering, Ferdowsi University of Mashhad, Iran

**Abstract** - In this paper, we present our method which is a performance improvement to the Directed Acyclic Graph Support Vector Machines (DAG SVM). It suggests a weighted multi-class classification technique which divides the input space into several subspaces. In the training phase of the technique, for each subspace, a DAG SVM is trained and its probability density function (pdf) is guesstimated. In the test phase, fit in value of each input pattern to every subspace is calculated using the pdf of the subspace as the weight of each DAG SVM. Finally, a fusion operation is defined and applied to the DAG SVM outputs to decide the class label of the given input pattern. Evaluation results show the prominence of our method of multi-class classification compared with DAG SVM. Some data sets including synthetic one, the iris, and the wine data sets relative standard DAG SVM, were used for the evaluation.

**Keywords:** DAG SVM, classifier combination, multi-class classification.

## 1 Introduction

Support Vector Machines (SVMs) [1] are very popular and powerful in learning systems because of attending high dimensional data, providing good generalization properties, their ability to classify input patterns with minimized structural misclassification risk and finding the optimal separating hyperplane (OSH) between two classes in the feature space. Moreover, SVMs have many usages in pattern recognition and data mining applications such as text categorization [3, and 4], phoneme recognition [5], 3D object detection [6], image classification [7], bioinformatics [8], and etc.

In spite of all advantages, there are some limitations in using SVMs. In the first place, it is originally formulated for two-class (binary) classification problems and an extension to multi-class problems is not straightforward and unique. DAG SVM [9] is one of the several methods that have been proposed to solve this problem. DAG SVM, in the training stage, determines  $n(n-1)/2$  hyperplanes similar to pairwise SVMs, where  $n$  is the number of classes, and in the testing stage resolves unclassifiable regions problem by using a decision tree. Another problem with the standard SVMs is that the training stage has  $O(m^3)$  time and  $O(m^2)$  space complexities, where  $m$  is the training set size. Operations on large matrices in the training stage, such as calculating their inverse and determinant are highly time-consuming. For this reasons, it is not suitable for large data sets, applications requiring great classification speed, and

when fast real-time response is needed. To overcome these deficiencies, many solutions are proposed such as [10-12].

In this paper, at first, we introduce our Weighted DAG SVM (WDAG SVM) classifier. The input space is divided into several subspaces using a clustering algorithm and then using our training data set a DAG SVM is trained. Later, the class label of every test data is determined by combining all results of classifiers. The main advantage of this method is the increased ability of each SVM for its corresponding subspace. Plus, as will mention later, some parts of the training stage can be performed in parallel and therefore, the training time is decreased. Also, it enables us to use large amount of training data to train SVMs. Other benefits of WDAG SVM include: increased space dimensionality and combination of SVM classifiers to make final decision (in fact we do a voting mechanism in this stage). According to the experimental results, WDAG SVM shows better accuracy in multi-class classification problems in comparison with DAG SVM.

The remainder of this paper is arranged as follows. In section 2, we briefly review binary and multi-class SVM classifiers. Section 3 includes the description of DAG SVM. WDAG SVM is our proposed methods which will be introduced in section 4. In section 5, we show the experimental results. Some conclusions are summarized in section 6.

## 2 Binary and Multi-class SVM

### 2.1 Binary SVM

Let  $d$ -dimensional inputs  $x_i$  ( $i=1, \dots, m$ ,  $m$  is the number of samples) which belong to either class I or class II and their associated labels be  $y_i = 1$  for class I and  $y_i = -1$  for class II, respectively. For linearly separable data, the SVM determines a canonical hyperplane  $f(x) = w^T x + b = 0$ , called also optimal separating hyperplane (OSH), where  $w$  is a  $d$ -dimensional vector and  $b$  is a scalar. It divides the training samples into two classes with maximal margin. For non-linearly separable case, the input samples are mapped into a high-dimensional feature space by a  $\phi$ -function, and then an OSH is trained in this space:

$$f(x) = w^T \phi(x) + b = 0 \quad (1)$$

and the decision function for a test data is:

$$D(x) = \text{sign}(w^T \phi(x) + b) \quad (2)$$

Considering the noise with slack variables  $\xi_i$  and error penalty term  $C \sum_{i=1}^m \xi_i$ , the optimal hyperplane can be found by solving the following quadratic optimization problem:

$$\begin{aligned} \min & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} & y_i(w \cdot \varphi(x_i + b)) \geq 1 - \xi_i \end{aligned} \quad (3)$$

### 2.2 Multi-class SVM

SVM formulation has been originally developed for binary classification problems and finding the direct formulation for multi-class case is not easy and still an ongoing research issue. In two ways we can have a multi-class SVM classifier; one is to directly consider all data in one optimization formulation, and the other is to decompose multi-class problem to several binary problems. The second solution is a better idea and has been considered more than the first approach [13-17] because binary classifiers are easier to implement and moreover some powerful algorithms such as Support Vector Machine (SVM) are inherently binary [18]. Two major decomposition implementations are: "one-against-all" and "one-against-one".

The one-against-all [1] method constructs  $n$  SVMs where  $n$  is the number of classes. The  $i^{\text{th}}$  SVM is trained to separate the  $i^{\text{th}}$  class from remaining classes. The one-against-one [14] (pairwise SVMs) instead, constructs  $n(n-1)/2$  decision functions for all the combinations of class pairs. In determination of a decision function for a class pair, we use the training data for the corresponding two classes. Thus, in each training session, the number of the training data is reduced considerably compared to one-against-all support vector machines, which use all the training data. Experimental results in [19] indicate that the one-against-one is more suitable for practical use. We continue our discussion with focus on the pairwise SVMs method in next section (more details appeared in [20]).

### 3 DAG SVM

A problem with both on-against-all and pairwise support vector machines is unclassifiable regions. In pairwise SVMs, let the decision function for class  $i$  against class  $j$ , with the maximal margin, be:

$$D_{ij}(x) = w_{ij}^T \varphi(x) + b_{ij}, \quad (5)$$

where  $w_{ij}$  is the  $d$ -dimensional vector,  $\varphi(x)$  is a mapping function that maps  $x$  into the  $d$ -dimensional feature space,  $b_{ij}$  is the bias term, and  $D_{ij}(x) = -D_{ji}(x)$ .

The regions  $R_i$  are shown in Figure 1 with labels of class I, II, and III.

$$R_i = \{x | D_{ij}(x) > 0, j = 1, 2, \dots, n, j \neq i\}. \quad (6)$$

If  $x$  is in  $R_i$ , we classify  $x$  into class  $i$ . If  $x$  is not in  $R_i (i = 1, 2, \dots, n)$ ,  $x$  is classified by voting. Namely, for the input vector  $x$ ,  $D_i(x)$  is calculated at follow:

$$D_i(x) = \sum_{i \neq j, j=1}^n \text{sign}(D_{ij}(x)), \quad (7)$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{for } x \geq 0, \\ -1 & \text{for } x < 0, \end{cases} \quad (8)$$

and  $x$  is classified into class:

$$\arg \max_{i=1,2,\dots,n} D_i(x). \quad (9)$$

If  $x \in R_i, D_i(x) = n - 1$  and  $D_k(x) < n - 1$  for  $k \neq i$ . Thus,  $x$  is classified into  $i$ . But if any of  $D_i(x)$  is not  $n - 1$ , (9) may be satisfied for plural  $i$ . In this case,  $x$  is unclassifiable. In the shaded region in Figure 1,  $D_i(x) = 0 (i = 1, 2, \text{ and } 3)$ . Therefore, this region is unclassifiable, although the unclassifiable region is much smaller than that for the one-against-all support vector machine.

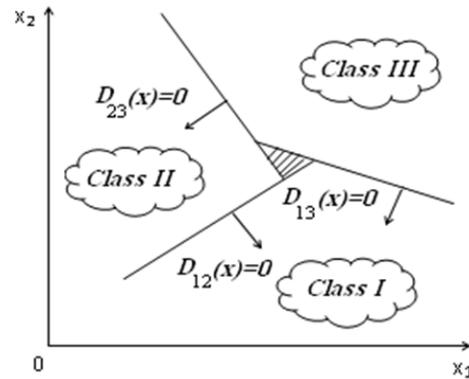


Figure 1: Unclassifiable regions by the pairwise formulation.

In pairwise SVMs, classification reduces the unclassifiable regions that occur for one-against-all support vector machines but it still exists. To resolve this problem, Vapnik [2] proposed to use continuous decision functions. Namely, we classify a datum into the class with maximum value of the decision functions. Inoue and Abe [21] proposed fuzzy support vector machines, in which membership functions are defined using the decision functions. Another popular solution is DAG SVM that uses a decision tree in the testing stage. Training of a DAG is the same as conventional pairwise SVMs. Classification by DAGs is faster than by conventional pairwise SVMs or pairwise fuzzy SVMs. Figure 2 shows the decision tree for the three classes shown in Figure 1. In the figure,  $\bar{i}$  shows that  $x$  does not belong to class  $i$ . As the top-level classification, we can choose any pair of classes. And except for the leaf node if  $D_{ij}(x) \geq 0$ , we consider that  $x$  does not belong to class  $j$ , and if  $D_{ij}(x) < 0$  not class  $i$ . Thus, if  $D_{12}(x) > 0$ ,  $x$  does not belong to class II. Therefore, it belongs to either class I or class III, and the next classification pair is classes I and III.

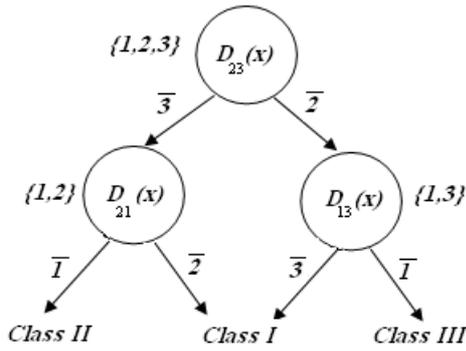


Figure 2: DAG classification.

The generalization regions become as shown in Figure 3. Unclassifiable regions are resolved, but clearly the generalization regions depend on the tree formation.

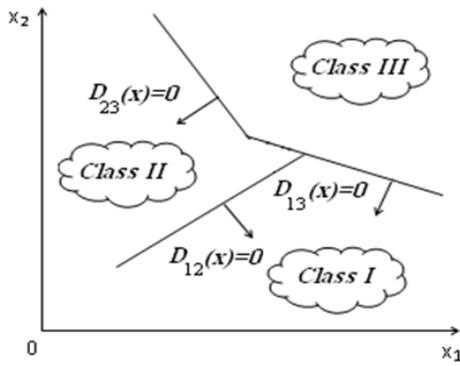


Figure 3: Generalization region by DAG classification.

### 4 The Proposed Method

In this section we propose WDAG SVM. Suppose that we have a large group of data with different class labels that are completely mixed together. Usual methods for this multi-class classification problem such as one-against-all or DAG face some difficulties:

- The kernel matrices that are constructed in the training stage are large and computations on such matrices, for example finding their inverse and determinant, consume tremendous CPU time and takes lots of space.
- The accuracy of classification is low especially when the classes mixed.

WDAG SVM attempts to overcome these limitations by dividing the input space into several subspaces and train a DAG SVM for each of them.

Figure 4 shows the idea of WDAG SVM abstractly. Firstly, we see all training data that are clustered into three regions (the regions are colored red, green, and blue if the paper is printed in color). In each region, a DAG SVM is trained but the pdf (probability density function) of each cluster specifies the significance of each DAG SVM. For each cluster, this important degree is large in the middle but decreases in the periphery, which is shown with blurred color in peripheries. Figure 5 shows details and following sub-sections explain the proposed approach with more details.

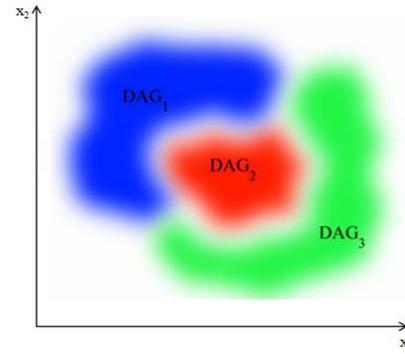


Figure 4: The overall view of WDAG SVM with 3 clusters in 2-D spaces.

#### Description of Training Module

The training stage (Figure 5.a) consists of three levels. In the first level, the training data is divided into N clusters by K-Means algorithm. Then the statistical parameters of each cluster for normal pdf are extracted. These parameters include covariance and mean vectors which are defined as follows:

Suppose that  $X_T = [X_1, X_2, \dots, X_n]^T$  and  $Y_T = [Y_1, Y_2, \dots, Y_n]^T$  are vectors in  $R^d$  space. Also,  $X_i = (x_1, x_2, \dots, x_d)$  and  $Y_i = (y_1, y_2, \dots, y_d)$  are  $i^{th}$  sample ( $i = 1, 2, \dots, n$ ) in  $R^d$ . Therefore:

$$\bar{X}_T = \frac{1}{n} \sum_{i=1}^n X_i, \tag{10}$$

$$\sum_{X_i Y_i} = \frac{1}{d-1} \sum_{i=1}^d (x_i - \bar{X}_T)(y_i - \bar{Y}_T), \tag{11}$$

where  $\bar{X}_T$  is the mean vector with d members, and  $\sum_{X_i Y_i}$  is the covariance matrix (d×d), where d is the number of features for the input data. These parameters are used to assign a Gaussian distribution function (12) to each cluster that is used for weighting procedure in the testing stage:

$$f(x) = \frac{1}{\sqrt{2\pi}|\Sigma|^{\frac{1}{2}}} \exp(-(x - \bar{x})\Sigma^{-1}(x - \bar{x})), \tag{12}$$

where  $\bar{x}$  and  $\Sigma^{-1}$  are the mean vector and the covariance matrix of the  $M^{th}$  cluster. These Gaussians can overlap which enables us to make a fuzzy boundary between clusters. Finally, in each cluster, existent data is trained using DAG SVM in the third level. Training of DAG SVM is similar to one-against-one SVMs.

The outputs of the training stage are a Gaussian distribution function and optimal decision hyperplanes of each cluster that are used in the testing stage. Note that we assumed that the samples are not independent and have use covariance matrix to obtain a more accurate distribution function for the representation of each cluster.

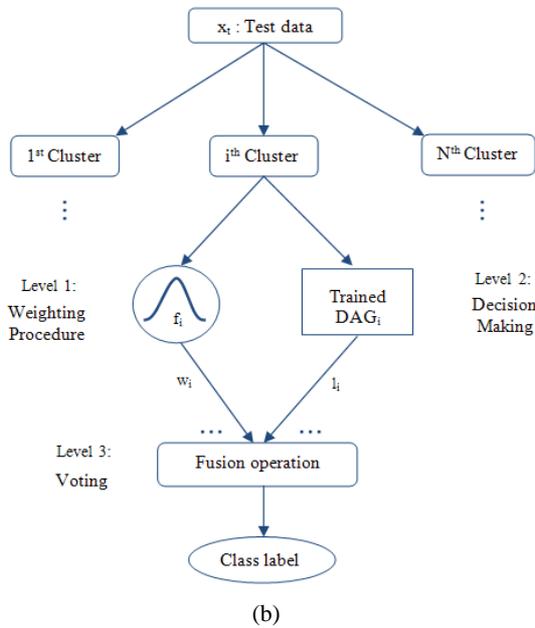
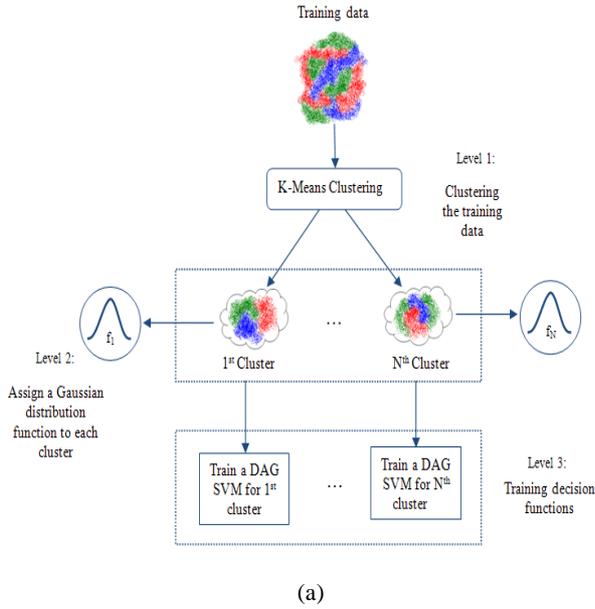


Figure 5: General overview of the proposed WDAG SVM scheme, (a) training module and (b) testing module.

**Description of Testing Module**

Testing module includes basic levels such as weighing procedure (WP), decision making, and voting which are shown in Figure 5.b and discussed separately as follows:

**Level 1: Weighting Procedure (WP)**

In this level, the membership degree of each testing data to each cluster is calculated using Gaussian distribution functions which have been achieved in the previous section. Let  $\{x_t\}_{t=1}^K \subset R^n$  be a set of  $n$ -dimensional test samples. Membership degree of  $x_t$  in relation to  $M$ th cluster ( $M = 1, 2, \dots, N$ ) according to (12) is given as:

$$w_M = \frac{1}{\sqrt{2\pi|\Sigma|^2}} \exp\left(-\frac{1}{2}(x_t - \bar{x})\Sigma^{-1}(x_t - \bar{x})\right). \quad (13)$$

Note that the output of the weighting procedure is  $w_M$  where  $M=1, \dots, N$  and  $N$  is the number of clusters.

**Level 2: Decision Making**

The label of  $x_t$  (where  $x_t$  is a test data) is calculated using majority-based decision obtained from DAG SVM which is trained for the  $M^{\text{th}}$  cluster.

Let the decision function for class  $i$  against class  $j$ , for the  $M^{\text{th}}$  cluster with the maximal margin, be:

$$D_{ij}^M(x_t) = w_{ij}^T(x_t) + b_{ij}, \quad (14)$$

and in each cluster we follow the corresponding DAG SVM structure for determining class label of  $x_t$ . At the end of this stage, each cluster returns a local class label  $l_i$  ( $i=1, \dots, N$ ), where  $N$  is the number of clusters. Voting is the final stage that determines the global class label of  $x_t$  considering clusters judgment and membership degrees.

**Level 3: Voting**

Output of this level is achieved by the fusion operation. The combining procedure can improve classification rate.

Suppose that  $L = \{l_1, l_2, \dots, l_N\}$  is set of labels for a given test sample  $x_t$  that are determined by  $N$  DAG SVM, also  $W = \{w_1, w_2, \dots, w_N\}$  is set of membership degrees of a same test sample  $x_t$  to each cluster, where  $N$  is the number of clusters. Finally, output is achieved by:

$$\text{Class Label} = \arg(\max_{i=1..n} \{\tau_i\}), \quad (15)$$

where

$$\tau_i = \sum_{j \in \Omega_i} w_j, \quad i = 1, \dots, n \quad (16)$$

and

$$\Omega_i = \{j | l_j = i\}, \quad j = 1, \dots, N, \quad (17)$$

where  $n$  is the number of classes.  $\Omega_i$  is set of labels which mention to class  $i$  and  $\tau_i$  is sum of their weights. Equation (15) demonstrate that  $x_t$  is assigned to a class with maximum  $\tau_i$ . In the other words, two following points are required to classification of input samples:

- Number of labels assigned to each class
- Weight (degree of importance that is achieved from WP) of each classifier that participates in voting procedure.

Therefore, if a class is mentioned with a large number of DAG SVM and also weights of these classifiers be high enough, then the test sample is assigned to it.

**5 Experimental Evaluation**

The proposed method has been evaluated over a series of synthetic data. In Table 1, the accuracy of DAG SVM and WDAG SVM classifiers for a series of synthetic data has been compared. The number of synthetic data is 150 (50 data in each class). Four types of data are considered. In data type 1, separate data is used and gradually in data set 2 to 4 classes are interlaced (shown in Figure 6). As shown in Table 1, comparing the two methods, WDAG SVM gives better results in each level of interlacing.

The power of our method is apparent especially when the data are completely mixed. Also our proposed method is evaluated for iris and wine data sets and its results are summarized in Table 2. According to Table 2 it can be seen that the accuracy of WDAG SVM classifier for both noisy and without noise iris data set and also wine data set is more than the DAG SVM.

It is mentionable that choosing the number of clusters is a trade-off. This value must be chosen in a way that in each cluster, there are data points from whole of the classes. We tested different values for N (the number of clusters) in our experiments and we found constant recognition rate for N=3 and greater.

We also claim that our method is more efficient in execution time. Table 3 shows the learning stage times for synthesise, iris, and wine data sets. The number of data points in Data 1 data set is 1500 (500 data points for each class) and in Data 2 data set is 6000 (2000 data points for each class).

Table 1: Experimental results of DAG SVM and WDAG SVM in 2-dimensional spaces (N=3)

Number of testing data	Recognition rate of DAG SVM	Recognition rate of WDAG SVM
40 sample of Data1	100	100
40 sample of Data2	96	100
40 sample of Data3	96	100
40 sample of Data4	69	98

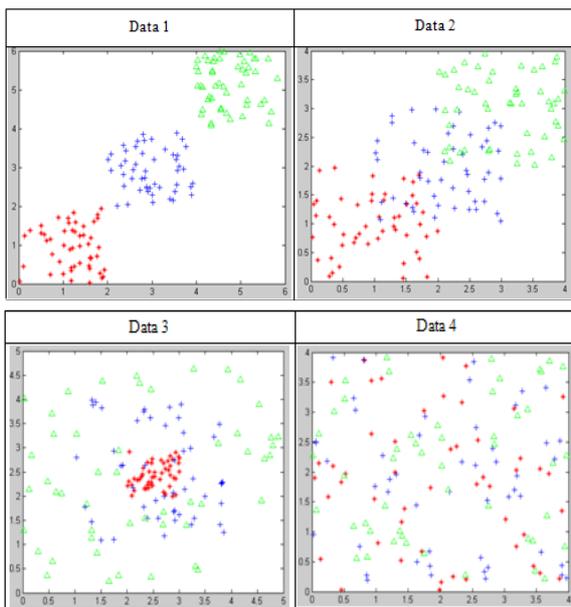


Figure 6: Synthetic data for experiment

## 6 Conclusion

In this paper, we proposed a weighted DAG support vector machine algorithm for multi-class classification problems. The important contributions of the proposed algorithm can be summarized as follows:

- Dividing input space to subspaces (space linearization)
- Applying multi expert system instead of one classifier
- Using the power of classifier fusion in the mixing results

After dividing of samples using the clustering algorithm, for each part a DAG SVM is trained and weighting procedure helps to fusion multi-trained SVMs. Captured results over synthetic sample dataset showed mixed data could be classified with high precisions. The outcome recommends applying the WDAG SVM (the proposed method) to low signal to noise ratio environment and fused samples. Also we applied the WDAG SVM to the standard iris and wine data sets and we encountered up to 17% recognition rate over DAG SVM.

Plus, we claim that the proposed method is more efficient in required time and space because the train matrices in each cluster are smaller and computation on them is faster. Moreover, data in each cluster is independent from others and hence we can perform this algorithm on parallel machines with good performance and high scalability. Finally, it is suitable for large data sets.

It is mentionable that determining a fusion operation plays an important role in the algorithm's results so it must be defined carefully. We used a simple fusion operation to judge on the votes of clusters. One future work can be finding better and more precise fusion operations.

Table 2: Experimental results of DAG SVM and WDAG SVM on Iris and Wine data sets (N=3)

Data set	Number of training data	Number of testing data	Recognition rate of DAG SVM	Recognition rate of WDAG SVM
Iris	120	30(without noise)	100	100
	120	30+%10 noise	95	98.33
	120	30+%20 noise	91.67	95.33
	90	60(without noise)	93.67	98.33
	90	60+%10 noise	80.67	90.33
	90	60+%20 noise	78.33	83.67
Wine	122	56	82.14	98.21

Table3: Comparison of the training time for DAG SVM and WDAG SVM

Type of data	Number of training data	Elapsed time for training of WDAG SVM (seconds)	
		DAG SVM	WDAG SVM
Data1	1050	4.11	1.13
Data2	4500	201.4	23.84
Iris	120	2.89	1.35
Wine	122	2.34	0.64

## 7 References

- [1] Vapnik, V., "The Nature of Statistical Learning Theory", New York: Springer-Verlag, 1995.
- [2] Vapnik, V., "Statistical Learning Theory", John Wiley & Sons, New York, NY, 1998.
- [3] Joachims, T., "Text categorization with support vector machines: Learning with many relevant features", Technical report, University of Dortmund, 1997.
- [4] Wang, T.-Y., Chiang, H.-M., "Fuzzy support vector machine for multi-class text categorization", *Information Process and Management*, 43, 914–929, 2007.
- [5] Salomon, J., "Support vector machines for phoneme classification", M.Sc Thesis, University of Edinburgh, 2001.
- [6] Pontil, M., Verri, A., "Support Vector Machines for 3D Object Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 6, 1998.
- [7] Takeuchi, K., Collier, N., "Bio-Medical Entity Extraction using Support Vector Machines", *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, 57-64, 2003.
- [8] Foody, M.G., Mathur, A., "A Relative Evaluation of Multiclass Image Classification by Support Vector Machines", *IEEE Transactions on Geoscience and Remote Sensing*, 42, 1335–1343, 2004.
- [9] Platt, J., Cristianini, N., Shawe-Taylor, J., "Large margin DAGs for multiclass classification", *Advances in Neural Information Processing Systems 12*. MIT Press, 543–557, 2000.
- [10] Tsang, I., Kwok, J., Cheung, P., "Core vector machines: fast SVM training on very large data sets", *J. Mach. Learn. Res.* 6, 363-392, 2005.
- [11] Joachims, T., "Training linear SVMs in linear time", in *proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pp. 217-226, 2006.
- [12] Keerthi, S., Chapelle, O., DeCoste, D., "Building Support Vector Machines with Reduced Classifier Complexity", *J. Mach. Learn. Res.* 7, 1493-1515, 2006.
- [13] Wu, T.F., Lin, C.J., Weng, R.C., "Probability estimates for multi-class classification by pairwise coupling", *J. Mach. Learn. Res.* 5, 975–1005, 2004.
- [14] Hastie, T., Tibshirani, R., "Classification by pairwise coupling", *Ann. Stat.* 26 (2), 451–471, 1998.
- [15] Allwein, E.L., Schapire, R.E., Singer, Y., "Reducing multiclass to binary: a unifying approach for margin classifiers", *J. Mach. Learn. Res.* 1, 113–141, 2000.
- [16] Rennie, J.D.M., Rifkin, R., "Improving multiclass text classification with the support vector machine", MIT AI Memo 026, 2001.
- [17] Knerr, S., Personnaz, L., Dreyfus, G., "Single-layer learning revisited: a stepwise procedure for building and training a neural network". In J. Fogelman, editor, *Neurocomputing: Algorithms, Architecture and Applications*. Springer-Verlag, 1990.
- [18] Zhou, J., Peng, H., Suen, C.Y., "Data-driven decomposition for multi-class classification", *Pattern Recognition*, Volume 41, Issue 1, 67-76, 2008.
- [19] Hsu, C.-W., Lin, C.-J., "A comparison of methods for multiclass support vector machines", *IEEE Trans. Neural Networks* 13(2), 415-425, 2002.
- [20] Abe, S., Singh, S., "Support Vector Machine for Pattern Classification", Springer-Verlag London Limited, 2005.
- [21] Inoue, T., Abe, S., "Fuzzy support vector machines for pattern classification", in *proceedings of International Joint Conference on Neural Networks*, volume 2, 1449-1454, 2001.