ENHANCING THE PERFORMANCE OF SVM CLASSIFICATION BASED ON FISHER LINEAR DISCRIMINANT

B.Sunitha[1], Dr.M.Seetha[2]

[1]*Computer Science & Engineering, G. Narayanamma Institute of Technology & Science (for Women), Hyderabad, India*
[2]*Professor, Dept.of CSE, GNITS, Hyderabad*

Abstract

Support Vector Machine (SVM) is one of the important classification method used in many areas. Normal support vector machine is not suitable for classification of large data sets due to its high training complexity. Training of SVM with data number n has time complexity between $O(n^2)$ and $O(n^3)$. This paper introduces a novel data reduction method Fisher's decision tree for SVM classification. It is a classifier uses the dimensionality reduction of Fisher Linear Discriminant and decomposition strategy of decision trees. The proposed classifier has distinctive advantages on dealing with large data sets.

Keywords: *SVM, decision tree, Fisher Linear Discriminant*

----------***----------

1. INTRODUCTION

Classification is an important task in pattern recognition, machine learning and data mining. It consists in predicting the category, class or label of previously unseen objects. Methods that implement this task are known as classifiers.

Support vector machine (SVM) is a highly desirable classification method and it is a well known classifier due to its excellent classification accuracy, generalization and compact model. It offers a hyper plane that represents the largest separation (or margin) between the two classes [4]. It minimizes the empirical classification error and maximizes the geometric margin. However this kind of maximum-margin hyper plane may not exist because of class overlapping or mislabeled examples. In the linearly separable case the hyper plane is easy to compute, however in the general case it is necessary to use a soft margin SVM by introducing slack variables (Vapnik 1995). In this way it is possible to find a hyper plane that splits the examples as cleanly as possible. In order to find a separation hyper plane, it is necessary to solve a Quadratic Programming Problem (QPP). This is computational expensive, it has $O(n3)$ time and $O(n2)$ space complexities with n data [3]. So the standard SVM is unfeasible for large data sets [21].

Many researchers have tried to find possible methods to apply SVM classification for large data sets. These methods can be divided into four types: a) reducing training data sets (data selection) [15], b) using geometric properties of SVM [1], c) modifying SVM classifiers [25], d) decomposition [21] e) other methods. The data selection method chooses the objects which are possible to be the support vectors (SV). These data are used for SVM training. Generally, the number of the support vectors is much small compared with the whole data. Clustering is another effective tool to reduce data set size, for example, hierarchical clustering [16] and parallel clustering [20]. The geometric properties of SVM (Franc and Hlavac 2003) can also be used to reduce the training data. In separable case, the maximum-margin hyper plane is equivalent to finding the nearest neighbors in the convex hulls of each class [29]. The random sampling method [15] is simple and commonly used for large data sets. However it needs to be applied several times and the obtained results are not repeatable.

Data selection methods choose objects which are support vectors (SV). These data are used for training the SVM classifier. Generally, the number of the support vectors is a small subset of the whole data [26]. The goals of this type of method are: a) fast detection of support vectors (SV) which define the optimal separating hyper plane, b) remove the data which are impossible to be SVs, c) obtain similar accuracy using the reduced data set. In this project data selection technique is used to train SVM with large data sets.

The Decision Tree (DT) classifier is a method that has been used as a preprocessing step for SVM in recent years. Generally, the classification accuracy of SVM is better than DT. The training time of SVM is longer than DT for large data sets. Combination of SVM and DT can overcome two shortcomings of SVM: computational burden and multi-class classification. A DT can be learned or induced by splitting the input space into subsets, based on an attribute value test. This process is repeated for each derived subset in a recursive manner [2]. DT has some advantages over other methods such as to be tolerant to noise, support missing values and be able to produce models easily interpreted by human beings. In addition, the induction of DT is not costly. The general algorithm to induce decision trees from data works separating

or splitting the data recursively, in such way that partitions are increasingly purer up to certain criterion is satisfied.

Generally there are two main problems in SVM classification by using DT: SVM has to be used to compute the margins at each leaf of DT which increases computation cost [17]; the classification accuracy is poor when DT is not big enough [22], or the data are imbalance [28]. To improve the classification accuracy, the separability (margin measures) information is inserted in [5]. To find a linear combination of features for separating two or more classes of object a component analysis algorithm Fisher Linear discriminant (FLD) is used. In fact, SVM can be regarded as a way to sparsity FLD [24]. FLD has been successfully applied in feature selection [27] and face recognition. The geometric properties of SVM can also be used to reduce the training data. In the linearly separable case, the maximum-margin hyper plane is equivalent to finding the closest pair of points in the convex hulls [30]. Neighborhood properties of the objects can be applied to detect SV and to improve classification accuracy of SVM. SVM with FLD uses the geometric properties of SVM. It is difficult to apply geometric properties in high-dimension and multiclass classification. While DT can overcome this disadvantage of SVM+FLD.

In this paper proposed a novel data reduction method for SVM, it uses DT and Fishers Linear Discriminant (FLD). FLD is used to select the part of data in each partition generated by DT. The combination of DT and FLD is nothing but a Fisher's decision Tree. Fisher's Decision Tree makes it possible for SVM to classify large data sets.

2. CLASSIFICATION METHODS

2.1 Support Vector Machine

Basically SVM classification can be grouped into two types: linearly separable and linearly inseparable cases. The nonlinear separation can be transformed into linear case via a kernel mapping. In the linear inseparable case, the convex hulls intersect. The convex hull based methods do not work well for SVM, because SVs are generally located on the exterior boundaries of data distribution. On the other hand, the vertices of the convex-concave hull are the border points and they are possible to be the support vectors [23]. Support Vector Machines (SVM) is based on statistical learning theory developed by Vapnik. It classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space.

The training set X is given as

$$X = \{x_i, y_i\}_{i=1}^{n} \qquad (1)$$

Where $x_i \in R^d$, $y_i \in (1.....C_L)$. The C_L is the number of classes. SVM classifies data sets with an optimal separating hyper plane, which is given by

$$\omega^T \varphi(x_i) + b \qquad (2)$$

This hyper plane is obtained by solving the following quadratic programming problem

$$min_{\omega,b} J(\omega) = \frac{1}{2}\omega\omega^T + C \sum_{i=1}^{n} \varepsilon_i \quad (3)$$

Such that: $y_i [\omega^T \varphi(x_i + b] \geq 1 - \varepsilon_i$

Where $\boldsymbol{\varepsilon_i > 0, i = 1 \ldots \ldots n}$, are the slack variables, to tolerate mis-classifications. C>0 is a regularization parameter. (3) is equivalent to the following dual problem with the Lagrange multipliers $\propto_i > 0$

$$\boldsymbol{max_{\propto i} J(\omega) = -\frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i y_i \alpha_i y_i K \langle x_i . x_j \rangle + \sum_{i=1}^{n} \propto_i} \qquad (4)$$

Such that: $\sum_{i=1}^{n} \propto_i y_i = 0, C \geq \propto_i \geq 0, i = 1,2, \ldots, n$

With $C > 0, \propto_i \geq 0, i = 1,2, \ldots, n$, the coefficients corresponding to x_i. All x_i with nonzero $\propto_i$ are called support vectors. The function K is the kernel which must satisfy the Mercer condition [4]. The resulting optimal decision function is

$$\boldsymbol{y_i \, sign\left(\sum_{i=1}^{n} \propto_i y_i K \langle x_i . x_j \rangle \; b\right)} \qquad (5)$$

Where $x = [x_1, x_2, \ldots\ldots\ldots x_n]$ is the input data, is the input data, $\propto_i \; and \; y_i$ are Lagrange multipliers.

A previously unseen sample x can be classified by (5). There is a Lagrangian multiplier α for each training point. When the maximum margin of the hyper plane is found, only the closed points to the hyper plane satisfy α > 0. These points are called support vectors (SV), the other points satisfy α = 0. So the solution is sparse. Here b is determined by Karush- Kuhn-Tucker conditions:

$$\boldsymbol{\frac{\partial L}{\partial \omega} = 0, \quad \omega = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)}$$
$$\boldsymbol{\frac{\partial L}{\partial b} = 0, \quad \sum_{i=1}^{n} \propto_i y_i = 0} \qquad (6)$$

$$\boldsymbol{\frac{\partial L}{\partial \varepsilon_i} = 0, \quad \alpha_i - c \geq 0}$$
$$\boldsymbol{\propto_i \{y_i[\omega^T \varphi(x_i) + b] \geq 1 - \varepsilon_i\} = 0}$$

The basic idea behind support vector machine is illustrated with the example shown in Figure 1.

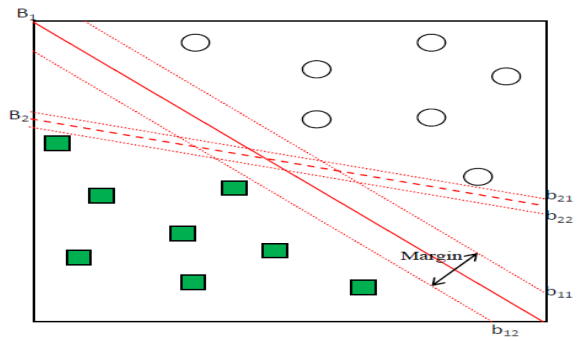

Fig 1: An example of a two separating hyper planes B1 and B2

In this example the data is assumed to be linearly separable. Therefore, there exists a linear hyper plane (or decision boundary) that separates the points into two different classes. In the two-dimensional case, the hyper plane is simply a straight line. In principle, there are infinitely many hyper planes that can separate the training data. Figure 1 shows two such hyper planes, *B1* and *B2*. Both hyper planes can divide the training examples into their respective classes without committing any misclassification errors.

Although the training time of even the fastest SVMs can be extremely slow, they are highly accurate, owing to their ability to model complex nonlinear decision boundaries. They are much less prone to over fitting than other methods.

2.2 Decision Tree

Decision tree is a classifier method able to produce models that can be comprehensible by human experts (Breiman et al., 1984). Among the advantages of decision trees over other classification methods are (Cios, Pedrycz, Swiniarski, & Kurgan, 2007): robustness to noise, ability to deal with redundant attributes, generalization ability via post-pruning strategy and a low computational cost, this last is more notorious when decision trees are trained using data sets with nominal attributes.

When a data set X is large, the computational burden of (4) is heavy. We first use decision tree to separate X into several subsets. A decision tree is a classifier whose model resembles a tree structure T, this structure is built from a labeled data set X. A decision tree is composed of nodes, and edges that connect the nodes. There are two type nodes: internal and terminal. An internal node has branches to connect to other ones, called its sons. A terminal node does not have any sons. The terminal node is called a leaf L of T.

In order to explain the general training process of decision trees on numeric attributes, let's represent the training data sets as in (7).

$$X = \{(x_i, y_i), i = 1, \ldots . M\} \quad (7)$$

Such that: $x_i \epsilon R^d, y_i \epsilon \{C_1, \ldots \ldots, C_L\}$

With
X Training set,
x_i A vector that represents an example or object in X,
y_i The category or class of x_i,
M Number of examples in training set,
d Number of features,
L Number of classes.

The general methodology to build a decision tree is as follows: beginning from root node (it contains X), split the data into two or more smaller disjoint subsets, each subset should contain all or most of its elements with the same class, however this is not necessary. By partitioning input data into pure regions with respect to certain measurement of the impurity. The measurements are defined in terms of instance distribution of the input splitting region. The measurements of impurities followed in this paper are entropy, misclassifications and Gini (8).

$$Entropy(t) = -\sum_{i=0}^{C_L - 1} p\left(\frac{i}{t}\right) log_2 p\left(\frac{i}{t}\right) \quad (8)$$

$$Classification\ error(t) = 1 - max\left[p\left(\frac{i}{t}\right)\right]$$

$$Gini = 1 - \sum_{i-0}^{C_L - 1}\left[p\left(\frac{i}{t}\right)\right]^2$$

Where $C_{L\ is}$ the number of classes, $p\left(\frac{i}{t}\right)$ is the probability of example i to be of the class t, which is the number of examples in the class t divided by the size of the set X, i.e., $p\left(\frac{i}{t}\right) = \frac{|y_i = t|}{|x|}$.

$att_i < split\ point_i$ is used to create partitions in the input space, where att_i is attribute of training set X, $split\ point_i$ is a value of the same type with the attribute i.

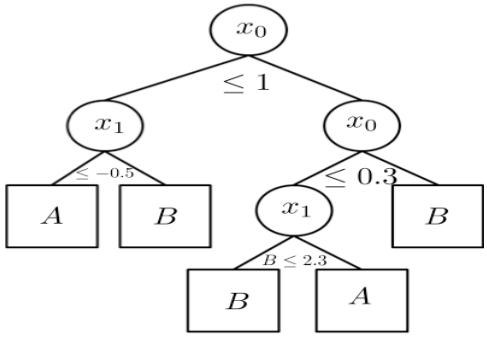

Fig 2: Example of a decision tree

2.3 Fisher Linear Discriminant

Based on geometric properties of SVM, Fisher Linear Discriminant is used to find out the most possible data, it is used to find a linear combination of features which separates two or more classes of objects, which will be used for SVM training. Fisher Linear Discriminant can be considered as a method for linear dimensionality reduction. The method is based on minimizing the projected class overlapping that maximizes the distance between class means while minimizing the variance within each class.
Consider a training data set as (7)

$$X = \{(x_i, y_i), i = 1, \ldots . M\}$$

Such that: $x_i \epsilon R^d, y_i \epsilon \{C_1, \ldots \ldots, C_L\}$

Let's separate X into two subset X^+ and X^- as

$$X^+ = \{x_i \epsilon X \; s.t \; y_i = c_1\}$$
$$X^- = \{x_i \epsilon X \; s.t \; y_i = c_{12}\} \qquad (9)$$

FLD searches for a vector ω that maximizes the separation between the means of X+ and X−, meanwhile minimizes their scattering. In order to measure the scattering of X+ and X−, the scatter $\mu^{\pm}$ for projected objects on ω is defined as

$$\mu^{\pm} = \frac{1}{|X^{\pm}|} \sum_{i=1}^{|X^{\pm}|} X_i \, , X \in X^{\pm} \qquad (10)$$

With
$X^{\pm}$ Represent x^+ or x^-
$\mu^{\pm}$ Represent either $\mu^+ \in R^d \; or \; \mu^- \in R^d$

Let be $\omega \in R^d$ a vector used to project every example in X, then $x_i = \omega^T x$

A problem with Fisher Linear Discriminant occurs when data distribution is multi-modal and when there exist overlapping between classes, under these situations the vector x is not enough to clearly discriminate between classes (Li, 2005), this can be though as a weak learning algorithm (Schapire, 1990). Based on the recursive strategy of decision tree in this paper proposed a stronger classifier DT with FLD known as Fisher's Decision Tree.

An example of a Fisher'sLlinear Discriminant is shown in figure 3.

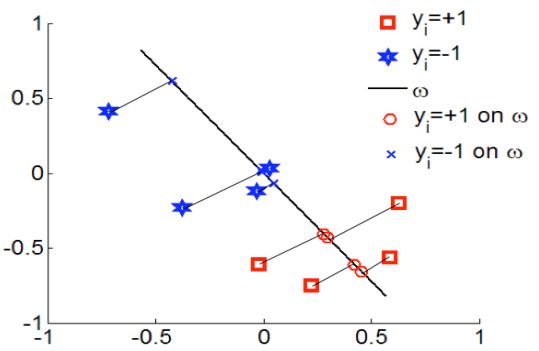

Fig 3: Example of Fisher's linear discriminant in two dimensions, the black line is vector ω

2.4 Fisher's Decision Tree

The core of a proposed method is based on a combination of DT and FLD. The Fisher's decision tree is a two class classifier that takes advantage of dimensionality reduction of Fisher's linear discriminant, which is able of perfectly classify objects that belong to data sets that are linearly separable. Because most real world data sets are not linearly separable, the method follows a decision tree's philosophy splitting the data, recursively. The splits are produced using only one artificial attribute.

The projections of objects x_i on the artificial attribute x (11) are used to create the possible splitting points; each split point is between two consecutive projections.

$$p_i = \frac{\omega^T x_i}{\|\omega\|} \omega, i = 1, \ldots \ldots, N-1 \qquad (11)$$

$$p_{test} = \frac{\omega^T (p_i + p_{i+1})}{2\|\omega\|} \omega \qquad (12)$$

The splitting point p_{test} is used to create two partitions and then an impurity measure is used. The number of tests that must be realized at each node of the tree is N-1. Each p_{test} creates a split which separates the data similar to a linear decision boundary, this is completely determined: it is orthogonal to vector ω and passes through the point (12).

The main difference between the Fisher's tree and decision tree is in how the splits are created and instead of testing every attribute only the artificial attribute is used. It seems that Fisher's tree suffers from repetition, which is the phenomenon that occurs when an attribute is repeatedly tested along a given branch of the tree, this is different because there is only one attribute to split.

3 PROPOSED METHOD: SVM WITH FISHER'S DECISION TREE

According to the geometric properties of SVM, the separating hyper plane is determined by the support vectors which are a small subset of whole data. The support vectors are close to their opposite class i.e., support vectors are near to the boundaries. The aim of Fisher's decision tree is to find the data which are near to the support vectors.

The proposed data selection method i.e., selecting the reduced data through Fisher's decision tree and train those data with support vector machine.

The proposed method works as follows:
1. Select the large Data Set
2. Train a DT using whole data set. Use C4.5
3. Apply Fisher Linear Discriminant for separating the two or more classes of objects
4. Recover all the leaves of DT, these are treated as clusters with low entropy
5. Select some of the examples from those selected clusters based on impurity measures (entropy, Gini, mis-classification)
6. Train SVM using the selected examples.

Otherwise use Fisher's decision tree instead of using step 2 & 3.

The proposed classification strategy can be described in the following steps:
1. Discover all regions that contain all or most of their examples with the same label. This label is the majority class for that region.
2. Determine all its adjacent or neighbor regions whose majority class is opposite. Because in this detecting data that are located near to the opposite label.
3. Search the data with shorter distances. Use FLD to each pair of adjacent low entropy regions.
It is important to notice that high entropy regions will contain support vectors; they do not need to be analyzed with FLD in step 3.

4. COMPARISON OF DIFFERENT APPROACHES

In order to test the effectiveness of decision tree, recently it was tested on iris data set. It has characteristics as size 100, dimensionality 4, numeric features and it has 2 classes of data points. Here C4.5 decision tree is used, those results are placed (Asdrubal Lopez Chau 2012).

This paper presents different approaches used for Classification of large data sets using support vector machines. First, generally in recent years DT is used preprocessing step for SVM. In each disjoint region (partition) discovered by a decision tree is used to train a SVM. In general regions found by a DT are less complicated than the region occupied by the entire training set. A SVM is applied to each region; the computational cost is less expensive compared with training a SVM with the whole data set.

Second, data filter algorithm, which implements a heuristic searching method to obtain the relevant data points from the whole data set. It applies SVM on a training data set in order to obtain a sketch the SV, and then obtains a reduced data set filtering data points that are far from sketch of the SV. Next, it uses a DT in order to classify data points that are near of SV and then filters less important data points from the original data set.

Third, convex-concave hull is used to classify large data sets and then trains SVM. Before convex-concave hull it uses grid processing is used to detect optimal partitions. Next convex hull is used to find extreme points. Then Jarvis march method to determine the concave hull for the inseparable points. Finally the vertices of the convex-concave hull are applied for SVM training.

Fourth, Fisher's decision tree is a two class classifier that takes advantage of dimensionality reduction of Fisher's Linear Discriminant, which is able of perfectly classify objects that belong to data sets that are linearly separable. Most real world data sets are not linearly not separable; Fisher's decision tree is followed the philosophy of DT splitting the data recursively. But the splits are produced using only one artificial attribute. The different approaches used for classifying large data sets to train SVM are shown in table 1.

Mostly large data sets are used to implement this proposed method, because for small data sets FDT unable to find actual support vectors. For large data sets FDTSVM will get classification accuracy are almost the same or even better than the other SVMs because soft margin method is used to deal with misleading points.

5. CONCLUSIONS

This paper introduces a novel data reduction method for SVM classification (FDTSVM). FDT uses an artificial attribute created with the vector computed by the Fisher's Linear Discriminant, and the objects in training data set are projected on it. It takes an advantage of the dimensionality reduction of Fisher's linear analysis and uses the decomposition strategy of the decision trees. The key point of this method is to find the low entropy regions and differentiates the opposite class regions which are closed to decision boundaries. It is ascertained that the SVM classification based on fisher discriminant outperforms the classification techniques.

Table 1: Different data reduction methods used as preprocessing step for SVM

S No.	Approach	Pros	Cons	Applications	Data Sets
1	Decision Tree	Robustness to noise, ability to deal with redundant attributes	Duplication can occur	Agriculture, Biomedical Engineering, Medical diagnosis, Text processing	For large data sets it gets low frequency. Mostly suitable for small data sets (iris.arff)
2	Data Filtering	Helps to process data and improves processing time	Filtering is a prerequisite for classification	Signal processing, for filtering spam mails	Useful for text data and voice data
3	Convex-Concave Hull	For large data sets classification accuracy is very high	Classification accuracy becomes lower when there are inseparable points	Edge detection, detecting collisions in games development	More suitable for large Data sets (spam base.arff)
4	Decision Tree & Fisher Linear Discriminant	Generalization ability via post-pruning strategy and low computational cost	Overlapping will exist between classes	Text processing, Financial analysis, Face Recognition,	Public available data sets are classified efficiently (Ionosphere.arff)
5	Fisher's Decision Tree	Perfectly classify objects that belong to data sets that are linearly separable	Little bit difficult to find low entropy regions	Remote sensing, Molecular Biology, Data mining, Object Recognition	Most real world data sets are not linearly separable, FDT is applied for real world data sets like (ijcnn1.arff)

REFERENCES

[1]. Asdrubal Lopez Chau, Xiaoou Li, Wen Yu, Data Selection Using Decision Tree for SVM Classification, 2012 IEEE 24 International conference on Tools with Artificial Intelligence

[2]. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Wadsworth, 1984.

[3]. C. M. Bishop, Pattern Recognition and Machine Learning, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[4]. N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods,CambridgeUniversity Press, Mar. 2000.

[5]. J. Chen, C. Wang, and R. Wang, Combining support vector machines with a pairwise decision tree, IEEE Geoscience and Remote Sensing Letters, vol. 5, no. 3, pp. 409 –413, july 2008.

[6]. C. Chih-Chung and L. Chih-Jen, Libsvm: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 1–27, 2011.

[7]. R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training SVM. Journal of Machine Learning Research, vol.6, 1889-1918, 2005

[8]. F. Chang, C.-Y. Guo, X.-R. Lin, and C.-J. Lu, Tree decomposition for large-scale svm problems,J. Mach. Learn. Res., vol. 9999, pp. 2935-2972, December 2010.

[9]. R.Collobert, S.Bengio, SVMTorch: Support vector machines for large regression problems, Journal of Machine Learning Research, Vol.1, 143- 160, 2001.

[10]. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification (2nd Edition). Wiley-Interscience, 2000.

[11]. B. Fei and J. Liu, Binary tree of svm: A new fast multiclass training and classification algorithm, IEEE Transactions on Neural Networks, vol. 17, no. 3, pp. 696 -704, may 2006.

[12]. T.Ho and E. Kleinberg, Checkerboard data set, http://www.cs.wisc.edu/math-prog/mpml.html, 1996.

[13]. S. Katagiri and S. Abe, "Selecting support vector candidates for incremental training," in Systems, Man and Cybernetics, 2005 IEEE International Conference, vol. 2, oct. 2005, pp. 1258–1263.

[14]. M.A.Kumar and M. Gopal, "A hybrid svm based decision tree," Pattern Recogn., vol. 43, no. 12, pp. 3977–3987, Dec. 2010

[15]. Y. Lee and O. L. Mangasarian, Rsvm: Reduced support vector machines, Data Mining Institute, Computer Sciences Department, University of Wisconsin, 2001, pp. 100–107.

[16]. X.Li, J.Cervantes, W.Yu, Fast Classification for Large Data Sets via Random Selection Clustering and Support Vector Machines, Intelligent Data Analysis, Vol.16, No.6, 897-914, 2012

[17]. M. Lu, C. L. P. Chen, J. Huo, and X. Wang, Multi stage decision tree based on inter-class and inner-class margin of svm, Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics, Piscataway, NJ, USA: IEEE Press, 2009, pp. 1875–1880.

[18]. G. Fung and O. L.Mangasarian, Proximal Support Vector Machine Classifiers, Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA, 2001, pp. 77–86

[19]. Michael E. Mavroforakis and Margaritis Sdralis and Sergios Theodoridis, A Geometric Nearest Point Algorithm for the Efficient Solution of the SVM Classification Task, IEEE Transactions on Neural Networks, Vol. 18, 2007, pp. 1545–1549

[20]. C.Pizzuti, D.Talia, P-Auto Class: Scalable Parallel Clustering for Mining Large Data Sets, IEEE Trans. Knowledge and Data Eng., vol.15, no.3, pp.629-641, 2003.

[21]. J. Platt, Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods: Support Vector Machines, pp. pp. 185–208, 1998.

[22]. J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.

[23]. A.Moreira, M.Y.Santos, Concave Hull: A K- earest Neighbours Approach for the Computation of the Region Occupied by a Set of Points, *GRAPP (GM/R)*, Pages 61–68, 2007

[24]. A. Shashua, "On the relationship between the support vector machine for classification and sparsified fisher's linear discriminant," Neural Process. Lett., vol. 9, no. 2, pp. 129–139, Apr. 1999.

[25]. J.A.K.Suykens , J.Vandewalle , Least squares support vector machine classifiers, Neural Processing Letters, vol. 9, no. 3, Jun. 1999, pp. 293– 300.

[26]. D. M. J. Tax and R. P. W. Duin, "Support vector data description," Mach. Learn., vol. 54, no. 1, pp. 45–66, Jan. 2004.

[27]. E.Youn, .Koenig, M. K.Jeong, .H. Baek , Support vector-based feature selection using Fisher's linear discriminant and Support Vector Machine, Expert Systems with Applications, Volume 37 Issue 9, 6148-6156, 2010

[28]. H. ZHAO, Y. YAO, AND Z. LIU, A classification method based on non-linear svm decision tree, in Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery , 2007, pp. 635– 638.

[29]. K.P. Bennett , E.J. Bredensteiner, Duality and Geometry in SVM Classifiers, *17th International Conference on Machine Learning,* San Francisco, CA, 2000

[30]. M.Berg, O.Cheong, M.Kreveld, M.Overmars, Computational Geometry: Algorithms and applications, Springer-Verlag, 2008