International Journal Of Computer Science And Applications Vol. 1, No. 3, December 2008
ISSN 0974-1003

Multi-class SVM Classifier With Neural Network For Handwritten Character Recognition

Prof. Shubhangi D C
Appa IET , Gulbarga,
Karnataka,India
09448716838
shubhangidc@yahoo.co.in
shubhangiramesh2000@rediffmail.com

Dr. P.S. Hiremath
Gulbarga University, Gulbarga
Karnataka ,India
08472-275684
hiremathps@yahoo.co.in

ABSTRACT
The paper describes the process of character recognition using the Multi Class SVM classifier combined with a neural Network approach. The character recognition techniques or the OCRs are either a printed document recognition or the handwritten character recognition. SVM (Support Vector Machine) classifiers often have superior recognition rates in comparison to other classification methods. In this paper, a cooperation of combining SVM classifiers with Neural Network for handwritten digit recognition, using morphological feature set is proposed. An approach is proposed over here which can be used to detect and recognize both the categories of the characters. The morphological features or the shape structures would be extracted from the image and would be given as the training vector of the SVM classifier. The SVM classifier would classify the structures and generate the hyper-plane. Furthermore any given character would be recognized by first generating a Neural network containing all the support vectors from a class and then recognizing the character using the neural network technique.

Keywords
SVM, ANN, Character Recognition

1 INTRODUCTION

1.1 Brief review of Support Vector Machines

SVM is a binary classifier. The proposed adaptive hierarchical multi-class SVM classification scheme is a binary SVM tree. We describe how to build the adaptive hierarchical multi-class SVM classifier (SVM tree) in the training stage and illustrate how to use the SVM tree to classify new input patterns during the test phase. The core desired solution of SVM is to find optimal hyper plane between two classes with the symbols $y \in \{1, -1\}$ and y_i is label of the i[th] training instance $x_i \in R^n$ without distinction, we will refer x_i as instances, points , examples, or vectors. So given a training set of instance-label plain($\boldsymbol{x_i, y_i}$) , the standard SVM require the solution of the following Quadratic Programming (QP) optimization problem which can separate two different classes

$$\begin{aligned} \min: \quad & \frac{1}{2}(w * w) + C\sum_{i=1}^{n}\xi_i \\ \textit{subject to}: & \, y_i\left[w * \phi(x_i) + b\right] - 1 + \xi_i \geq 0 \\ & \xi_i \geq 0 \end{aligned} \tag{1}$$

Here instance x_i are mapped into a higher dimensional space by the function (phi). Then SVM finds a linear discrimination hyper plane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter to control trade-off between error ξ and margin. Furthermore,

$K(x_i, x_j) = \phi(x_i) * \phi(x_j)$ is called the kernel function. The normal coefficient of optimal discrimination hyper plane $\boldsymbol{w^*}$ is described as following formula:

$$w^* = \sum_{i=1}^{n}\alpha_i^* y_i x_i \qquad (\alpha_i^* \geq 0) \tag{2}$$

The final classification function is

$$label(x)=sgn[<w^*,x> + b^*] \tag{3}$$

Here b^* is scalar, which can be calculated according to $b^* = \boldsymbol{y_i} - \boldsymbol{W^* x_i}$

1.2 Learning Algorithm

Boosting was introduced as a method for boosting the performance of a weak learning algorithm. In boosting, each individual learner is trained using the training instances chose K according to the instance's probability distribution that is updated in proportional to the erromess of the instance. The idea of learning algorithm of different SVM-classifier in this paper is similar to that of boosting. For training the $(i+1)^{th}$ SVM classifier *Ki,* , we build a set of training instances

$$S_i = \left\{(x_1^i, y_1^i) \middle| (x_1^i, y_1^i) \in S_{i-1}, K_i(x_1^i) \neq y_1^i\right\}$$

A set is obtained by selecting the instances that are estimated incorrectly by SVM classifier K_i among data set s_i . This implies that the instances which are bard to classify are selected more frequently, these instances will be used to train next SVM classifier. Discrimination hypothesis is biased to classify the

examples which are most difficult to classify by the preceding hypotheses. This learning procedure repeats until all training instances can be classified correctly by classifier or a premeditated criterion is satisfied. Figure 1 gives the pseudo code of learning procedure of individual classifier.

$S_0 = \left\{ (x_1,\ y_1), \ldots\ldots, (x_i,\ y_i) \middle| x_i \in R^n,\ y_i \in (+1, -1) \right\}$

2. Set SVM type and parameter

3. Train $1, \ldots, N$ SVM classifier

4. For $i = 1$ to N

Training i^{th} SVM classifier K_i over S_{i-1};

Classifying S_{i-1} with K_i, then building new set

$S_i = \left\{ \left(x_1^i,\ y_1^i\right) \middle| \left(x_1^i,\ y_1^i\right) \in S_{i-1}, K_i\left(x_1^i\right) \neq y_1^i \right\}$

5. Output $1, \ldots N$ SVM classifier

Figure 1. Learning Algorithm Of Individual SVM-Classifier.

The training for the hierarchical SVM-tree classifier starts from the whole dataset. We recursively partition the current dataset into two non-overlapping data subsets. These two subsets are used as positive and negative samples to train an SVM classifier. The resulting SVM classifier is the current node classifier of the final SVM tree classifier. Since each classifier divides the data into two sets, [N – 1] such classifiers are needed to solve a N-class classification problem. One issue for the training algorithm of the SVM tree classifier is how to divide the classes into two separate subsets. The optimal partition is the one that the gap between these two subsets after partition is the largest. But in order to find the largest gap of the optimal partition, C2 n comparisons are needed. To alleviate the heavy computation cost of direction comparison method, we view the division of the class sets into two subsets as a clustering problem. We wish to group the classes into two non-overlapping subsets. We represent each class by its corresponding mean vector $\mu_k = \frac{1}{n_k} \sum_{x_i \in C_k} x_i$, where n_k is the number of samples in class Ck and Xi is the data vector. Then we cluster the $N\ \mu_k S(k = 1, \ldots, n)$ into two algorithm (k = 2).

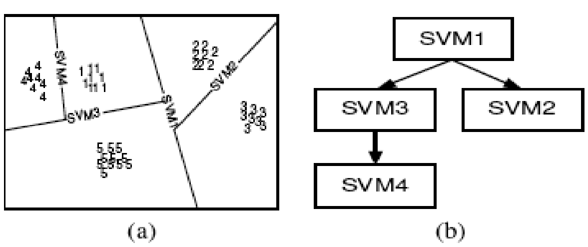

Figure 2. (a) Adaptive training of multi-class SVM classifier. (b) SVM tree.

1.3 Testing

The training algorithm described previously yields a binary SVM tree classifier. The SVM tree is very similar to the decision tree classifier. In proposed algorithm, we will replace this tree with neural network node. The only difference from the decision tree is that the decision function for the node is a SVM classifier. Therefore, the classification of new pattern is carried out in a top-to-bottom manner. The test pattern start from the root nodes. At every node, based on the classification result of the current node SVM (positive/negative), the test pattern is passed to the left/right branch of the tree. The final classification decision of the test pattern is given by the classification result of the final leaf node SVM classifier. The smallest computation of classifying the test pattern is just one SVM evaluation when we can make the decision at the top node. The worst case is [N – 1] SVM evaluations when we have to traverse all [N – 1] SVM nodes classifiers before we get the classification decision. The test phase for one pattern by one-against-one, one-against-rest and DAGSVM approaches require N(N - 1)=2, N and [N – 1] SVM evaluations respectively. Compared with those approaches, the proposed SVM ANN classifier is more efficient in the test phase. The efficiency gained in testing phase is very important for many practical applications since the classification stage in many practical applications are required to be online and requires fast response, while training can be done offline.

2 RELATED WORK

SVM [1] is a new approach of pattern recognition based on Structural Risk Minimization which is suitable to deal with magnitude features problems with a given finite amount of training data. SVM has been successfully exploited to a number of applications ranging from handwritten digit recognition, face identification, text categorization, bioinformatics to database marketing [2]. In spite of the already quite large body of work done in the domain of SVM, being a very active approach, there exist still a number of open questions that should be deliberated by future research. One of the most important ones seems to be improving the performance of SVMs for more satisfactory results. Prior knowledge about invariance is included for recognizing the handwritten digit in a principled way via the virtual SV mechanism of the feature space by Scholkopf et al DeCoste & Burl [6] expand a training set by applying various transformations inside the kernel function earlier. Moreover, constructing good ensembles of SVM classifiers has attracted more attention based on the premise that ensembles are oilen much more accurate than the individual classifiers that make them up. V.Vapnik [7] suggested the idea of the SVM ensemble and used the boosting technique to train each individual SVM and took another SVM for combining several SVMs. Hyun-Chul Kim et al [8] proposed to main different SVM by manipulating the training set (in methods of bootstrapping [IO] or boosting [ll]. In the SVM ensemble, the trained individual SVMs are aggregated to make a collective decision in several ways such as the majority voting, least-squares estimation-based weighting, and the double-layer hierarchical combining.

3 METHODOLOGY

Any character is a binary image, specifically recognized by the shapes of the images. The shape can be considered as the zernieke moments or the edges. In any document there could be optical noises present along with the documents. Specially in the handwritten documents the character shapes may not be unique always. Hence a preprocessing is mandatory. We will first apply a erosion with 3x3 structuring element which will eliminate the one bit errors and give a smooth edge. Then the characters are dilated with 2x2 element. The mean and the standard deviation will be extracted from the image and will be considered as the training vectors. We have considered 4 classes of 3,4,5,6 images. 10 images written in different formats are considered. W.r.t. each class all the vectors will be extracted. Now a SVM plane will be generated. This plane will provide clear boundaries for each plane. Now a neural network with 4 nodes would be generated instead of all the 40 points generated from all the vectors. A training image will be given as input and will be estimated using the neural network. The convolutional neural network (CNN) architecture proposed by LeCun , treats the feature extractor and the classification component identically. The feature extraction filters are implemented as a hidden layer with shared weights that are optimized together with the weights of the classification component so that the total classification error is minimized. Since the optimization is based on a gradient-based minimization procedure, the resulting feature extraction filters can have arbitrary continuous coefficients, not like the binary rectangular filters used in [8]. Neural networks trained with the usual mean-squared-error (or cross-entropy) minimization target function are prone to over fitting to the training data and may perform poorly on independent data, The support vector machine (SVM) classifier is optimizing an error function that minimizes the misclassifications on the training set and the L2 norm

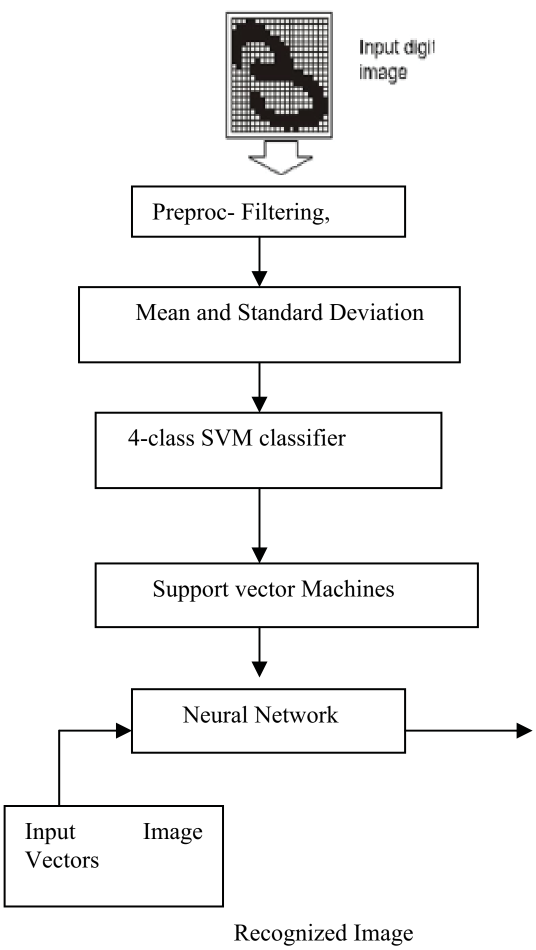

Figure 3: Character Recognition System.

3.1 Results

It is been observed by considering handwritten characters that the

classifier classifies correctly upto 97%. The system is not be tested for scan error or Gaussian errors. If the document is exposed to such errors, suitable spatial filtering need to be embedded in the preprocessing step.

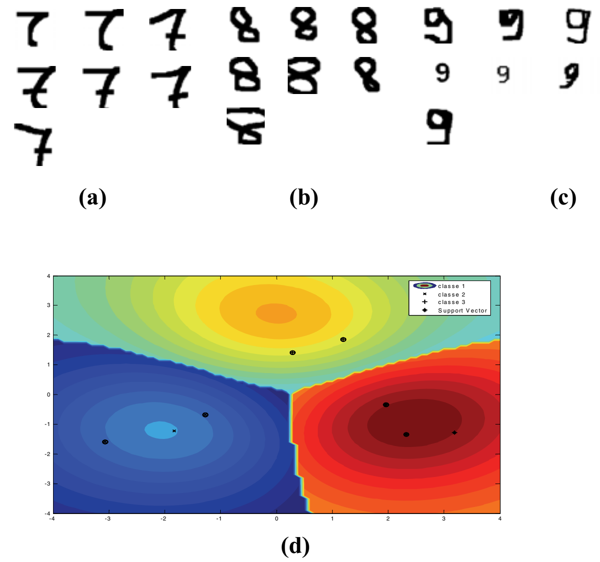

Figure 4: (a) Input Dataset Of Class 7 (b) Input Dataset Of Class 8 (c) Input Dataset Of Class 9

(d) Multiclass SVM Classifier With Gaussian Kernel

3.2 Conclusion

Increasing the number of training samples, indeed increases recognition rates of individual classifiers and their cooperation. On the other hand, increasing recognition rates of individual classifiers also increases their correlation that reduces the possibility for improvement of the cooperation recognition rates. In this paper, the cooperation of four feature sets for handwritten digit recognition using SVM classifiers is examined. The presented results show that it is efficient to achieve the recognition rate of a SVM Combined with NN applied on the feature set that includes features by extracting the morphological features. However, the classifier cooperation schemes reduce the classifier complexity and need for samples, and sometimes can increase the classifier performance.

4 ACKNOWLEDGMENT

The author is great full to the referees . Author admire the indispensable help by the president Dr. Poojya APPAJI for his flawless and forthright suggestions and a support . Also the author is indebted to prof. Dr. P.S. Hiremath , Chairman, Department Of Computer Science And Research Centre , Gulbarga University ,Gulbarga for great guidance , useful suggestions and encouragement. Further the author is also greatfull to Principal , Appa Institute Of Engg. And Technology, Gulbarga for their helpful support and encouragement during this work.

ISSN 0974-1003

5 REFERENCES

[1] .Corinna Cortes. Vladimir Vapnik. Support –Vector Networks. ***Machine Learning,*** 20,273-297 (1995)

[2] Christopher J.C. Burges. A Tutorial on Support Vector Machines for Pattem Recognition. ***Data Mining*** and ***Knowledge Discovery,*** 2,121-167 (1998)

[3] Isabelle Guyon et al. Gene Selection for Cancer Classification using Suppmt Vector Machines. ***Machine Learning,*** 46,389-422,2002

[4] TJoachims. Text categorization with support vector machines. In ***European Conference*** on ***Machine learning,*** 1998.

[5] Francis E.H. Tay., L.1. Cao Modified support vector machines in financial time series forecasting Neurocomputing.48 (2002) 847 – 861

[6] Scholkopf, B., Burges, C., & Vapnik, V. (1996). Incorporating invariances in support vector leaming machines. In C. von der Malsburg, W. von Seelen, J. C. Vorbi'uggen, & B. Sendhoff (Fds.). ***Arfijicial neural network-ICANWW*** (pp. 47-52). Berlin: Springer. Lecw Notes in Computer Science (Vol. 11 12).

[7] DeCoste, D. & Burl, M. C. (2000). Distortion-invariant recognition via jittered queries. In Computer Vision and Pattem Recognition (CVPR-2000). V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1999.

[8] Hyun-Chul Kim, Shaoning Pang, Hong-Mo le, Daijin Kim,Sung Yang Bang. Constructing support vector machine ensemble. Pattem Recognition 36 (2003) 2757 – 2767

[9] Breiman, L. (1996h). Bagging predictors. ***Machine Learning,*** 24,123-140.

[10] Yu Ying, Wang Xiaolong, etc. Efficient SVM-based Recognition of Chinese Personal Names. High Technology Letters. 2004(3)

[11] T. Joachims, 11 in: Making large-scale SVM Learning Practical. Advances in Kemel Methods - Support Vector Leaming, B. Scholkopf and C. Burges and A. Smola *(ed.)*.MlT Press, 1999.