Support Vector Machines with Huffman Tree Architecture for Multiclass Classification*

Gexiang Zhang**

School of Electrical Engineering, Southwest Jiaotong University,
Chengdu 610031 Sichuan, China
gxzhang@ieee.org

Abstract. This paper proposes a novel multiclass support vector machine with Huffman tree architecture to quicken decision-making speed in pattern recognition. Huffman tree is an optimal binary tree, so the introduced architecture can minimize the number of support vector machines for binary decisions. Performances of the introduced approach are compared with those of the existing 6 multiclass classification methods using U.S. Postal Service Database and an application example of radar emitter signal recognition. The 6 methods includes one-against-one, one-against-all, bottom-up binary tree, two types of binary trees and directed acyclic graph. Experimental results show that the proposed approach is superior to the 6 methods in recognition speed greatly instead of decreasing classification performance.

1 Introduction

Support vector machine (SVM), developed principally by Vapnik [1], provides a novel means of classification using the principles of structure risk minimization. The subject of SVM covers emerging techniques that have been proven to be successful in many traditional neural network-dominated applications [2]. SVM is primarily designed for binary classification problems. In real world, there are many multiclass classification problems. So how to extend effectively it to multiclass classification is still an ongoing research issue [3]. The popular methods are that multiclass classification problems are decomposed into many binary-class problems and these binary-class SVMs are incorporated in a certain way [4]. Some experimental results [3-9] verify that the combination of several binary SVMs is a valid and practical way for solving muticlass classification problems. Currently, there are mainly 6 methods for combining binary-class SVMs. They are respectively one-against-all (OAA) [3,5], one-against-one (OAO) [3,6], directed acyclic graph (DAG) [3,7], bottom-up binary tree (BUBT) [8,9], two types of binary trees labeled as BT1 and BT2 [10]. For an N-class classification problem, these methods need test at least $\log_2 N$ binary SVMs for classification

[*] This work was supported by the National EW Laboratory Foundation (NEWL51435QT220401)

[**] Student Member, IEEE

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 24–33, 2005.

decision. To decrease the number of binary SVMs needed in testing procedure, a novel multiclass SVM with Huffman tree architecture (HTA) is proposed in this paper. The outstanding characteristic of the introduced method lies in faster recognition speed than OAA, OAO, DAG, BUBT, BT1 and BT2 instead of lowering classification capability.

2 Support Vector Machines

For many practical problems, including pattern matching and classification, function approximation, optimization, data clustering and forecasting, SVMs have drawn much attention and been applied successfully in recent years [1-9]. An interesting property of SVM is that it is an approximate implementation of the structure risk minimization induction principle that aims at minimizing a bound on the generation error of a model, rather than minimizing the mean square error over the data set [2]. SVM is considered as a good learning method that can overcome the internal drawbacks of neural network [1].

The main idea of SVM classification is to construct a hyperplane to separate the two classes (labelled $y \in \{-1, +1\}$) [1]. Let the decision function be

$$f(x) = sign(\mathbf{w} \cdot \mathbf{x} + b) \tag{1}$$

where $\mathbf{w}$ is weighting vector, and b is bias and $\mathbf{x}$ is sample vector. The following optimization problem is given to maximize the margin [1], i.e. to minimize the following function

$$\phi(\mathbf{w}, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i \tag{2}$$

Subject to

$$\begin{array}{ll} y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 - \xi_i & \\ \xi_i \geq 0 & i = 1, 2, \cdots, l \end{array} \tag{3}$$

In (4) and (5), y_i is the label of the ith sample vector $\mathbf{x}_i$; ξ_i and l are the ith relax variable of the ith sample vector and the dimension of sample vector, respectively [1].

The dual optimization problem of the above optimization problem is represented as

$$W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

Subject to

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{l}\alpha_i y_i, \quad i = 1, 2, \cdots, l \tag{5}$$

where $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function. $\mathbf{x}_i$ and $\mathbf{x}_j$ are the ith sample vector and the jth sample vector, respectively. α is a coefficient vector,$\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_l]$ [1]. The decision function of the dual optimization problem becomes the form:

$$f(x) = sign[(\sum_{i=1}^{l} \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + b)] \tag{6}$$

3 SVM with HTA

On the basis of fast-speed and powerful-function computers, various methods for signal recognition, character recognition and image recognition were presented [11]. However, comparing with human brain, the methods are obviously too slow. One of the most important reasons is that man identifies objects or patterns in an unequal probability way and most of the existing pattern recognition methods are based on a consideration: all patterns appear in an equal probability. However, in some applications such as radar emitter signal recognition, handwritten digit recognition in postal service and letter recognition in natural text, some patterns may come up frequently, while the others emerge rarely. If all patterns are recognized equiprobably, the efficiency may be very low. On the contrary, if the patterns with high probability are classified preferentially, the speed of recognizing all patterns can be quickened greatly. According to this idea, Huffman tree architecture is introduced to combine multiple binary-SVMs for multiclass classification problems.

An example of HTA with 8 nodes is given in Fig.1. HTA solves an N-class pattern recognition problem with a hierarchical binary tree, of which each node makes binary decision with an SVM. Using different probabilities of occurrence of different patterns, Huffman tree can be constructed using the following algorithm [12,13].

Step 1. According to N probability values $\{p_1, p_2, \cdots, p_N\}$ given, a set $F = \{T_1, T_2, \cdots, T_N\}$ of N binary trees is constructed. For every binary tree T_i $(i = 1, 2, \cdots, N)$, there is only one root node with probability value p_i and its both left-child tree and right-child tree are empty.

Step 2. Two trees in which root nodes have the minimal probability values in F are chosen as left and right child trees to construct a new binary tree. The probability value of root node in the new tree is summation of the probability values of root nodes of its left and right child trees.

Step 3. In step 2, the two trees chosen in F are deleted and the new binary tree constructed is added to the set F.

Step 4. Step 2 and step 3 are repeated till only one tree left in F. The final tree is Huffman tree.

Huffman tree is an optimal binary tree [13], which can minimize the number of SVMs for binary decisions. Once the probabilities of all nodes are given, the structure of HTA is determinate and unique. The SVM-HTA classifier takes advantage of both the efficient computation of HTA and the high classification accuracy of SVMs.

To bring into comparison, the performances of the 7 methods including OAA, OAO, DAG, BUBT, BT1, BT2 and HTA are analyzed in the following description.

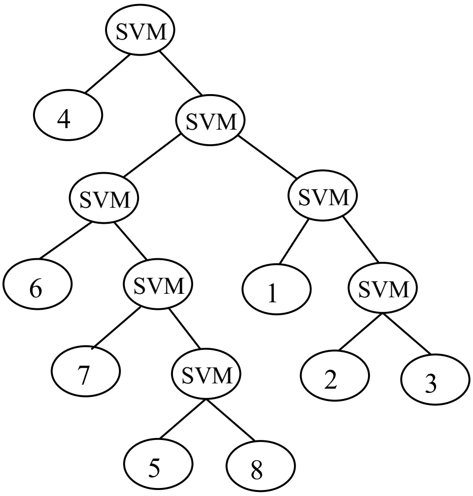

Fig. 1. Huffman tree architecture with 8 nodes

OAA is perhaps the simplest scheme for combining binary SVMs to solve multiclass problems. In OAA, every class need train to distinguish the rest classes, so there are N binary SVMs to be trained for an N-class classification problem, while in testing procedure, *Max Wins* strategy is usually used to classify a new example and consequently N binary decision functions are required to solve. The *Max Wins* strategy is

$$f(x) = arg \max_i (\mathbf{w}_i \cdot \mathbf{x} + b_i) \tag{7}$$

Another scheme called pairwise is used in OAO, DAG and BUBT. In this approach, each binary SVM separates a pair of classes and $N(N-1)/2$ binary SVMs in total are trained when there are N classes. In decision phase, there is much difference among the three methods. OAO uses traditional *Max Wins* strategy and need test $N(N-1)/2$ SVMs. DAG employs directed acyclic graph in which every class is eliminated step by step from the list composed of all classes. Thus, for a problem with N classes, $N-1$ binary SVMs will be tested in order to drive an answer. In BUBT, a bottom-up binary tree architecture is introduced to incorporate $N(N-1)/2$ binary SVMs trained and a tournament strategy is used to classify a new example. Similar to DAG, BUBT also need test $(N-1)$ binary SVMs for the classification decision. BT1 and BT2 use a hierarchical scheme that a multiclass classification problem is decomposed into a series of binary classification sub-problems. The difference between BT1 and BT2 lies in different decomposition method. BT1 separates one class from the rest classes with an SVM. In every step of decomposition, there is at least one terminal node between two siblings. Thus, for an N-class problem, BT1 need train $(N-1)$ binary SVMs and $(N^2+N-2)/(2N)$ binary decision functions are required to solve in testing procedure. While BT2 usually decomposes an N-class problem in a peer-to-peer way into $(N-1)$ binary classification subproblems. So there are $(N-1)$ binary SVMs to train in training procedure and only $\log_2 N$ binary SVMs need test in decision phase.

According to the above analysis, OAA, OAO, DAG, BUBT, BT1 and BT2 need train at least $(N-1)$ SVMs and require to test at least $\log_2 N$ SVMs for an N-class classification problem. While in HTA illustrated in Fig.1, only $(N-1)$ binary SVMs need be trained for N-class problem. Because Huffman tree is the optimal binary tree that has the minimal average depth, HTA need test much smaller than $\log_2 N$ SVMs for the classification decision. For example, in Fig.1, if the probability values of node 1 to node 8 are 0.135, 0.048, 0.058, 0.39, 0.039, 0.23, 0.067 and 0.033, respectively, HTA need test 2537 SVMs and BT2 need test 3000 SVMs when the number of testing samples is 1000. So among the 7 multiclass SVM classifiers, HTA need the minimal SVMs both in training and in testing procedures.

4 Simulations

4.1 Performance Test

HTA is evaluated on the normalized handwritten digit data set, automatically scanned from envelops by U.S. Postal Service (USPS) [7,14,15]. The USPS database contains zipcode samples from actual mails. This database is composed of separate training and testing sets. The USPS digit data consists of 10 classes (the integer 0 through 9), whose inputs are pixels of a scaled image. The numbers 0 through 9 have 1194, 1005, 731, 658, 652, 556, 664, 645, 542, 644 training samples respectively and have 359, 264, 198, 166, 200, 160, 170, 147, 166, 177 testing samples respectively. Thus, there are totally 7291 samples in training set and 2007 samples in the testing set. Every sample is made up of 256 features. The difference of the number of the 10 integers extracted from actual mails verifies that the 10 integers occur in an unequal probability. To be convenient for testing, the occurring probabilities of the 10 classes 0 through 9 in testing set are used to construct a Huffman tree. The probabilities of 0 through 9 are respectively 0.1789, 0.1315, 0.0987, 0.0827, 0.0997, 0.0797, 0.0847, 0.0732, 0.0827 and 0.0882. The constructed Huffman tree architecture is illustrated in Fig.2.

Seven approaches OAA, OAO, DAG, BUBT, BT1, BT2 and HTA are used to make comparison experiments. The computational experiments are done on a Pentium IV-2.0 with 512 MB RAM using MATLAB implementation by Steve Gunn. Gaussian kernel function $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma}} \tag{8}$$

and the same parameter C and σ are used in 7 SVM classifiers. We use similar stop-ping criteria that the KKT violation is less than 10^{-3}. For each class, 504

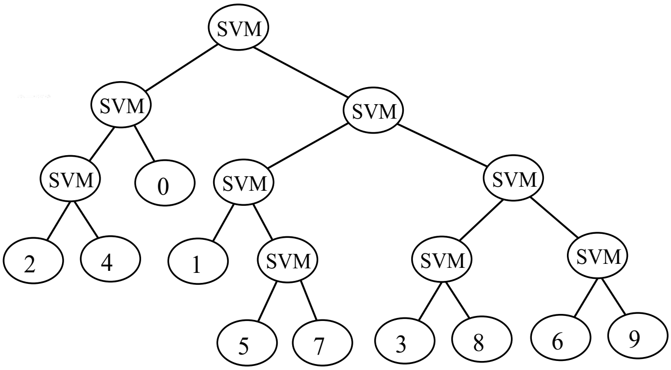

Fig. 2. Huffman tree architecture for digit recognition

samples selected randomly from its training set are used to train the SVM classifiers. The criterions for evaluating the performances of the 7 classifiers are their error rate and recognition efficiency including training time and testing time. All samples in the testing set are used to test the performances of the 7 classifiers. Statistical results of many experiments using the 7 classifiers respectively are given in Table 1.

Table 1 shows the results of experiments. HTA, BT1, BT2 and OAA consume much shorter training time than OAO, DAG and BUBT. Because HTA, BT1 and BT2 need train the same number of binary SVMs, the training time of the three methods has small difference. Similarly, the three methods including OAA, BUBT and DAG consume nearly same training time because they train the same number of SVMs. In the 7 methods, the testing time of HTA is the shortest. In Table 1, HTA consumes 445.44 seconds of testing time, which is a litter shorter than that of BT1 and BT2 and much shorter than that of OAA, OAO, DAG and BUBT. From the recognition error rate, HTA is much superior to OAA and OAO; HTA is a little superior to BUBT and BT1; HTA is not

Table 1. Experimental results of digit recognition

Methods	Training Time (sec.)	Testing time (sec.)	Error rate (%)
HTA	8499.21	445.44	3.42
OAA	9470.81	1391.57	95.59
OAO	44249.48	6068.75	89.57
DAG	43153.70	1217.69	2.32
BUBT	44938.34	1255.61	3.52
BT1	8397.35	641.14	4.83
BT2	8125.30	463.57	3.40

inferior to DAG and BT2. In a word, experimental results indicate that HTA has high recognition efficiency and good classification capability.

4.2 Application

In this subsection, an application example of radar emitter signal recognition is applied to make the comparison experiments of OAA, OAO, DAG, BUBT, BT1, BT2 and HTA. In the example, there are 8 modulation radar emitter signals (labeled as RES1, RES2, RES3, RES4, RES5, RES6, RES7, RES8, respectively). Some features of these radar emitter signals have been extracted in our prior work [21,22]. Two features obtained by the feature selection method [23] are used to recognize the 8 modulation radar emitter signals. In experiments, every radar emitter signal uses 360 training samples and thereby there are 2880 training samples in total. The training samples are employed to draw a feature distribution graph shown in Fig.3 to illustrate distribution of radar emitter signal features in feature space.

According to experts' experiences, the occurrence probabilities of the 8 modulation signals can be approximately considered as 0.135, 0.048, 0.058, 0.39, 0.039, 0.23, 0.067 and 0.033, respectively. Thus, the Huffman tree architecture constructed using 8 radar emitter signals is shown in Fig.1. In testing phase, there are 8000 testing samples in total and the number of testing samples of 8 radar emitter signals is computed in the proportion of 13.5%, 4.8%, 5.8%, 39%, 3.9%, 23%, 6.7% and 3.3%, respectively. Both training samples and testing sam-

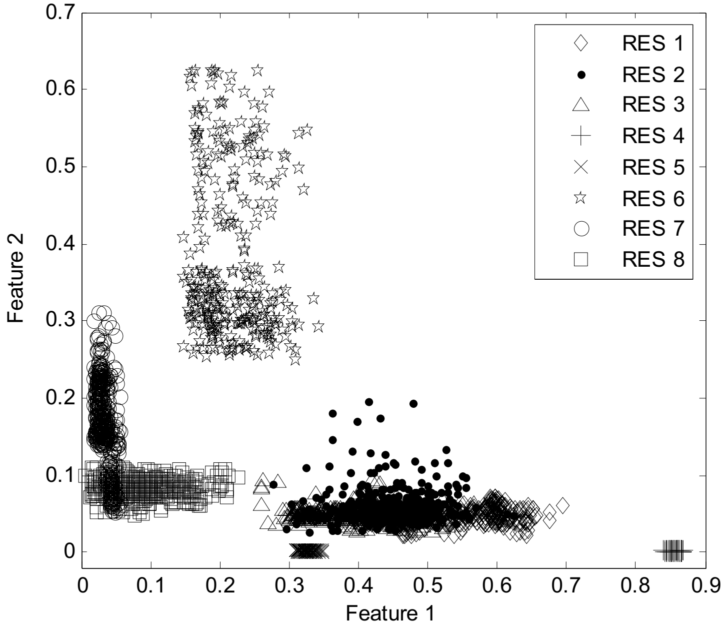

Fig. 3. Feature distribution graph

Table 2. Experimental results of RES recognition

Methods	Training Time (sec.)	Testing time (sec.)	Error rate (%)
HTA	1917.70	85.55	12.28
OAA	2154.95	255.08	45.64
OAO	8007.84	815.58	84.83
DAG	8151.94	199.75	13.40
BUBT	7737.49	238.01	12.25
BT1	1951.73	134.31	26.85
BT2	1910.94	112.59	22.00

ples are extracted from radar emitter signals when signal-to-noise (SNR) varies from 5 dB to 20 dB. Experimental results of OAA, OAO, DAG, BUBT, BT1, BT2 and HTA are given in Table 2.

Figure 3 shows that there are some overlaps between RES 7 and RES 8 and there is much confusion among RES 1, RES 2 and RES 3. This brings many difficulties to correct recognition. Also, the features of 8 radar emitter signals have good clustering. Table 2 presents the results of comparing 7 multiclass SVM classifiers. Although HTA is appreciably inferior to BUBT in recognition error rate and it needs a little more training time than BT2, HTA has higher recognition efficiency than OAA, OAO, BUBT, DAG and BT1. Especially, HTA is the best among the 7 methods for the testing time and it achieves lower recognition error rate than OAA, OAO, DAG, BT1 and BT2.

The experimental results of digit recognition and radar emitter signal recognition are consistent with theoretical analysis in Section 3. In pattern recognition including radar emitter signal recognition and USPS digit recognition, training is off-line operation, while testing is usually on-line operation. So the testing speed of classifiers is more important, especially in radar emitter signal recognition. Experimental results verify that HTA is the fastest among the 7 multiclass SVM classifiers instead of decreasing classification performance. This benefit is especially useful when the number of classes is very large.

5 Concluding Remarks

In the methods for combining multiple binary SVMs to solve multiclass classification problems, binary tree architecture is a good one because it needs small binary SVMs both in training phase and in testing phase. However, how to choose the root nodes in each layer is a very important issue in engineering applications when binary tree architecture is used to combine multiple binary-support-vector-machines. From the view of intelligent aspects of human brain in pattern recognition, this paper introduces Huffman tree architecture to design a multiclass classifier. For a real problem, the Huffman tree architecture is unique. The outstanding characteristic of the introduced architecture lies in faster recognition speed than the existing 6 methods. Though, this paper discusses the technique for quickening recognition speed only from the architecture for combining support vector machines. In fact, the recognition speed has also relation to the number of support vectors obtained in training phase of support vector machines. This problem will be further discussed in later paper.

References

1. Vapnik, V.N.: Statistical Learning Theory. New York, Wiley, (1998)
2. Dibike, Y.B., Velickov, S., and Solomatine, D.: Support Vector Machines: Review and Applications in Civil Engineering. Proceedings of the 2nd Joint Workshop on Application of AI in Civil Engineering, (2000) 215-218
3. Hsu, C.W., Lin, C. J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks. Vol.13, No.2. (2002) 415-425
4. Cheong, S.M., Oh,S.H., and Lee, S.Y.: Support Vector Machines with Binary Tree Architecture for MultiClass Classification. Neural Information Processing: Letters and Reviews. Vol.2, No.3. (2004) 47-51
5. Rifkin, R., Klautau, A.: In Defence of One-Vs-All Classification. Journal of Machine Learning Research. Vol.5, No.1. (2004) 101-141
6. Furnkranz, J.: Round Robin Classification. Vol.2, No.2. (2002) 721-747
7. Platt, J.C., Cristianini, N., and Shawe-Taylor, J.: Large Margin DAG's for Multiclass Classification. Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, Vol.12. (2000) 547-553
8. Guo, G.D., Li, S.Z.: Content-based Audio Classification and Retrieval by Support Vector Machines. IEEE Transactions on Neural Networks. Vol.14, No.1. (2003) 209-215
9. Guo, G.D., Li, S.Z., and Chan, K.L.: Support Vector Machines for Face Recognition. Image and Vision Computing, Vol.19, No.9. (2001) 631-638
10. Huo, X.M., Chen, J.H., and Wang, S.C., et al: Support Vector Trees: Simultaneously Realizing the Principles of Maximal Margin and Maximal Purity. Technical report. (2002) 1-19 (Available: www.isye.gatech.edu /research /files/tsui-2002-01.pdf)
11. Jain, A.K., Duin, R.P.W., and Mao, J.C.: Statistical Pattern Recognition: a Review. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.22, No.1. (2000) 4-37
12. Huang, J.H., Lai, Y.C.: Reverse Huffman Tree for Nonuniform Traffic Pattern. Electronics Letters. Vol.27, No.20. (1991) 1884-1886
13. Weiss, M.A.: Data Structures and Algorithm Analysis in C (2nd Edition). Addison Wesley, New York (1996)
14. Bredensteiner, E.J., Bennett, K.P.: Multicategory Classification by Support Vector Machines. Computational Optimization and Applications, Vol.12, No.1-3. (1999) 53-79
15. www-stat.Stanford.edu/ tibs/ElemStaLearn/datasets/zip.digits
16. Frey, P.W., Slate, D.J.: Letter Recognition Using Holland-Style Adaptive Classifiers. Machine Learning, Vol.6, No.2. (1991) 161-182
17. Murphy, P., Aha, D.W.: UCI Repository of Machine Learning Databases and Domain Theories. Available from [http: //www.ics.uci.edu / mlearn /MLRepository.html] (1995)
18. Gao, D.Q., Li, R.L., and Nie, G.P., et al.: Adaptive Task Decomposition and Modular Multilayer Perceptions for Letter Recognition. Proceedings of IEEE International Joint Conference on Neural Networks, Vol.4. (2004) 2937-2942

19. Vlad, A., Mitrea, A., and Mitrea, M., et al.: Statistical Methods for Verifying the Natural Language Stationarity Based on the First Approximation. Case Study: Printed Romanian. Proceedings of the International Conference Venezia per il trattamento automatico dellalingue, (1999) 127-132
20. Vlad, A., Mitrea, A., and Mitrea, M.: Two Frequency-Rank Laws For Letters In Printed Romanian. Procesamiento on Natural Language, No.24. (2000) 153-160
21. Zhang, G.X., Hu, L.Z., and Jin, W.D.: Intra-pulse Feature Analysis of Radar Emitter Sig-nals. Journal of Infrared and Millimeter Waves, Vol.23, No.6. (2004) 477-480
22. Zhang, G.X., Hu, L.Z., and Jin, W.D.: Resemblance Coefficient Based Intrapulse Feature Extraction Approach for Radar Emitter Signals. Chinese Journal of Electronics, Vol.14, No.2. (2005) 337-341
23. Zhang, G.X., Jin, W.D., Hu, L.Z.: A novel feature selection approach and its application. Lecture Notes in Computer Science. Vol.3314. (2004) 665-671.