A Comparison of Multi-class Support Vector Machine Methods for Face Recognition

Naotoshi Seo, sonots@umd.edu
Department of Electrical and Computer Engineering
The University of Maryland

December 6, 2007

Abstract

Support vector machines (SVMs) are originally designed for binary classification problem. How to effectively extend it for multi-class classification problem is still an on-going research issue. Several methods have been proposed where typically we construct a multi-class classifier by combining several binary classifiers. Some methods also have been proposed that consider all classes at once. In this paper, comparisons of these methods for face recognition problem were performed. The one-against-all [15], one-against-one [9][10], and DAGSVM [8] which are based on binary classifiers were compared. Furthermore, Weston's multi-class SVM [16] and Crammer's multi-class SVM [5] which originally solve multi-class classification problems were compared with them. As one of feature extraction methods for face recognition problem, the principal component analysis (PCA) [14][12] was applied for the facial images. This drastically reduced the number of attributes of feature vectors. The PIE facial image datasets [13] were used. The experimental results showed that the one-against-one method outperfomed the other methods and the DAGSVM method was the 2nd best. The DAGSVM would be preferred because of its less computational cost.

1 Introduction

Support vector machines (SVMs) [4] are originally designed for binary classification problem. How to effectively extend it for multi-class classification problem is stiill an on-going research issue. Several methods have been proposed where typically we construct a multi-class classifier by combining several binary classifiers. Some methods also have been proposed that consider all classes at once. As it is computationally more expensive to solve multi-class problems, comparisons of these methods using large-scale problems have not been seriously conducted yet.

In this paper, comparisons of these methods for face recognition problem which may be considered as one of large-scale problems are performed. In this paper, we compare one-against-all [15], one-against-one [9], and DAGSVM [8] which are based on binary classifiers. Furthermore, we compare them with Weston's multi-class SVM [16] and Crammer's multi-class SVM [5] which originally solve multi-class classification problems. My hypothesis is that the one-against-one method outperforms other methods and the DAGSVM is the 2nd best as results in [7].

As one of feature extraction methods for face recognition problem, the principal component analysis (PCA) is applied for the facial images [12]. A face recognition based on the PCA feature extraction method is proposed by Turk, et al [14] as *Eigenface* system which is robust to illumination changes. The PIE facial image database [13] whose pictures were taken under different illumination conditions is used for experiments.

In chapter 2, I review the binary SVM method. In chapter 3, I give introductions of multi-class SVM methods based on binary classifiers. In chapter 4, I give brief introductions of multi-class SVM methods which originally solve multi-class classification problems. In chapter 5, I review the Principal Component Analysis. In chapter 6, the experimental results are shown. Finally conclusions are given in chapter 7.

2 Support Vector Machines

The support vector machine [4], given labeled training data

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}, \quad \mathbf{x}_i \in \mathbf{X} \subset \mathbb{R}^d, \quad y_i \in \mathbf{Y} = \{-1, +1\},$$

constructs a maximal margin linear classifier in a high dimensional feature space, $\Phi(\mathbf{x})$, defined by a positive definite kernel function, $k(\mathbf{x}, \mathbf{x}')$, specifying an inner product in the feature space,

$$\Phi(\mathbf{x}).\Phi(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}').$$

A common kernel is the Gaussian radial basis function (RBF),

$$k(\mathbf{x}, \mathbf{x}') = e^{-\gamma||\mathbf{x}-\mathbf{x}'||^2}.$$

The discriminant function implemented by a support vector machine is given by

$$f(\mathbf{x}) = \left\{ \sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \right\} + b. \tag{1}$$

To find the optimal coefficients, α, of this expansion it is sufficient to maximize the functional,

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

in the non-negative quadrant,

$$0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, \ell, \tag{3}$$

subject to the constraint,

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{4}$$

C is a regularization parameter, controlling a compromise between maximizing the margin and minimizing the number of training set errors. The Karush-Kuhn-Tucker (KKT) conditions can be stated as follows:

$$\begin{align} \alpha_i = 0 \quad &\Longrightarrow \quad y_i f(\mathbf{x}_i) \geq 1, \tag{5} \\ 0 < \alpha_i < C \quad &\Longrightarrow \quad y_i f(\mathbf{x}_i) = 1, \tag{6} \\ \alpha_i = C \quad &\Longrightarrow \quad y_i f(\mathbf{x}_i) \leq 1. \tag{7} \end{align}$$

These conditions are satisfied for the set of feasible Lagrange multipliers, $\alpha^0 = \{\alpha_1^0, \alpha_2^0, \ldots, \alpha_\ell^0\}$, maximizing the objective function given by equation 2. The bias parameter, b, is selected to ensure that the second KKT condition is satisfied for all input patterns corresponding to non-bound Lagrange multipliers. Note that in general only a limited number of Lagrange multipliers, α, will have non-zero values; the corresponding input patterns are known as support vectors. Let I be the set of indices of patterns corresponding to non-bound Lagrange multipliers,

$$I = \{i \ : \ 0 < \alpha_i^0 < C\},$$

and similarly, let J be the set of indices of patterns with Lagrange multipliers at the upper bound C,

$$J = \{i \ : \ \alpha_i^0 = C\}.$$

Equation 1 can then be written as an expansion over support vectors,

$$f(\mathbf{x}) = \left\{ \sum_{i \in \{I,J\}} \alpha_i^0 y_i k(\mathbf{x}_i, \mathbf{x}) \right\} + b. \tag{8}$$

The support vector machine algorithm for binary classification problem was implemented as **SVM.m**.

3 Multi-class SVMs based on binary SVMs

In this chapter, I give introductions of multi-class SVM methods, one-against-all [15], one-against-one [9], and DAGSVM [8] which are based on binary classifiers.

Support vector machines are originally designed for binary pattern classification. Multi-class pattern recognition problems are commonly solved using a combination of binary SVMs and a decision strategy to decide the class of the input pattern. Each SVM is independently trained. The training data set $(\mathbf{x}_i, c_i)$ consists of N examples belonging to M classes. The class label $c_i \in 1, 2, \cdots, M$. We assume that the number of examples for each class is the same, i.e., N/M.

3.1 One-against-all

The one-aginst-all [15] (also known as 1-v-r or one-versus-rest) is the probably earliest implementation for multi-class SVM classification.

In this approach, an SVM is constructed for each class by discriminating that class against the remaining $(M-1)$ classes. The number of SVMs used in this approach is M. A test pattern $\mathbf{x}$ is classified by using the *winner-takes-all* decision strategy, i.e., the class with the maximum value of the discriminant function f(x) is assigned to it. All the N training examples are used in constructing an SVM for a class. The SVM for class k is constructed using the set of training examples and their desired outputs, $(\mathbf{x}_i, y_i)$.. The desired output y_i for a training example $\mathbf{x}_i$ is defined as follows:

$$y_i = \begin{cases} +1 & \text{if } c_i = k \\ -1 & \text{if } c_i \neq k \end{cases} \tag{9}$$

The examples with the desired output $y_i = +1$ are called *positive* examples and the examples with the desired output $y_i = -1$ are called *negative* examples. An optimal hyperplane is constructed to separate N/M positive examples from $N(M-1)/M$ negative examples.

The one-against-all algorithm was implemented in **MSVM.m** with option '1vr' which extends a binary SVMs implementation **SVM.m**.

3.2 One-against-one

The one-against-one [9] method is also known as 1-v-1 or one-versus-one method, and first introduced on SVM as pairwise SVM [10].

In this approach, an SVM is constructed for every pair of classes by training it to discriminate the two classes. Thus, the number of SVMs used in this approach is $M(M-1)/2$. An SVM for a pair of classes (k, m) is constructed using training examples belonging to the two classes only. The desired output y_i for a training example $\mathbf{x}_i$ is defined as follows:

$$y_i = \begin{cases} +1 & \text{if } c_i = k \\ -1 & \text{if } c_i = m \end{cases} \tag{10}$$

The maxwins [6] strategy is commonly used to determine the class of a test pattern $\mathbf{x}$ in this approach. In this strategy, a majority voting scheme is used. If $f_{km}(x)$, the value of the discriminant function of the SVM for a pair of classes (k, m), is positive, then class k wins a vote. Otherwise, class m wins a vote. Outputs of SVMs are used to determine the number of votes won by each class. The class with maximum number of votes is assigned to the test pattern. When there are multiple classes with the maximum number of votes, the class with maximum value of the total magnitude of discriminant functions (TMDF) is assigned. The total magnitude of discriminant functions for class k is defined as follows:

$$\text{TMDF}_k = \sum_m |f_{km}(\mathbf{x})| \tag{11}$$

where the summation is over all m with which class k is paired.

The one-against-one algorithm was implemented in **MSVM.m** with option '1v1' which extends a binary SVMs implementation **SVM.m**.

3.3 DAGSVM

The Directed Acyclic Graph Support Vector Machines (DAGSVM) is proposed in [8]. In the training phase, it works as the one-against-one method solving $M(M-1)/2$ binary SVMs. However, in the testing phase, it uses a rooted binary DAG which has $M(M-1)/2$ internal nodes and M leaves. Fig 1 shows the DAG scheme. Given a test sample $\mathbf{x}$, starting at the root node, a pairwise SVM decision is made and either class is rejected. Then it moves to either left or right depending on the result, and continues until reaching to one of leaves which indicates the predicted class. So the DAG requires $M-1$ comparisons and hence is more efficient than the one-against-one method.

The DAGSVM algorithm was implemented in **MSVM.m** with option 'DAG' which extends a binary SVMs implementation **SVM.m**.

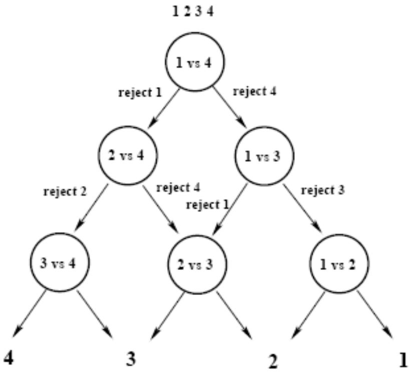


Figure 1: A DAG for four classes: each node is a pairwise classifier between two classes. A decision of rejecting one of the two class labels is made at each node.

4 Multi-class SVMs considering multi-class at once

4.1 Weston's multi-class SVM

In [16], an approach for multi-class problems by solving one single optimization problem was proposed.

The idea is similar to the one-against-all approach. M two-class decision rules where the m th function $\mathbf{w}_m^T\phi(\mathbf{x}) + b$ separates training vectors of the class m from the other are constructed.

Thus, there exists M decision functions. This approach solves an optimization problem formulated as follows:

$$\min_{\mathbf{W},\mathbf{B},\xi} \frac{1}{2}\sum_{m=1}^{M} \mathbf{w}_m^T \mathbf{w}_m + C\sum_{i=1}^{l}\sum_{m\neq y_i} \xi_i^m \tag{12}$$

subject to

$$\mathbf{w}_{y_i}^T \phi(\mathbf{x}_i) + b_{y_i} \geq \mathbf{w}_m^T \phi(\mathbf{x}_i) + b_m + 2 - \xi_i^m \tag{13}$$

$$\xi_i^m \geq 0 \text{ for } i = 1, \cdots, l, \tag{14}$$

$$m \in \{1, \cdots, M\} \backslash y_i. \tag{15}$$

The decision function is

$$\text{argmax}_{m=1,\cdots,M}(\mathbf{w}_m^T \phi(\mathbf{x}) + b_m) \tag{16}$$

which is as of the one-against-all method.

The Weston's multi-class SVM is implemented in **BSVM** [3].

4.2 Crammer's multi-class SVM

In [5], an approach for multi-class problems by solving one single optimization problem was proposed. The Crammer's multi-class SVM [5] is the one sometimes denoted as simply M-SVM.

This approach solves an optimization problem formulated as follows:

$$\min_{\mathbf{W},\xi} \frac{1}{2}\sum_{m=1}^{M} \mathbf{w}_m^T \mathbf{w}_m + C\sum_{i=1}^{l} \xi_i \tag{17}$$

subject to

$$\mathbf{w}_{y_i}^T \phi(\mathbf{x}_i) - \mathbf{w}_m^T \phi(\mathbf{x}_i) \geq e_i^m - \xi_i \text{ for } i = 1, \cdots, l \tag{18}$$

where $e_i^m = 1 - \delta_{y_i,m}$ where δ is the Kronecker deleta function. The decision function is

$$\text{argmax}_{m=1,\cdots,M}(\mathbf{w}_m^T \phi(\mathbf{x})) \tag{19}$$

The Crammer's multi-class SVM is implemented in **BSVM** [3].

5 Principal Component Analysis

Principal components analysis (PCA) [12] is a technique used to reduce multidimensional data sets to lower dimensions for analysis. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA is theoretically the optimum transform for a given data in least square terms.

PCA can be used for dimensionality reduction in a data set by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data.

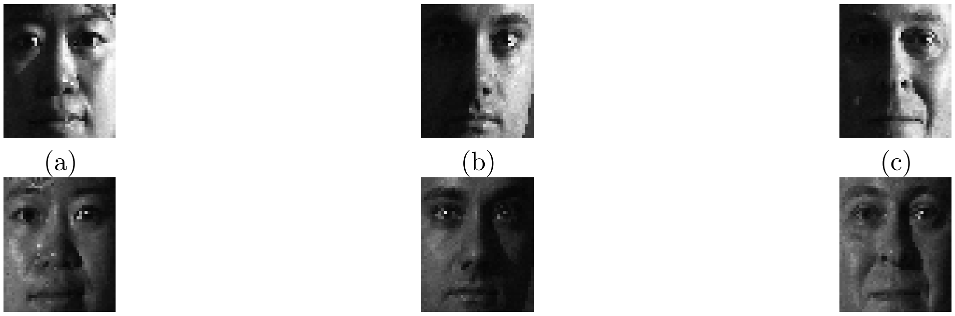
For a data $\mathbf{X}^T$ with zero empirical mean (the empirical mean of the distribution has been subtracted from the data set), where each row represents a different repetition of the experiment, and each column gives the results from a particular probe, the PCA transformation is given by:

$$\mathbf{Y}^T = \mathbf{X}^T\mathbf{W} \tag{20}$$
$$= \mathbf{V}\boldsymbol{\Sigma} \tag{21}$$

where $\mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T$ is the singular value decomposition (svd) of $\mathbf{X}^T$. Principal Component Analysis was implemented as **doPCA.m**, and PCA transformation was implemented as **doPCAProj.m**.

6 Experimental Results

A few sample facial images from PIE database [13] are shown in Fig 2. PIE database has 21 images of image size 48x40 for each person and there exist 81 people.



(a) (b) (c)

(d) (e) (f)

Figure 2: Examples from PIE database [13]. Original image size 48x40. (a-c) 3 persons (d-f) same 3 persons with different illuminations

6.1 Experiments on Binary Classification Problem

Experiments for binary classification problem which classifies person 1 and person 2 using PCA and binary SVMs were performed to estimate sufficient number of PCA coefficients (attributes). Here, I used the RBF kernel with parameter $\gamma = 0.5$ and regularization parameter $C = 1$ for SVM. The results are shown in Fig 6.1. There are 14 training data and 7 testing data for each person (thus 28 training data and 14 testing data totally) and error rate for testing data are shown. From this experiments, I concluded 5 attributes are sufficient. The parameters for SVM which were used during experiments may not be optimal, thus 5 attributes must be sufficient for the case with optimal parameters too. Note that each feature vector originally have 1920 attributes (48x40 image size), and they were drastically reduced into only 5 attributes.

6.2 Comparisons of Multi-class SVMs

Comparisons for multi-class SVMs are performed here. For each method, the optimal regularization parameter C and the kernel parameter γ were estimated by repeating classifications for $C = [2^{12}, 2^{11}, \cdots, 2^{-2}]$ and $\gamma = [2^4, 2^3, \cdots, 2^{-10}]$ (the total combination is $15 \times 15 = 225$). The classification accuracy of SVM methods with these optimal parameters were compared.

The 66.6% of dataset was used for training and 33.3% was used for testing, thus there exists 14 images and 7 images respectively for each person since the total is 21. The results for the 3-class (3-person classification), 4-class, and 5-class problems are shown in the Table 1. The results show

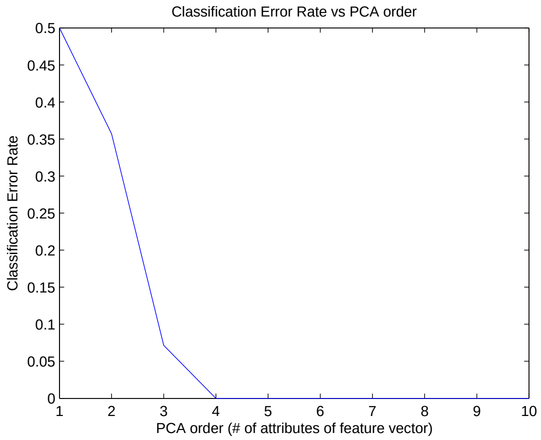


Figure 3: Classification error rate vs # of attributes of feature vectors

that the one-against-one method outperformed other methods and the 2nd best was the DAGSVM method.

3-class	One-against-all	One-against-one	DAGSVM	Weston	Crammer
Optimal (C, γ)	$(2^{-2}, 2^{-10})$	$(2^{1}, 2^{-1})$	$(2^{1}, 2^{-10})$	$(2^{1}, 2^{-2})$	$(2^{1}, 2^{-9})$
rate	0.8980	**0.9592**	**0.9592**	0.9048	0.8571

4-class	One-against-all	One-against-one	DAGSVM	Weston	Crammer
Optimal (C, γ)	$(2^{2}, 2^{-6})$	$(2^{-1}, 2^{-10})$	$(2^{0}, 2^{-1})$	$(2^{-2}, 2^{-10})$	$(2^{-2}, 2^{4})$
rate	0.6786	**0.9286**	**0.9286**	0.8571	0.8571

5-class	One-against-all	One-against-one	DAGSVM	Weston	Crammer
Optimal (C, γ)	$(2^{10}, 2^{-5})$	$(2^{-1}, 2^{-8})$	$(2^{-2}, 2^{1})$	$(2^{-2}, 2^{-0})$	$(2^{-2}, 2^{-1})$
rate	0.3714	**0.8286**	0.8000	0.8000	0.8000

Table 1: Comparison of Multi-class SVMs for face recognition. The # of PCA coefficients is 5.

Other experiments were also performed to find sufficient number of PCA coefficients (attributes) to achieve 0 error rate. For 3 and 4 class problems, the DAGSVM and the one-against-one method achieved 0 error rate when the number of PCA coefficients is greater than or equal to 6. For the 5-class problem, the 8 PCA coefficients were required for the DAGSVM and the one-against-one method. Finally, the 10-class classification with one-against-one method was performed. In this case, the 14 PCA coefficients were required to achieve 0 error rate.

7 Conclusion

The classification accuracy of multi-class SVM methods, one-against-all [15], one-against-one [9][10], DAGSVM [8] which are based on binary classifiers, furthermore, Weston's multi-class SVM [16], and Crammer's multi-class SVM [5] which originally solve multi-class classification problems were compared for face recognition system. As one of feature extraction methods for face recognition, the principal component analysis (PCA) [14][12] was applied for the facial images. The experimental result showed that the one-against-one method outperformed the other methods and the 2nd best was the DAGSVM as my hypothesis. The DAGSVM would be preferred if less computational cost is required.

Bibliography

[1] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[2] C.-C. Chang and C. j. Lin., *Libsvm: a library for support vector machines*, 2001, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] ______, *Bsvm: a library for multi-class support vector machines*, 2006, `http://www.csie.ntu.edu.tw/~cjlin/bsvm`.

[4] C. Cortes and V. Vapnik, *Support-vector network*, Machine Learning (1995), 273–297.

[5] K. Crammer and Y. Singer, *On the algorithmic implementation of multiclass kernel-based vector machines*, Technical report, School of Computer Science and Engineering, Hebrew University (2001).

[6] J. H. Friedman, *Another approach to polychotomous classification*, Technical report, Stanford, Department of Statistics (1996).

[7] C. Hsu and C. Lin, *A comparison of methods for multi-class support vector machines*, Technical report, Department of Computer Science and Information Engineering, National Taiwan University (2001), `citeseer.ist.psu.edu/hsu01comparison.html`.

[8] J. Shawe-Taylor J. Platt, N. Cristianini, *Large margin dags for multiclass classification*, in Advances in Neural Information Processing Systems 12 (2000), 547–553.

[9] S. Knerr, L. Personnaz, and G. Dreyfus, *Nurocosingle-layer learning revisited: A stepwise procedure for building and training a neural network*, Springer, 1990.

[10] U. Kressel, *Pairwise classification and support vector machines*, in Advances in Kernel Methods - Support Vector Learning (1999).

[11] Stan Z. Li and Anil K. Jain, *Handbook of face recognition*, Springer, 2004.

[12] David G. Stork Richard O. Duda, Peter E. Hart, *Patten classification, second edition*, Wiley Interscience, 2000.

[13] Terence Sim, Simon Baker, and Maan Bsat, *The cmu pose, illumination, and expression (pie) database*, Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, May 2002.

[14] M. Turk and A. Pentland., *Eigenfaces for recognition*, Journal of Cognitive Neurosicence **3** (1991), no. 1, 71–86.

[15] V. Vapnik, *Statistical learning theory*, Wiley, New York, 1998.

[16] J. Weston and C. Watkins, *Multi-class support vector machines*, Technical report CSD-TR-98-04 (1998).

[17] Wenyi Zhao and Rama Chellappa, *Face processing: Advanced modeling and methods*, Elsevier, 2006.