Fast Recognition of Multi-view Faces with Feature Selection

Zhi-Gang Fan and Bao-Liang Lu
Departmart of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, China
{zgfan, blu}@cs.sjtu.edu.cn

Abstract

We propose a discriminative feature selection method utilizing support vector machines for the challenging task of multi-view face recognition. According to the statistical relationship between the two tasks, feature selection and multi-class classification, we integrate the two tasks into a single consistent framework and effectively realize the goal of discriminative feature selection. The classification process can be made faster without degrading the generalization performance through this discriminative feature selection method. On the UMIST multi-view face database, our experiments show that this discriminative feature selection method can speed up the multi-view face recognition process without degrading the correct rate and outperform the traditional kernel subspace methods.

1. Introduction

Multi-view face recognition is a more challenging task than frontal view face recognition. Face recognition techniques have been developed over the past few decades. But many of those existing face recognition techniques are only effective for frontal view faces. The difficulties of multi-view face recognition is obvious because of the nonlinear manifolds existing in the data space. How to learn the global structure of the nonlinear manifolds is the key to solve this problem.

Subspace methods are classical paradigms for face recognition [16]. Eigenfaces method [14] is a first breakthrough for the subspace techniques. It uses the Principal Component Analysis (PCA) to produce a most expressive subspace for face representation and recognition. Fisherfaces method [2] is an example of the discriminating subspace methods. It uses the Linear Discriminant Analysis (LDA) to seek a set of features best separating face classes. The Bayesian algorithm using probabilistic subspace is proposed in [11]. It solves the face recognition problem through classifying intrapersonal and extrapersonal variations. Independent Component Analysis (ICA) has been applied into face recognition in [1]. It produces a nonorthogonal subspace for face representation and recognition. However, all those linear methods mentioned above are not effective for multi-view face recognition because of the nonlinear distribution of face patterns. For these nonlinear problems, some kernel subspace methods have been proposed in recent years. These nonlinear subspace methods combine the strengths of the traditional subspace methods and kernel machine algorithms to solve the nonlinear problems. Kernel PCA [13] combines PCA and kernel machine to form a nonlinear PCA method. Kernel Direct Discriminant Analysis (KDDA) [10] combines the strengths of the LDA and kernel machine algorithm for face recognition. It first nonlinearly maps the original input space to an implicit high-dimensional feature space,where the distribution of face patterns is hoped to be linearized and simplified. Then, a new variant of the LDA method is introduced to effectively solve the nonlinear problem and derive a set of optimal discriminant basis vectors in the feature space. However, these kernel methods still can not solve the small sample size problems very well. And, at the same time, their classification processes are very slow due to the vast calculating time in kernel machines.

In order to speed up the multi-view face recognition process and maintain the generalization performance, we propose a discriminative feature selection method utilizing support vector machines (SVMs) in this paper. Being different from subspace methods, this SVM based Discriminative Feature Selection (SVM-DFS) method direct selects most discriminative features without linearly combining the original features. Motivated by the success that the statistical learning theory [15] possesses for the small sample size problem, SVM-DFS integrates the two tasks, feature selection and multi-class classification, into a single consistent framework and effectively realizes the goal of discriminative feature selection according to the statistical relationship between feature selection and multi-class classification. SVMs are originally designed for binary classification and multi-class SVMs always work through combin-

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 IEEE

ing the outputs of multiple binary classifiers [12]. Utilizing the characteristics of SVMs, we can rank discriminative features for feature selection. In [5], a SVM based recursive feature elimination method has been proposed to select features for binary classification problems. Using multi-class SVMs, SVM-DFS is especially designed for multi-class classification problems with the fact that face recognition is a classical multi-class classification problem. To testify SVM-DFS's advantages over traditional kernel subspace methods, we have made experiments using SVM-DFS and KDDA respectively for comparison studies.

2. SVM based Discriminative Feature Selection (SVM-DFS)

Using multi-class SVMs, SVM-DFS is designed for multi-class classification problems.

2.1. Multi-class SVMs

Support vector machine is a machine learning technique that is well-founded in statistical learning theory. On the basis of the VC dimension concept, constructive distribution-independent bounds on the rate of convergence of learning processes can be obtained and the structural risk minimization principle has been found. The new understanding of the mechanisms behind generalization not only changes the theoretical foundation of generalization, but also changes the algorithmic approaches to pattern recognition. As an application of the theoretical breakthrough, SVMs have high generalization ability and are capable of learning in high-dimensional spaces with a small number of training data. It accomplishes this by minimizing a bound on the empirical error and the complexity of the classifier, at the same time [15]. With probability at least $1-\eta$, the inequality

$$R(\alpha) \leq R_{emp}(\alpha) + \Phi\left(\frac{h}{l}, \frac{-\log(\eta)}{l}\right) \tag{1}$$

holds true for the set of totally bounded functions. Here, $R(\alpha)$ is the expected risk, $R_{emp}(\alpha)$ is the empirical risk, l is the number of training examples, h is the VC dimension of the classifier that is being used, and $\Phi(\cdot)$ is the VC confidence of the classifier. It will bound the generalization error as in (1). This controlling of both the training set error and the classifier's complexity has allowed SVMs to be successfully applied to very high dimensional learning tasks [15].

The SVMs algorithm formulates the training problem as one that finds, among all possible separating hyperplanes, the one that maximizes the distance between the closest elements of the two classes. In practice, this is determined through solving a quadratic programming problem. The SVMs have the general form of the decision function for a point x is:

$$f(x) = \text{sign}\left(\sum_{\text{support vectors}} y_i \alpha_i K(x_i, x) - b\right) \tag{2}$$

where α_i are Lagrange parameters obtained in the optimization step, y_i are class labels, and $K(\cdot,\cdot)$ is the kernel function. The kernel function can be various type. The linear kernel function is $K(x,y) = x \cdot y$; the radial basis function (RBF) kernel function is $K(x,y) = \exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right)$; and the polynomial kernel function is $K(x,y) = (x \cdot y + 1)^n$.

SVMs are originally designed for binary classification. Multi-class SVMs are extension of the binary SVMs. Currently there are two types of approaches for multi-class SVMs. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation [12]. And the multiple binary classifiers combining methods mainly have three types: "one-versus-all", "one-versus-one" and "part-versus-part"[9].

One-versus-all multi-class SVMs construct k SVM models, where k is the number of classes. The ith SVM is trained with all of the examples in the ith class with positive labels, and all other examples with negative labels. It can be determined that x is in the class which has the largest value of the decision function

$$\text{class of } x \equiv \arg\max_{i=1,\ldots,k} f_i(x) \tag{3}$$

Another one-versus-one multi-class SVMs constructs $k(k-1)/2$ classifiers where each one is trained on data from two classes. The decision function for classification between the ith class and the jth class is $f_{ij}(x)$. After all $k(k-1)/2$ classifiers are constructed, the future testing usually uses a strategy of "Max Wins" voting. If $f_{ij}(x)$ indicates x is in the ith class, then the vote for the ith class is added by one. Otherwise, the jth is increased by one. Then we predict x is in the class with the largest vote.

In [12], all those approaches for multi-class SVMs are carefully studied and it has been pointed out that one-versus-all method is as accurate as any other methods. Being not more accurate, the one optimization formulation method is generally complicated to implement and slow to train. When the number k of the classes is large in multi-classification, the number $k(k-1)/2$ of the modular classifiers of the one-versus-one method will be very large and the respond speed of the final classification model will be very slow. However, the one-versus-all method only has k modular classifiers and can have faster respond speed than the one-versus-one method. In this paper, we use one-versus-all multi-class SVMs considering the classifiers' respond time.

2.2. Feature Selection in Binary Classification

In the linear case of binary classification, the decision function equation (2) can be reformed as:

$$f(x) = \text{sign}\,(w \cdot x - b) \tag{4}$$

where w obtained from

$$w = \sum_{\text{support vectors}} y_i \alpha_i x_i \tag{5}$$

The inner product of weight vector $w = (w_1, w_2, \ldots, w_n)$ and input vector $x = (x_1, x_2, \ldots, x_n)$ determines the value of $f(x)$. Intuitively, the input features in a subset of $(x_1, x_2, \ldots, x_n)$ that are weighted by the largest absolute value subset of $(w_1, w_2, \ldots, w_n)$ influence most the classification decision. If the classifier performs well, the input features subset with the largest weights should correspond to the most informative features [5]. Therefore, the weights $|w_i|$ of the linear decision function can be used as feature ranking criterion. According to the feature ranking criterion, we can select the most discriminative features for the binary classification task.

Furthermore, support vectors can be used as evidence for feature ranking [6][3]. Assume the distance between the optimal hyperplane and the support vectors is Δ, the optimal hyperplane can be viewed as a kind of Δ-margin separating hyperplane which is located in the center of margin $(-\Delta, \Delta)$. According to [15], the set of Δ-margin separating hyperplanes has the VC dimension h bounded by the inequality

$$h \leq \min\left(\left[\frac{R^2}{\Delta^2}\right], n\right) + 1 \tag{6}$$

where R is the radius of a sphere which can bound the training vectors $x \in X$. Inequality (6) points out the relationship between margin Δ and VC dimension: a larger Δ means a smaller VC dimension. Therefore, in order to obtain high generalization ability, we should still maintain margin large after feature selection. However, because the dimensionality of original input space has been reduced after feature selection, the margin is usually to shrink and what we can do is trying our best to make the shrink small to some extent. Therefore, in feature selection process, we should preferentially select the features which make more contribution to maintaining the margin large. This is another evidence for feature ranking. To realize this idea, a coefficient c_k is given by

$$c_k = \left| \frac{1}{l_+} \sum_{i \in SV_+} x_{i,k} - \frac{1}{l_-} \sum_{j \in SV_-} x_{j,k} \right| \tag{7}$$

where SV_+ denotes the support vectors belong to positive samples, SV_- denotes the support vectors belong to negative samples, l_+ denotes the number of SV_+, l_- denotes the number of SV_-, and $x_{i,k}$ denotes the kth feature of support vector i in input space R^n. The larger c_k indicates that the kth feature of input space can make more contribution to maintaining the margin large. Therefore, c_k can assist $|w_k|$ for feature ranking. The solution is that, combining the two evidences, we can order the features by ranking $c_k|w_k|$.

In the nonlinear case of binary classification, a cost function J is computed on training samples for feature ranking. $DJ(i)$ denotes the change in the cost function J caused by removing a given feature or, equivalently, by bringing its weight to zero. $DJ(i)$ can be used as feature ranking criterion. In [5], $DJ(i)$ is computed by expanding J in Taylor series to second order. At the optimum of J, the first order term can be neglected, yielding:

$$DJ(i) = \frac{1}{2} \frac{\partial^2 J}{\partial w_i^2} (Dw_i)^2 \tag{8}$$

where the change in weight Dw_i corresponds to removing feature i.

For the nonlinear SVMs with the nonlinear decision function $f(x)$, the cost function J being minimized is:

$$J = \frac{1}{2} \alpha^T H \alpha - \alpha^T v \tag{9}$$

where H is the matrix with elements $y_h y_k K(x_h, x_k)$, α is Lagrange parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, and v is a n dimensional vector of ones [5]. To compute the change in cost function caused by removing input component i, one leaves the α's unchanged and one re-computes matrix H. This corresponds to computing $K(x_h(-i), x_k(-i))$, yielding matrix $H(-i)$, where the notation $(-i)$ means that component i has been removed. Thus, the feature ranking criterion for nonlinear SVMs is:

$$DJ(i) = \frac{1}{2} \left(\alpha^T H \alpha - \alpha^T H(-i) \alpha \right) \tag{10}$$

Computation for $DJ(i)$ is a little more expensive than the linear case. However, the change in matrix H must be computed for support vectors only, which makes it affordable for small numbers of support vectors.

For the convenience of representation, in both linear and nonlinear cases of binary classification, we denote feature ranking criterion as r_i for the ith feature in the input space R^n. In linear case of binary classification, r_i is

$$r_i = c_i |w_i| \tag{11}$$

In nonlinear case of binary classification, r_i is

$$r_i = \frac{1}{2} \left(\alpha^T H \alpha - \alpha^T H(-i) \alpha \right) \tag{12}$$

Using feature ranking criterion r_i, we can select most discriminative features for binary classification task.

2.3. Feature Selection in Multi-class Classification

In the case of multi-class classification, we use one-versus-all method for multi-class SVMs. Because SVMs are originally designed for binary classification, how to effectively extend them for multi-class classification is still an ongoing research issue [12] [7]. In [12], all those approaches for multi-class SVMs are carefully studied and it has been pointed out that one-versus-all method is as accurate as any other methods. We choose one-versus-all method in our work because it can respond faster than one-versus-one method especially when the number of classes is very large. Multi-class classification problem is much more difficult than the binary one especially when the data are of high dimensionality and the sample size is small. The classification accuracy appears to degrade very rapidly as the number of classes increases [8]. Therefore, feature selection in multi-class classification is more challenging than the one in binary case. We should be more careful when extending feature selection from binary case to multi-class case. Using the statistical relationship between feature ranking and the multiple sub-models of multi-class SVMs, we propose the SVM-DFS method for feature selection.

One-versus-all multi-class SVMs constructs k decision functions where k is the number of classes. The jth decision function $f_j(x)$ is constructed with all of the examples in the jth class with positive labels, and all other examples with negative labels. The $f_j(x)$ is a binary classification sub-model for discriminating the jth class from the all other classes. The r_{ij}, calculated from $f_j(x)$, denotes the feature ranking criterion of the ith feature according to the binary classification sub-model $f_j(x)$. There are sure event E and impossible event $\emptyset$ in probability theory. Let ω_j denote the event that the jth class is true. According to probability theory, events $\omega_1, \omega_2, \ldots, \omega_k$ constitute a partition of the sample space

$$E = \omega_1 \cup \omega_2 \cup \ldots \cup \omega_k \tag{13}$$

and

$$\emptyset = \omega_i \cap \omega_j, \quad i \neq j. \tag{14}$$

$P(\omega_j)$ is the prior probability that the jth class is true. Define a random event S_i as "the ith feature is selected as discriminative feature". Let $P(S_i|\omega_j)$ denote the conditional probability of S_i given that ω_j occurred. When event ω_j occur, the jth binary classification sub-model $f_j(x)$ is just effective for determining the final classification result. Under the jth binary classification sub-model $f_j(x)$, we can calculate $P(S_i|\omega_j)$ through the feature ranking criterion r_{ij}

$$P(S_i|\omega_j) = \frac{r_{ij}}{\Sigma_{t=1}^{n} r_{tj}} \tag{15}$$

According to the theorem on the total probability, $P(S_i)$ can be calculated through $P(S_i|\omega_j)$ and $P(\omega_j)$

$$P(S_i) = \Sigma_{j=1}^{k} P(S_i|\omega_j)P(\omega_j) \tag{16}$$

Then, $P(S_i)$ can be used as feature ranking criterion for the whole multi-class classification problem. The algorithm of SVM-DFS is to rank the features by decreasing the value of $P(S_i)$ and select the top M features as discriminative features. M is the number of the features to be selected. M can be evaluated by retraining the SVM classifiers with the M selected features. M should be set to such a value that the margin Δ_i of the each retrained SVM sub-model $f_i(x)$ is large enough

$$\Delta_i = \frac{1}{\|w^{(i)}\|} \tag{17}$$

where $w^{(i)}$ denotes the weight vector of sub-model $f_i(x)$. According to [15],

$$\|w^{(i)}\|^2 = \sum_{\text{support vectors}} \alpha_j^{(i)} \tag{18}$$

where $\alpha_j^{(i)}$ denotes Lagrange parameter of sub-model $f_i(x)$. Define a coefficient L

$$L = \sum_{i=1}^{k} P(\omega_i) \left(\sum_{\text{support vectors}} \alpha_j^{(i)} \right) \tag{19}$$

We can use coefficient L to evaluate M. M should be set to such a value that the value of L is small enough. After the M discriminative features have been selected through SVM-DFS, the SVM models have to be retrained using the training data for the subsequent classification tasks.

3. Experiments

We have made two sets of experiments to illustrate the effectiveness of the SVM-DFS algorithm. In all experiments reported here, we use the UMIST face database [4], a multi-view database consisting of 575 gray-scale images of 20 subjects, each covering a wide range of poses from profile to frontal views as well as race, gender and appearance. All input images are of size 112×92 and the resulting input vectors are of dimensionality $N = 10304$. Figure.1 depicts some sample images of one subject in the UMIST database. The overall database is partitioned into two subsets: the training set and test set. The training set is composed of 240 images: 12 images per person are carefully chosen according to face poses. The remaining 335 images are used to form the test set. All of the experiments were performed on a 3.0GHz Pentium 4 PC with 1.0 GB RAM. The parameter $C = 10000$ and σ is set to the optimal values in the range $(10 - 1000)$ in SVM training.

Figure 1. Some face samples of one subject from the UMIST face database.

3.1. Comparison with Kernel Subspace Method

To compare SVM-DFS with the traditional kernel subspace method, we use KDDA [10] and our SVM-DFS in the experiments on the same conditions. The experimental settings of KDDA method are adjusted according to [10] and a rbf kernel has been used. In these experiments of SVM-DFS, the number M of the features selected as discriminative features by SVM-DFS is set to 2000 considering the value of the coefficient L and the tradeoff between classifiers' respond time and accuracy. During feature selection, the kernel which the SVM-DFS used is always the linear type.

Table 1. Test results in comparison with KDDA

Methods	Correct rate (%)	Test time (s)
KDDA	95.2239	2.9474
SVM-DFS-1	96.7164	0.0463
SVM-DFS-2	98.2090	1.0431

In Table 1, SVM-DFS-1 denotes that the SVMs of linear kernel are used for the classification after feature selection; SVM-DFS-2 denotes that the SVMs of RBF kernel are used for the classification after feature selection. The test time denotes the sum of all the test time of every individual test sample.

3.2. Performance Study under Various Dimensionality

This set of experiments consists of two subsets: SVM-DFS-L and SVM-DFS-R. Those experiments in SVM-DFS-L use SVMs of linear kernel for classification after SVM-DFS feature selection. Those experiments in SVM-DFS-R use SVMs of RBF kernel for classification after SVM-DFS feature selection. During feature selection, the kernel which the SVM-DFS used is always the linear type.

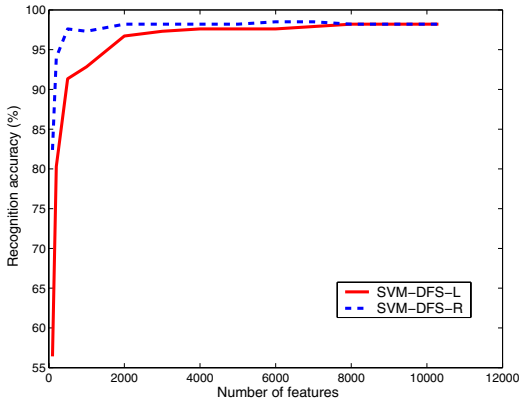

Figure 2. Recognition accuracy under various dimensionality.

Figure.2 shows the recognition accuracy under various dimensionality. Dimensionality, in fact, is the number of the features to be selected. Using SVM-DFS, various number of features have been selected and the accuracy varieties due to dimensionality varieties have been studied. Through Figure.2, We can see that high accuracies are maintained in a wide range of dimensionality, roughly from 2000 to 10304. Furthermore, SVM-DFS-R is more accurate than SVM-DFS-L under low dimensionality. The obvious reason is that SVM-DFS-R has used nonlinear kernel and it can construct complicated nonlinear decision surface for nonlinear classification.

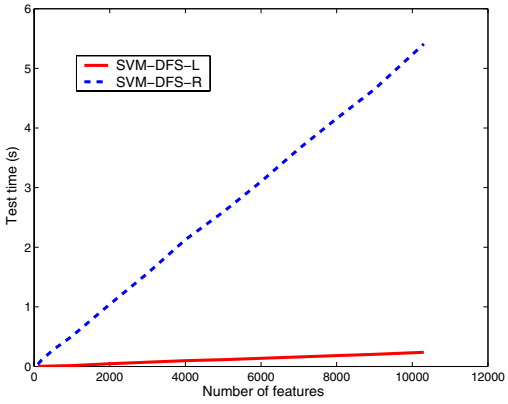

Figure 3. Test time under various dimensionality.

Figure.3 shows the test time under various dimensionality. The test time denotes the sum of all the test time of every individual test sample. Through Figure.3, We can see that the test time is decreasing along with decreasing dimensionality. Especially for SVM-DFS-R, the test time decreases rapidly during the dimensionality decreases.

In Table 2, "Linear" denotes SVM-DFS-L and "RBF" denotes SVM-DFS-R. Table 2 shows the overall test results under various dimensionality. Through these experimental results, we can see that SVM-DFS can speed up the classification process without degrading the correct rate.

Table 2. Test results varying dimensionality

	Correct rate (%)		Test time (s)	
Dimensionality	Linear	RBF	Linear	RBF
10304	98.209	98.209	0.23706	5.40881
9000	98.209	98.209	0.20403	4.65001
8000	98.209	98.209	0.18185	4.15786
7000	97.910	98.508	0.15922	3.64945
6000	97.612	98.508	0.13707	3.10010
5000	97.612	98.209	0.11422	2.59109
4000	97.612	98.209	0.09724	2.12712
3000	97.313	98.209	0.07161	1.56305
2000	96.716	98.209	0.04625	1.04305
1000	92.836	97.313	0.01541	0.50737
500	91.343	97.612	0.00711	0.27816
200	80.299	94.030	0.00295	0.11150
100	56.418	82.388	0.00167	0.03569

4. Conclusions and Future Work

We have presented a discriminative feature selection method SVM-DFS for the challenging task of multi-view face recognition. We integrate the two tasks, feature selection and multi-class classification, into a single consistent framework and effectively realize the goal of discriminative feature selection. Through the experimental results on the UMIST database, we can see that SVM-DFS can speed up the multi-view face recognition process without degrading the correct rate. And through the comparison with KDDA method, it can be seen that SVM-DFS has outperformed the traditional kernel subspace methods. In the future work, we will use wavelet transform for preprocessing before using SVM-DFS. We think that using wavelet rather than raw pixel data can make SVM-DFS more efficient.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60375022 and NSFC 60473040.

References

[1] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition by independent component analysis. *IEEE Trans. Neural Networks*, 13(6):1450–1464, 2002.

[2] P. Belhumeur, J. Hespanda, and D. Kiregeman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] Z. G. Fan, K. A. Wang, and B. L. Lu. Feature selection for fast image classification with support vector machines. *Proc. ICONIP 2004*, LNCS 3316:1026–1031, 2004.

[4] D. B. Graham and N. M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, 163:446–456, 1998.

[5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[6] B. Heisele, T. Serre, S. Prentice, and T. Poggio. Hierarchical classification and feature reduction for fast face detection with support vector machine. *Pattern Recognition*, 36:2007–2017, 2003.

[7] C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Networks*, 13(2):415–425, 2002.

[8] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.

[9] B. L. Lu, K. A. Wang, M. Utiyama, and H. Isahara. A part-versus-part method for massively parallel training of support vector machines. *Proc. IJCNN 2004*, pages 735–740, 2004.

[10] J. Lu, K. Plataniotis, and A. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Networks*, 14(1):117–126, 2003.

[11] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33:1771–1782, 2000.

[12] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[13] B. Scholkopf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[15] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 2000.

[16] X. Wang and X. Tang. Unified subspace analysis for face recognition. *Proc. ICCV 2003*, 1:679–686, 2003.