On Optimizing Subclass Discriminant Analysis Using a Pre-clustering Technique*

Sang-Woon Kim

Senior Member, IEEE, Dept. of Computer Science and Engineering, Myongji University, Yongin, 449-728 South Korea
kimsw@mju.ac.kr

Abstract. Subclass Discriminant Analysis (SDA) [10] is a dimensionality reduction method that has been proven to be successful for different types of class distributions. The advantage of SDA is that since it does not treat the class-conditional distributions as uni-modal ones, the nonlinearly separable problems can be handled as linear ones. The problem with this strategy, however, is that to estimate the number of subclasses needed to represent the distribution of each class, i.e., to find out the best partition, all possible solutions should be verified. Therefore, this approach leads to an associated high computational cost. In this paper, we propose a method that optimizes the computational burden of SDA-based classification by simply reducing the number of classes to be examined through choosing a few classes of the training set prior to the execution of SDA. To select the classes to be partitioned, the intra-set distance is employed as a criterion and a k-means clustering is performed to divide them. Our experimental results for an artificial data set and two face databases demonstrate that the processing CPU-time of the optimized SDA could be reduced *dramatically* without sacrificing either the classification accuracy or the computational complexity.

Keywords: Dimensionality Reduction, Subclass Discriminant Analysis, Clustering.

1 Introduction

Even from the infancy of the field of statistical Pattern Recognition (PR), researchers have had to wrestle with the "curse of dimensionality". The literature reports numerous strategies that have been used to tackle this problem. The most well-known one of these is the Principal Component Analysis (PCA) to compute the basis (eigen) vectors by which the class subspaces are spanned, thus retaining the most significant aspects of the structure in the data [1]. While PCA finds components that are efficient for *representation*, the class of Linear Discriminant Analysis (LDA) strategies seek features that are efficient for *discrimination* [1]. Being essentially linear algorithms, neither PCA nor LDA can effectively classify data which is inherently nonlinear. Consequently, LDA-extensions including two-stage LDA [2], direct LDA [3], kernel-based LDA [4], and

* This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2007-313-D00714).

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 292–300, 2008.

other new approaches [5], [6] have been proposed. Beside these, to discover the nonlinear manifold structure, various techniques including LLE (Locally Linear Embedding) [7] and MDA (Mixture Discriminant Analysis) [8], [9] have been proposed.

Recently, to solve the manifold-based problem, Martinez and his co-authors [10] proposed an approach called SDA (Subclass Discriminant Analysis), by which the underlying distribution of each class can be approximated with a mixture of Gaussians [10]. The basic idea is to represent the data samples of each class by a set of subclasses - capturing most of the variance in the data. In SDA, the major problem to be addressed is to determine the optimal number of Gaussians per class, i.e., the number of subclasses. To obtain a good estimate for the number of subclasses needed to represent each class pdf, the authors use a cross-validation test on the training set. The problem with this strategy, however, is that to estimate the number of subclasses needed to represent the distribution of each class, i.e., to find out the best partition, all possible solutions should be verified. Therefore, this approach leads to an associated high computational cost.

In practice, for a given data set of C classes and H subclass divisions per each class, the searching area for the best solution is *dramatically* increased in the order of H^C. Thus, the application of SDA is not allowable for PR in which the number of classes is very large. To overcome this limitation, in this paper, we propose a method that optimizes the computational burden of SDA by simply reducing the number of classes to be examined through choosing a few classes of the training set[1]. Rather than divide *all* classes of a training set, only a few classes of the set are selected to be bisected. To choose the classes to be clustered, the distribution variance of each class can be used as a criterion. To measure the variance of the class, the so-called *intra-set* distance is used. The distance possesses the capability of measuring the unbiased *variances* of components of the given data set. Thus, it has been used to successfully represent the global distribution structure. The above method is a way of reducing the computational burden of SDA without sacrificing the performance of classifiers designed on the subspace.

The main contribution of this paper is to demonstrate that SDA-based classification can be optimized by employing a pre-clustering step. This has been done by performing the clustering technique prior to the SDA process and by demonstrating its strength in terms of the processing CPU-time and the classification accuracy.

2 Subclass Discriminant Analysis (SDA)

The SDA Algorithm: To obtain the subspaces for a given data set $X = \{\boldsymbol{x}_i\}_{i=1}^n \in \Re^d$, in SDA, the following steps are performed. First, the remaining data set left-out $\boldsymbol{x}_i$, $X_i = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \cdots, \boldsymbol{x}_n\}$, is divided into H subclasses and then a projection matrix, V_q, is computed by performing the eigenvalue decomposition of $\Sigma_X^{-1}\Sigma_B V = V\Lambda_X$. Here, Σ_B is the between-subclass scatter matrix obtained with

$$\Sigma_B = \sum_{i=1}^{C-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{C}\sum_{l=1}^{H_k} p_{ij}p_{kl}(\mu_{ij}-\mu_{kl})(\mu_{ij}-\mu_{kl})^T \tag{1}$$

after dividing the data of each class into a set of subclasses, Σ_X is

[1] This strategy has been applied to various applications. An example can be found in [11].

$$\Sigma_X = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \mu)(\boldsymbol{x}_i - \mu)^T, \;\; \mu = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i, \tag{2}$$

and Λ_X is a diagonal matrix of the corresponding eigenvalues. In Eqs. (1) and (2), C is the number of classes, H_i is the number of subclass divisions in class i, and p_{ij} and μ_{ij} are the prior probability and mean of the jth subclass of class i, respectively. Then, the sample $\boldsymbol{x}_i$ is used for testing the divisions. Let $r_{i,H} = 1$ if $\boldsymbol{x}_i$ is correctly classified, otherwise $r_{i,H} = 0$. After repeating the above procedure n times, one for each of the samples, the recognition rate for a fixed value of H is obtained with $R_H = \frac{1}{n}\sum_{i=1}^{n} r_{i,H}$. A formalized SDA algorithm [10] is summarized in the following:

1. Initialization: $R_H = 0, \forall H$.
2. Perform the following steps from $i = 1$ to n by incrementing i per every iteration.
   (a) Generate the training set X_i using the NN clustering[2].
   (b) Compute Σ_X of (2) with X_i.
   (c) Perform the following steps from $H = C$ to tC by incrementing H per every iteration. Here, t is an experimental constant to guarantee that the minimum number of samples per subclass is sufficiently large.
      i. Compute Σ_B using (1).
      ii. Perform the eigenvalue decomposition $\Sigma_X^{-1}\Sigma_B V = V\Lambda_X$.
      iii. Project the data set X_i onto the subspaces of V_q, i.e., $Y_i = V_q^T X_i$.
      iv. If the sample $\boldsymbol{z}_i (= V_q^T \boldsymbol{x}_i)$ is classified correctly, then $R_H = R_H + 1$.
3. Achieve the optimal value of H with $H_o = argmax_H\{R_H\}$.
4. After calculating Σ_X and Σ_B using $X$and H_o, obtain the final projection matrix V_q^* given by the first q columns of V, where ${\Sigma_X}^{-1}\Sigma_B V = V\Lambda_X$.

In the above algorithm, the optimal Σ_B^* can be computed with H_o. The resulting projection matrix V_q^* is $d \times q$ matrix whose columns are the eigenvectors associated with the q largest eigenvalues when $H = H_o$, while the dimensionality of the sample vectors is d. Thus, the dimensionality of the projected vectors is $q(<< d)$.

The Computational Complexity of SDA: The time complexity of SDA can be analyzed as follows: First, let the computation times for the operations of addition (or subtraction), substitution (or comparison), and multiplication (or division) be t_a, t_s, and t_m, respectively. The time required for Step 1 is $t_1 = Ht_s$. The time needed for Step 2 is a sum of times to compute the three sub-steps of 2(a), 2(b), and 2(c). The times[3] for these sub-steps are: $t_{2(a)} = (Cn_i + CH_i)t_a + (Cn_i^3 + CH_i)t_s + (Cn_i^2 d + CH_i)t_m$, $t_{2(b)} = d(n-1)t_a + dnt_m$, $t_{2(c)} = \left(nH_i + \frac{1}{2}C^2 H_i dn_i\right)t_a + \left(\frac{1}{4}C^2 H_i^3 + nH_i\right)t_s + \left(C^2 H_i^2 d^2\right)t_m + 8d^3 + q(n-1)(dt_m + (d-1)t_a) + (n-1)(qt_m + qt_a + 3t_s) + t_a + t_s$. The four sub-steps from 2(c) i. to iv. should be repeated at least $t\ (= C^{-1}H_i^C)$ times. Following these iterations, the outer sub-steps of (a), (b), (c) are repeated n times again. Thus, when the number of samples, n, is increased, the time complexity of Step 2 will also increase and become significant. Next, the time used for Step 3 to achieve the optimal value H is $t_3 = (n-1)t_s$. Finally, the time consumed for

[2] In this clustering, the vectors in class i are sorted and divided into H_i subclasses. The details of the algorithm are omitted here in the interest of compactness, but can be found in [10].

[3] Here, the time complexity for the eigenvalue decomposition is that of $O(8d^3)$.

Step 4 is: $t_4 = (Cn_i + CH_i)t_a + (Cn_i^3 + CH_i)t_s + (Cn_i^2 d + CH_i)t_m + d(n-1)t_a + dnt_m + (nH_i + \frac{1}{2}C^2 H_i dn_i)t_a + (\frac{1}{4}C^2 H_i^3 + nH_i)t_s + (C^2 H_i^2 d^2)t_m + 8d^3$. Thus, the total time required for the whole procedure to process *high*-dimensional images (refer to *Experimental Data* in Section 4) under the condition $t_s \leq t_a \leq t_m$ is $t_{SDA} \simeq n^2(n_i d + tqd) + ntC^2 H_i^2 d^2 + 8ntd^3)t_m$. From the above analysis, the reader can observe that the time complexity of SDA is $O(ntC^2 H_i^2 d^2 + 8ntd^3)$ and the required time primarily depends on the parameters of n, d, and C. The space complexity of SDA is $O(d^2 + dn)$. (Details of the analysis are omitted here in the interest of compactness.)

3 Optimizing SDA-Based Classification

SDA-based Classification: A SDA-based classification is summarized in the following:

1. Obtain the projection matrix, V_q^*, from the training samples, T, with the SDA algorithm using C, H_i, and t as the input parameters.
2. Project the data set T into a feature space by using V_q^*. To test a sample $\boldsymbol{z}$, compute a feature vector, $\boldsymbol{z}'$, using the same matrix.
3. Achieve a classification based on invoking a classifier built in the transformed space and operating on the vector of $\boldsymbol{z}'$.

In the above algorithm, most of the processing CPU-times are consumed to find the best partition from the solution space. For example, consider a classification task of $C = 2$ and $H_i = 2, \forall i$, i.e., *two*-class and bimodal distribution. In this case, the best partition is found from $\{(1,1),(1,2),(2,1),(2,2)\}$, where h_1 and h_2 of the notation $\{(h_1, h_2)\}$ are the numbers of subclass divisions in class C_1 and C_2, respectively. Similarly, for the classification task of $C = 3$ and $H_i = 2, \forall i$, the best partition is chosen from $\{\ (1,1,1),(1,1,2),(1,2,1),(1,2,2),(2,1,1),(2,1,2),(2,2,1),(2,2,2)\ \}$. From these considerations, it should be clear that the searching area for the best solution is *dramatically* increased by the factor of H_i^C (i.e., 2^2 and 2^3 in the above examples). An approach to overcome this problem is to reduce the number of classes to be evaluated for division from C to C' ($\leq C$). Rather than divide *all* classes of the training set T, in this paper, a few of the classes are bisected. To choose the classes to be clustered, the distribution variance of each class can be used as a criterion. To measure the variance of the class, the so-called intra-set distance is used.

Intra-set Distance: Assume that $T = \{\boldsymbol{x}_i\}_{i=1}^n \in \Re^d$ is a labeled data set so that T can be decomposed into C disjoint subsets $\{T_1, \cdots, T_C\}$, $T_i = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{n_i}\}, n = \sum_{i=1}^C n_i$. Then, a criterion associated with T_i is defined as follows: For a pattern $\boldsymbol{x}_j \in T_i$, the mean of $d(\boldsymbol{x}_j, T_i - \{\boldsymbol{x}_j\})$ over T_i is called the *intra-set distance* of T_i, and is denoted as $D^2(T_i) = \frac{1}{n_i(n_i-1)} \sum_{j=1}^{n_i} \sum_{l=1}^{n_i} \sum_{k=1}^{d} (x_{jk} - x_{lk})^2$. By conveniently rearranging the elements in the triple summation and considering the relations of $\overline{x_{jk}} = \overline{x_{lk}}$ and $\overline{(x_{jk})^2} = \overline{(x_{lk})^2}$ for arbitrary j and l, the intra-set distance can be expressed in terms of the unbiased *variances* of components of the given patterns as follows: $D^2(T_i) = 2\sum_{k=1}^d \sigma_k^2$, where $\sigma_k^2 = \frac{n_i}{n_i - 1}\left\{\overline{(x_{jk})^2} - \left(\overline{x_{jk}}\right)^2\right\}$, $\forall \boldsymbol{x}_j \in T_i$. This is the

rationale of the scheme for employing the intra-set distance as a criterion to select the classes to be clustered.

Optimized SDA-based Classification: In SDA, to find the optimal division of each class, Zhu and Martinez [10] propose two criteria, namely, the Leave-One-Out-Test criterion and the Stability criterion. The first solution uses the leave-one-out test to find the most convenient division. Here, the samples left out are used to determine the subdivision that works best. Although this is a reasonably good approach, it comes with an associated high computational cost [10]. In this paper, to choose the class to be evaluated, first, the intra-set distances for all classes are computed. Then, the class which has the largest distance among the objects is divided into two or more sub-classes. The proposed approach, which is referred to as a Optimized SDA-based (OSDA) classification, is summarized in the following:

1. Compute the intra-set distances of the training data set T_i for all i, $1 \leq i \leq C$, and sort them as: $D^2(T_1) \geq D^2(T_2) \geq \cdots \geq D^2(T_C)$. Then, set $C' = \rho$.
2. Obtain the projection matrix, V_q^*, from the training samples, T, with the SDA algorithm using C', H_i, and t as the input parameters.
3. This step is the same as Step 2 in the SDA-based classification algorithm.
4. This step is the same as Step 3 in the SDA-based classification algorithm.

In Step 1 of the above algorithm, the threshold value ρ is determined experimentally. More importantly, the number of iterations of the step is determined with C', not C. Thus, the time complexity of OSDA is $O(ntC'^2H_i^2d^2 + 8ntd^3)$. Then, the space complexity of OSDA is the same as that of SDA, namely, $O(d^2 + dn)$.

4 Experimental Results: Artificial/Real-Life Data Sets

Experimental Data: The proposed method has been tested and compared with conventional methods. This was done by performing experiments on an artificial data set (which is named as XOR4) and two well-known benchmark face databases, namely, the AT&T[4] and Yale[5] databases. The data set named "XOR4", which has been included in the experiments as a baseline data set, was generated from a mixture of four 4-dimensional Gaussian distributions as follows: $p_1(x) = \frac{1}{2}N(\mu_{11}, I_4) + \frac{1}{2}N(\mu_{12}, I_4)$ and $p_2(x) = \frac{1}{2}N(\mu_{21}, I_4) + \frac{1}{2}N(\mu_{22}, I_4)$, where $\mu_{11} = [-2, -2, 0, 0]$, $\mu_{12} = [2, 2, 0, 0]$, $\mu_{21} = [2, -2, 0, 0]$, and $\mu_{22} = [-2, 2, 0, 0]$. Also, I_4 is the 4-dimensional *Identity* matrix. Here, it is clear that each class contains *two* clusters. Thus, this case is better treated as a *four*-class problem rather than a *two*-class one. The face database captioned "AT&T" consists of ten different images of 40 distinct subjects for a total of 400 images. "Yale" contains 165 gray scale images of 15 individuals.

Experimental Method: In this paper, all experiments were performed using a "leave-one-out" strategy. To classify an image of an object, that image was removed from the training set and the project matrix was computed with the $n-1$ images. After repeating this n times for every sample, a final result was obtained by averaging them. In this

[4] http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html
[5] http://www1.cs.columbia.edu/ belhumeur/pub/images/yalefaces

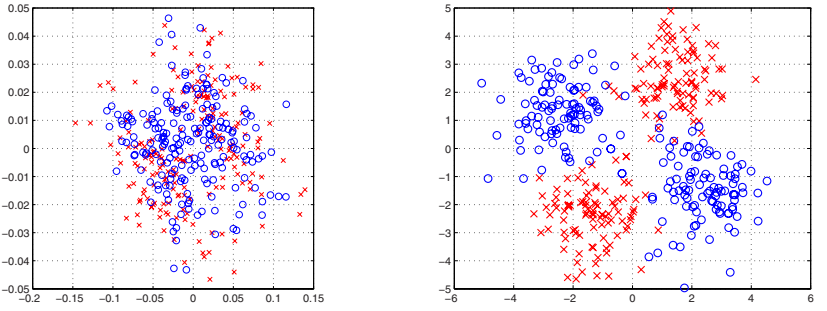

Fig. 1. Plots of the 2-dimensional data set projected from the original 4-dimensional XOR4: (a) left and (b) right. (a) is reduced with the direct LDA [3] and (b) is obtained with the optimized SDA method, in which the reduction of dimensionality is carried out after doing a pre-selection.

experiment, to cluster the training samples of the selected classes, a k-means clustering algorithm was utilized (of course, other clustering methods could also be considered). After computing intra-set distances for the data samples of each class, the class with the largest intra-set distance was clustered first. Then, to simplify the classification task for the paper, only three approaches, namely, LDA [2], PCA+LDA [2], and direct LDA [3] were invoked to evaluate the optimized SDA in terms of classification accuracy[6]. After reducing the dimensionality, classification was performed by invoking the k-Nearest Neighbor Classifier (`knnc`) implemented with PRTools[7].

Experimental Results: The run-time characteristics of the proposed algorithm for the experimental data are reported below and shown in Tables 1 and 2. First, the results of the dimensionality reduction obtained with the proposed SDA scheme in Section 3 were probed into. Fig. 1 shows plots of two 2-dimensional data sets projected from XOR4. In the conventional scheme, the direct LDA [3] was applied to XOR4, while OSDA was carried out after doing a pre-selection for the data set in the proposed method. From the figure, it should be observed that the accuracy of the dimensionality reduction step for the artificial data set can be improved by employing the philosophy of a partition. This is clearly shown in the classification boundary built between the two classes ($\times$, $\circ$) in both pictures. This characteristic could also be observed from the real benchmark face databases. The details are omitted here in the interest of compactness.

Secondly, as the main results, to examine the rationality of employing the pre-selection technique in SDA, the processing CPU-time required to compute the projection matrix, V_q^*, was experimented. Table 1 shows a comparison of the processing CPU-times (in seconds) of SDA and OSDA for the experimental data sets.

From Table 1, the reader can see that the processing CPU-time of OSDA can be reduced significantly by merely employing the pre-selection philosophy. Indeed, this is achieved without sacrificing the classification accuracy. Consider the Big-oh

[6] To maintain the diversity, it would also be desirable to do an experimental comparison with the latest approaches [5], [6] instead of the ones in [2] and [3]. This is currently being investigated.

[7] PRTools (http://www.prtools.org/) is a MATLAB toolbox for pattern recognition.

Table 1. A comparison of the processing CPU-times (in seconds) of SDA and OSDA (Optimized SDA). The details of the table are discussed in the text.

Experimental Data	Parameters					Big-oh Computation		Experimental Measures	
	n	d	C	C'	H_i	SDA	OSDA	SDA	OSDA
XOR4	400	4	2	2	2	6.14×10^5	6.14×10^5	7.93×10^1	7.80×10^1
AT&T	400	2756	40	1	2	1.97×10^{24}	1.09×10^{14}	4.90×10^{14}	9.13×10^5
Yale	165	4880	15	3	2	3.43×10^{17}	4.09×10^{14}	2.88×10^{10}	7.01×10^6

Table 2. A comparison of the classification accuracies (%) of LDA-based and SDA-based classifiers (k-NN classifiers). The details of the table are discussed in the text.

Experimental Data	Input Space Classification	LDA-based Classification			SDA-based Classification		
		LDA	PCA+LDA	direct LDA	$H_i = 1$	$H_i = 2$	Intra-set D
XOR4	89.25	53.25	60.75	52.25	53.50	**92.50**	**92.50**
AT&T	97.75	97.75	95.75	99.00	**99.25**	98.00	99.00
Yale	79.39	91.52	93.33	89.70	**98.18**	**98.18**	**98.18**

computation column for Yale. If the 165 4880-dimensional samples of 15 objects were processed with SDA, the time complexity computed is 3.43×10^{17}, where each class was divided into two subclasses. However, if only three objects were selected from the fifteen classes and evaluated for divisions, the time complexity obtained is 4.09×10^{14}. From this consideration, the reader should observe that if the optimization of pre-selecting the objects to be evaluated can be done before performing SDA, the time required is only *about one-thousandth* of the time that the original SDA would take.

Then, to highlight the advantage, experimental measures for the experimental data were compared. Consider again the processing CPU-times (in seconds) for Yale. Here, the processing times of both SDA and OSDA are numerically estimated as follows: The times for obtaining V_q^* is $n \times t \times \gamma$, where n is the number of samples, t is the number of iterations for dividing each class into *two* subclasses, and γ is the time for computing Step 2(c) of the SDA algorithm. For example, in the case of Yale, $n = 165$, $t = 2^{15}$ (or 2^3 for OSDA), and 5309.45 seconds[8]. Using these values of the parameters, the processing CPU-times for SDA and OSDA can be computed as 2.88×10^{10} and 7.01×10^6, respectively. From this consideration, the reader should notice that the time required for OSDA is only *a small fraction* of the time which SDA would take. Even if the near-optimal selections were done, the advantage would undoubtedly be very significant. Identical comments can also be made for the other databases.

Finally, it should be noted that it is possible to keep almost the same classification performance even though only a few classes, not all of them, are chosen to be evaluated for division by employing the philosophy of pre-selection. Table 2 shows an experimental comparison of the classification performances of LDA-based and SDA-based classifiers. In LDA-based classification, the dimensionality reduction has been done with LDA, PCA+LDA, and direct LDA. In SDA, two types of SDA are performed:

[8] This time is obtained with a simulation on a Windows platform (CPU: 2.40GHz, RAM: 2GB).

(1) without division, i.e., $H_i = 1, \forall i$ and (2) with division such that each of all classes is divided into two subclasses, i.e., $H_i = 2, \forall i$. On the other hand, in the optimized SDA (this is the proposed version), a few classes selected prior to the SDA step are divided into two subclasses.

From Table 2, it is clear that the classification accuracies can be improved by employing the pre-clustering technique (see the bold-faced ones). The details are omitted here in the interest of compactness.

From the table, however, it should also be mentioned that the classification accuracies of Yale are much more increased compared to those of AT&T. This result seems to originate from the fact that each image of Yale has bigger variations than that of AT&T in their illumination, facial expression, and background. From this consideration, it can be mentioned that the proposed scheme is useful in capturing the nonlinear structure.

5 Conclusions

In this paper, a method that seeks to optimize Subclass Discriminant Analysis (SDA) has been considered. The method involves a class-pre-selecting step prior to the execution of SDA to find out the classes of a mixture of Gaussians and divide them into a set of subclasses. The experimental results for an artificial data set and two well-known face databases demonstrate that the proposed scheme works well without sacrificing the classification accuracy rates. Even though an investigation has been made, focusing on the possibility of the pre-selecting technique being used to solve the computational burden problem of SDA, many problems remain. The classification performance could be improved by developing an optimal selection method and by designing suitable classifiers in the subclass space. The research concerning this is a future aim of the authors.

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, San Diego (1990)
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 19(7), 711–720 (1997)
3. Yu, H., Yang, J.: A direct LDA algorithm for high-dimensional data - with application to face recognition. Pattern Recognition 34, 2067–2070 (2001)
4. Yang, M.-H.: Kernel Eigenfaces vs. kernel Fisherfaces: Face recognition using kernel methods. In: Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 215–220 (2002)
5. Loog, M., Duin, R.P.W.: Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Cherno Criterion. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 26(6), 732–739 (2004)
6. Rueda, L., Herrera, M.: A New approach to multi-class linear dimensionality reduction. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 634–643. Springer, Heidelberg (2006)
7. Roweis, S., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)

8. Frley, C., Raftery, A.E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. The Computer Journal 41(8), 578–588 (1998)
9. Halbe, Z., Aladjem, M.: Model-based mixture discriminant analysis - An experimental study. Pattern Recognition 38, 437–440 (2005)
10. Zhu, M., Martinez, A.M.: Subclass discriminant analysis. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 28(8), 1274–1286 (2006)
11. Kim, S.-W., Duin, R.P.W.: On using a pre-clustering technique to optimize LDA-based classifiers for appearance-based face recognition. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 466–476. Springer, Heidelberg (2007)