On Using Dimensionality Reduction Schemes to Optimize Dissimilarity-Based Classifiers*

Sang-Woon Kim and Jian Gao

Dept. of Computer Science and Engineering, Myongji University, Yongin, 449-728 South Korea
{kimsw,marsgao}@mju.ac.kr

Abstract. The aim of this paper is to present a strategy by which a new philosophy for pattern classification pertaining to dissimilarity-based classifiers (DBCs) can be efficiently implemented. Proposed by Duin and his co-authors, DBCs are a way of defining classifiers among classes; they are not based on the feature measurements of individual patterns, but rather on a suitable dissimilarity measure among the patterns. The problem with this strategy is that we need to select a representative set of data that is both compact and capable of representing the entire data set. However, it is difficult to find the optimal number of prototypes and, furthermore, selecting prototype stage may potentially lose some useful information for discrimination. To avoid these problems, in this paper, we propose an alternative approach where we use *all* available samples from the training set as prototypes and subsequently apply dimensionality reduction schemes. That is, we prefer not to directly select the representative prototypes from the training samples; rather, we use a dimensionality reduction scheme after computing the dissimilarity matrix with the *entire* training samples. Our experimental results demonstrate that the proposed mechanism can improve the classification accuracy of conventional approaches for two real-life benchmark databases.

Keywords: Dissimilarity Representation, Dissimilarity-based Classification, Dimensionality Reduction Schemes, Appearance-based Face Recognition.

1 Introduction

One of the most recent and novel developments in the field of statistical pattern recognition (PR) [1] is the concept of dissimilarity-based classifiers (DBCs) proposed by Duin and his co-authors [2]. DBCs are a way of defining classifiers among the classes; and the process is not based on the feature measurements of individual patterns, but rather on a suitable dissimilarity measure among the individual patterns. The major questions we encountered when designing DBCs are summarized as follows: (1) how to select prototypes; (2) how to measure dissimilarities between object samples; and (3) how to design classifiers in the dissimilarity space. Several strategies have been used to explore these questions [3], [4], [5]. The details of the strategies are omitted here, but we now attempt to explain the first question in the present paper.

Various methods have been proposed in the literature as a means of selecting a representative set of data that is both compact and capable of representing the entire data

* This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2007-211-D00109).

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 309–316, 2008.

set. To select the representative set, Duin and his colleagues [3] discussed the following methods : *Random*, *RandomC*, *KCentres*, *ModeSeek*, *LinProg*, *FeatSeal*, *KCentres-LP*, and *EdiCon*. In these methods, a training set, T, is pruned to yield a set of representative prototypes, Y, where, without loss of generality, $|Y| \leq |T|$. On the other hand, by invoking a prototype reduction scheme (PRS), Kim and Oommen [5] also obtained a representative subset, Y, which is utilized by the DBC. Aside from using PRSs, Kim and Oommen simultaneously proposed the use of the Mahalanobis distance as the dissimilarity-measurement criterion. With that criterion, they were able to increase the classification accuracy of DBCs by using the second-order properties of the data.

In DBCs, a good selection of prototypes seems to be crucial to succeed with the classification algorithm in the dissimilarity space. The prototypes should avoid redundancies in terms of selection of similar samples, and prototypes should include as much information as possible. However, it is difficult for us to find the optimal number of prototypes. Furthermore, there is a possibility that we lose some useful information for discrimination when selecting the prototypes [6],[7]. To avoid these problems, we propose an alternative approach where we use *all* available samples from the training set as prototypes (i.e., $Y = T$) and subsequently apply dimensionality reduction schemes. That is, we prefer not to directly select the representative prototypes from the training samples; rather, we use dimensionality reduction schemes after computing the dissimilarity matrix with the *entire* training samples This approach is more principled and allows us to avoid the problem of finding the optimal prototype selection strategy [7].

The main contribution of this paper is to demonstrate that dissimilarity-based classification can be optimized by employing a dimensionality reduction scheme. This has been done by performing the reduction technique after computing the dissimilarity matrix with the *entire* training samples. Here, the dimensionality reduction scheme is used to accommodate some useful information for discrimination and to avoid the problem of finding the optimal prototype selection strategy. The remainder of the paper is organized as follows: In Section 2, we present a brief overview of dissimilarity representation and dimensionality reduction and a schema for the proposed solution. In Section 3, we present the experimental results of two real-life benchmark databases. In Section 4, we present our concluding remarks.

2 Optimizing DBCs with DRS

Foundations of DBCs: A dissimilarity representation of a set of samples, $T = \{\boldsymbol{x}_i\}_{i=1}^{n} \in \Re^d$, is based on pairwise comparisons and is expressed, for example, as an $n \times m$ dissimilarity matrix $D_{T,Y}[\cdot,\cdot]$, where $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m\}$, a prototype set, is extracted from T and the subscripts of D represent the set of elements on which the dissimilarities are evaluated. Thus each entry $D_{T,Y}[i,j]$ corresponds to the dissimilarity between the pairs of objects $\langle \boldsymbol{x}_i, \boldsymbol{y}_j \rangle$, where $\boldsymbol{x}_i \in T$ and $\boldsymbol{y}_j \in Y$. Consequently, an object $\boldsymbol{x}_i$ is represented as a column vector as follows:

$$[d(\boldsymbol{x}_i, \boldsymbol{y}_1), d(\boldsymbol{x}_i, \boldsymbol{y}_2), \cdots, d(\boldsymbol{x}_i, \boldsymbol{y}_m)]^T, 1 \leq i \leq n. \quad (1)$$

Here, the dissimilarity matrix $D_{T,Y}[\cdot,\cdot]$ is defined as a *dissimilarity space* on which the d-dimensional object, $\boldsymbol{x}$, given in the feature space, is represented as an m-dimensional vector $\delta(\boldsymbol{x}, Y)$, where if $\boldsymbol{x} = \boldsymbol{x}_i$, $\delta(\boldsymbol{x}_i, Y)$ is the i-th row of $D_{T,Y}[\cdot,\cdot]$. In this paper, the column vector $\delta(\boldsymbol{x}, Y)$ is simply denoted by $\delta_Y(\boldsymbol{x})$.

A conventional algorithm for DBCs is summarized in the following:

1. Select the representative set, Y, from the training set, T, by resorting to one of the prototype selection methods as described in [3], [5].

2. Using Eq. (1), compute the dissimilarity matrix, $D_{T,Y}[\cdot,\cdot]$, in which each individual dissimilarity is computed on the basis of the measures described in [3], [5].

3. For a testing sample, $\boldsymbol{z}$, compute a dissimilarity column vector, $\delta_Y(\boldsymbol{z})$, by using the same measure used in Step 2.

4. Achieve the classification by invoking a classifier built in the dissimilarity space and by operating the classifier on the dissimilarity vector, $\delta_Y(\boldsymbol{z})$.

From these four steps, we can see that the performance of the DBCs relies heavily on how well the dissimilarity space, which is determined by the dissimilarity matrix, $D_{T,Y}[\cdot,\cdot]$, is constructed. To improve the performance, we need to ensure that the dissimilarity matrix is well designed.

Dimensionality Reduction Schemes: Various strategies have been used to tackle the “dimensionality reduction” problem (some of them are [8], [10], [11], [12], [13], [14], [15], [16], [17], and [18]). To optimize DBCs, in this paper, we use a strategy of reducing the dimensionality after computing the dissimilarity matrix. With regard to reducing the dimensionality of the dissimilarity matrix, we make use of the well-known dimensionality reduction schemes (DRSs) proposed in the literature. In the interest of completeness, we now offer a brief introduction of DRSs[1]. 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, numerous LDA-extensions including two-stage LDA [8], direct LDA [10], kernel-based LDA [11], discriminative common vectors (DCV) [12], and other new approaches [13], [14], [15] have been proposed in the literature. Beside these, to discover the nonlinear manifold structure, various techniques including LLE (Locally Linear Embedding) [16], LLDA (Locally Linear Discriminant Analysis) [17], and MDA (Mixture Discriminant Analysis) and its variants [18], [19] have been proposed. The details of these methods are omitted here in the interest of compactness, but can be found in the literature.

Schema for the Proposed Solution: As mentioned earlier, there are several ways by which the classification efficiency of DBCs can be optimized. In our method of optimizing DBCs, we use a strategy of reducing the dimensionality after computing the dissimilarity matrix with the entire training samples. The basic strategy is to solve the classification problem by first computing the dissimilarity matrix with the *entire* training samples and then *reducing* its dimensionality with the DRS; finally, DBCs are designed on the dissimilarity space to reduce the classification error rates.

[1] Our overview is necessarily brief, but additional details can be found in [1], [8], [10], and [15].

An optimized algorithm for DBCs is summarized in the following:

1. Select the entire training samples T as the representative set Y.

2. Using Eq. (1), compute the dissimilarity matrix, $D_{T,T}[\cdot,\cdot]$, in which each individual dissimilarity is computed on the basis of the measures described in [2], [5]. After computing the $D_{T,T}[\cdot,\cdot]$, reduce its dimensionality by invoking a DRS.

3. This step is the same as Step 3 in DBC (see the previous section).

4. This step is the same as Step 4 in DBC.

The rationale of this strategy is presented in a later section together with the experimental results.

3 Experimental Results

Experimental Data: The proposed strategy was tested and compared with conventional methods by conducting experiments on the two well-known benchmark databases "AT&T" and[2], "Yale"[3]. The face database of AT&T, formerly known as the ORL database of faces, consists of ten different images of 40 distinct subjects for a total of 400 images. The size of each image is 112×92 pixels for a total dimensionality of 10304. The face database termed as Yale contains 165 gray scale images of 15 individuals. The size of each image is 243×320 pixels for a total dimensionality of 77760. In this experiment, to reduce the computational complexity, facial images of AT&T and Yale databases were down-sampled into 56×46 and 61×80, respectively, and then represented by a centered vector of normalized intensity values.

Experimental Method: All our experiments were performed with a "leave-one-out" strategy. To classify an image, we removed the image from the training set and computed the dissimilarity matrix with the $n-1$ images. This process was repeated n times for every image, and a final result was obtained by averaging the results of each image.

To construct the dissimilarity matrix, we first selected all training samples as the representative set. We then measured the dissimilarities between each sample and the prototypes. For this measurement, we used a conventional measurement system, such as Euclidean distance (ED), Hamming distance (HD), regional distance (RD) [9], or spatially weighted gray-level Hausdorff distance (WGHD) [4]. After computing the dissimilarity matrix, we reduced the dimensionality of the matrix with a DRS, such as PCA [8], direct LDA [10], PCA-plus- LDA [8], LDA-plus-KFT [15], or DCV [12]. In a subsequent section these systems are named as PCA, LDA, PCALDA, LDAKFT, and DCV, respectively. In the PCA, LDA, LDAKFT, and DCV approaches, we reduced the dimension $n-1$ to $c-1$, where n is the total number of training samples and c is the number of classes. In the PCALDA method, we reduced the dimensionality in two steps: first we reduced the dimension $n-1$ into an intermediate dimension $n-c$ using PCA; we then reduced the $n-c$ to $c-1$ using LDA[4]. In the conventional methods of *Random*, *RandomC*, *KCentres*, and *ModeSeek*, on the other hand, we selected $c-1$ samples from the training data set as the prototypes of DBCs.

[2] http://www.uk.research.att.com/facedatabase.html

[3] http://www1.cs.columbia.edu/ belhumeur/pub/images/yalefaces

[4] Similar to the approaches with prototype selection methods, the number of dimensions is not given beforehand. Thus, the problem of selecting the optimal dimension remains unresolved.

To maintain the diversity between the dissimilarity-based classifications, we designed different classifiers, such as the k-nearest neighbor classifiers (k = 1, 3, 5, 7), the nearest mean classifiers, the support vector classifier, and the regularized normal density-based linear/quadratic classifiers. These classifiers, which were implemented with PRTools[5], are denoted in the next section as 1-NN, 3-NN, 5-NN, 7-NN, NMC, SVC, RLDC, and RQDC, respectively. Here, SVC is a support vector classifier that employs the most widely used RBF kernel function.

Experimental Results: The run-time characteristics of the proposed strategy for AT&T and Yale are reported below. The classification accuracy rates (%) of the DBCs are first illustrated in graphs. A numerical comparison of the processing CPU-times (seconds) is then made in relation to the conventional methods and the proposed strategy.

Figure 1 shows a comparison of the classification accuracy rates (%) for the AT&T and Yale databases. These pictures confirm the possibility of improving the performance of DBCs by effectively reducing the dimensionality. The improvement can be seen by observing how the classification accuracy rates (%) change. For example, in Figure 1 (a), (b), (c), and (d), for almost all the nonparametric classifiers, namely 1-NN, 3-NN, 5-NN, 7-NN, NMC, and SVC, the classification results of the proposed reducing methods of PCA, LDA, PCALDA, LDAFKT, and DCV (which are marked as $\circ$, $\times$, $+$, $*$, and $\diamond$, respectively) are *significantly* more accurate than those of the conventional reducing methods of *Random*, *RandomC*, *KCentres*, and *ModeSeek* (which are indicated as $\triangle$, $\bigtriangledown$, $\triangleleft$, and $\triangleright$, respectively)[6]. For the parametric classifiers RLDC and RQDC, the classification results of the proposed strategy are also *marginally* accurate than those of the conventional methods. The same trend is evident in Figure 1 (e), (f), (g), and (h), which were obtained with the Yale database. The description of the results is omitted here to avoid repetition. However, with the proposed strategy, some classifiers failed to improve their classification accuracies. The problem of theoretically analyzing this observation remains unresolved.

In general, increasing the cardinality of the representative subset improves the average classification accuracy of the resultant DBCs. To further investigate the advantage of using the proposed strategy, we repeated the above experiment again for the benchmark databases. However, the DBCs were designed in the dissimilarity matrices constructed with the cardinality of $2c$, not $c - 1$. The details of the experimental results are omitted here in the interest of compactness, but we observed the same characteristics as in Figure 1. Although the dimensionality of the dissimilarity matrix increase by two times, for all the nonparametric classifiers, namely 1-NN, 3-NN, 5-NN, 7-NN, NMC, and SVC, the classification results of the proposed strategy are more accurate than (or, for some classifiers, almost the same as) those of the conventional methods.

Using the whole training set as prototypes leads to higher computational complexity as more distances have to be calculated. In comparing the conventional and new schemes, rather than embark on yet another analysis of the computational complexity of the latter, we simply measured the processing CPU-times (seconds) of the DBCs for the real benchmark databases. Table 1 shows a comparison of the averaged processing

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

[6] In *Wholeset* method, the entire training data set T is selected as a representative subset Y. The result of the method (which is identified as a hexagonal symbol) is included as a reference.

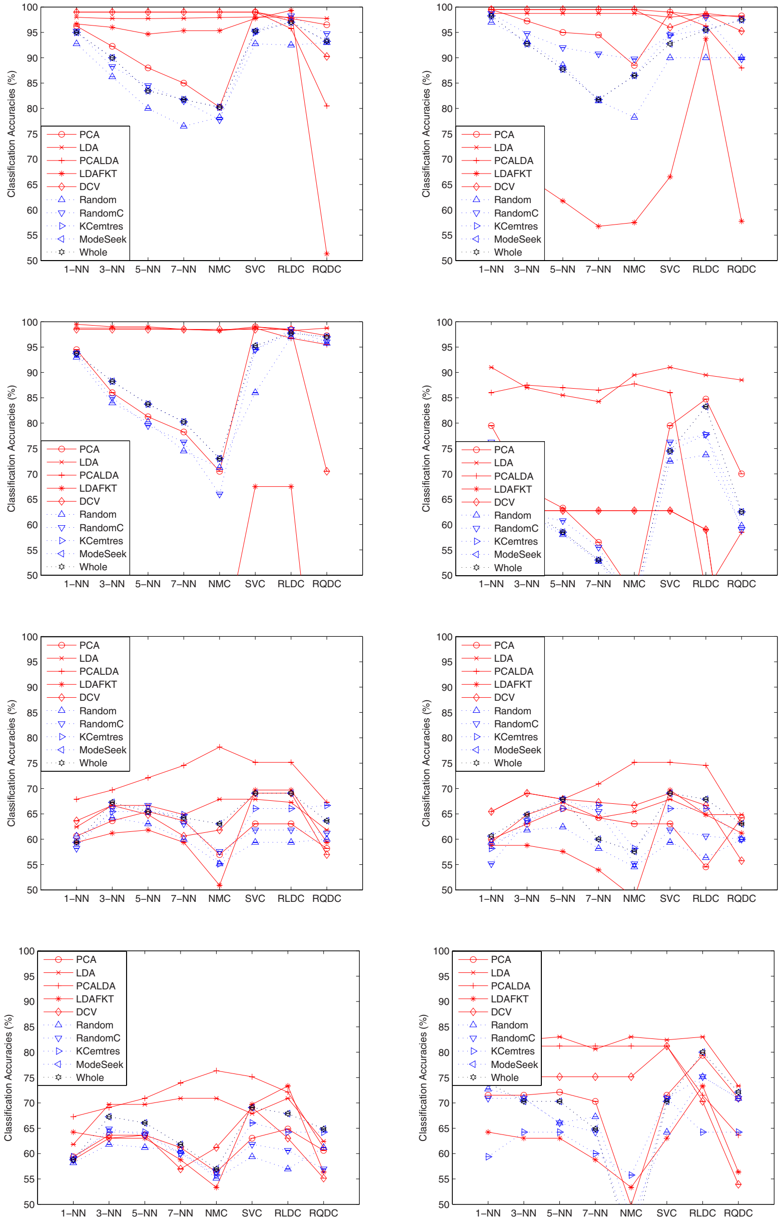

Fig. 1. A comparison of the classification accuracies for the AT&T and Yale databases: (a) top left, (b) top right, · · ·, (g) bottom left, and (h) bottom right; (a) - (d) are for AT&T and (e) - (h) are for Yale. The measuring methods of ED, HD, RD, and WGHD have been used to construct, respectively, the dissimilarity matrices of (a) and (e), (b) and (f), (c) and (g), and (d) and (h).

Table 1. A comparison of the averaged processing CPU-times (seconds) of DBCs for the AT&T and Yale databases. Each number of the table is obtained by averaging the results of five iterations on a Windows platform (CPU: 2.40 GHz, RAM: 2GB).

Experimental	AT&T				Yale			
Methods	ED	HD	RD	WGHD	ED	HD	RD	WGHD
PCA	1893.62	1899.91	1635.32	1589.48	37.17	38.19	36.16	43.10
LDA	195.50	239.88	334.60	360.62	3.68	3.73	3.65	3.67
PCALDA	3020.88	2957.91	2872.92	2965.47	28.80	28.42	28.11	42.76
LDAFKT	1373.62	1407.51	1406.44	1387.25	15.41	15.36	15.82	32.30
DCV	743.25	769.88	726.86	746.91	14.49	14.55	13.69	15.10
Random	3.83	3.92	3.96	3.96	0.16	0.16	0.19	0.16
RandomC	4.79	4.80	5.02	5.01	0.35	0.36	0.34	0.36
KCentres	344.70	336.32	377.30	377.05	27.52	30.36	29.62	39.83
ModeSeek	49.66	53.90	53.54	53.54	6.92	6.92	6.87	6.30

CPU-times (for the process of dimensionality reduction or prototype selection) of DBCs for the AT&T and Yale databases. Table 1 shows that the processing CPU-times (seconds) increased when the proposed technique was applied. An example of this change is the processing times of ED measuring method for AT&T. The processing times of the PCA, LDA, PCALDA, LDAFKT, and DCV methods are, respectively, 1893.62, 195.50, 3020.88, 1373.62, and 743.25, while those of *Random*, *RandomC*, *KCentres*, and *ModeSeek* are, respectively, 3.83, 4.79, 344.70, and 49.66. The same characteristic could also be observed in the HD, RD, WGHD measuring methods. The results of Yale are omitted here again to avoid repetition.

In review, the experimental results show that when the proposed strategy was applied to the dissimilarity representation, the classification accuracies of DBCs increased, but the processing CPU-times also increased. In addition, in terms of classification accuracies, the proposed strategy is clearly more useful for the nonparametric classifiers, such as k-NN and SVC, but not for the parametric classifiers, such as RLDC and RQDC.

4 Conclusions

In our efforts to optimize DBCs, we used dimensionality reduction schemes instead of selecting a representative set of data. Rather than deciding to discard or retain the training points with the prototype selection method, we reduced the dimensionality after computing the dissimilarity matrix with the entire training samples. This approach overcomes the problems caused by finding the optimal number of prototypes. The proposed strategy was tested on two well-known benchmark databases, and the results were compared with the results of conventional methods. The experimental results demonstrate that the proposed strategy is better than conventional methods in terms of classification accuracy. Although we have shown that DBCs can be optimized with our proposed strategy, many tasks remain unchallenged. One of them is to reduce the processing CPU-time by developing a new dimensionality reduction scheme in the dissimilarity space. Our aim is to conduct further research on this subject in the future.

References

1. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 22(1), 4–37 (2000)
2. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications. World Scientific Publishing, Singapore (2005)
3. Pekalska, E., Duin, R.P.W., Paclik, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39, 189–208 (2006)
4. Kim, S.-W.: Optimizing dissimilarity-based classifiers using a newly modified Hausdorff distance. In: Hoffmann, A., Kang, B.-h., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 177–186. Springer, Heidelberg (2006)
5. Kim, S.-W., Oommen, B.J.: On using prototype reduction schemes to optimize dissimilarity-based classification. Pattern Recognition 40, 2946–2957 (2007)
6. Riesen, K., Kilchherr, V., Bunke, H.: Reducing the dimensionality of vector space embeddings of graphs. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 563–573. Springer, Heidelberg (2007)
7. Bunke, H., Riesen, K.: A family of novel graph kernels for structural pattern recognition. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 20–31. Springer, Heidelberg (2007)
8. Belhumeour, 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)
9. Adini, Y., Moses, Y., Ullman, S.: Face Recognition: The problem of compensating for changes in illumination direction. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 19(7), 721–732 (1997)
10. Yu, H., Yang, J.: A direct LDA algorithm for high-dimensional data - with application to face recognition. Pattern Recognition 34, 2067–2070 (2001)
11. Yang, M.-H.: Kernel eigenfaces vs. kernel Fisherfaces: Face recognition using kernel methods. In: Proceedings of 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp. 215–220 (2002)
12. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative common vectors for face recognition. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 27(1), 4–13 (2005)
13. 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)
14. 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)
15. Zhang, S., Sim, T.: Discriminant subspace analysis: A Fukunaga-Koontz approach. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 29(10), 1732–1745 (2007)
16. Roweis, S., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
17. Kim, T.-K., Kittler, J.: Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. IEEE Trans. Pattern Anal. and Machine Intell. PAMI 27(3), 318–327 (2005)
18. 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)
19. Halbe, Z., Aladjem, M.: Model-based mixture discriminant analysis - An experimental study. Pattern Recognition 38, 437–440 (2005)