On Optimizing Kernel-Based Fisher Discriminant Analysis Using Prototype Reduction Schemes*

Sang-Woon Kim[1] and B. John Oommen[2]

[1] Dept. of Computer Science and Engineering, Myongji University, Yongin, 449-728 Korea
kimsw@mju.ac.kr

[2] School of Computer Science, Carleton University, Ottawa, ON, K1S 5B6, Canada
oommen@scs.carleton.ca

Abstract. Fisher's Linear Discriminant Analysis (LDA) is a traditional dimensionality reduction method that has been proven to be successful for decades. Numerous variants, such as the Kernel-based Fisher Discriminant Analysis (KFDA) have been proposed to enhance the LDA's power for nonlinear discriminants. Though effective, the KFDA is computationally expensive, since the complexity increases with the size of the data set. In this paper, we suggest a novel strategy to enhance the computation for an *entire family* of KFDA's. Rather than invoke the KFDA for the entire data set, we advocate that the data be first reduced into a smaller representative subset using a Prototype Reduction Scheme (PRS), and that dimensionality reduction be achieved by invoking a KFDA on *this* reduced data set. In this way data points which are ineffective in the dimension reduction and classification can be eliminated to obtain a significantly reduced kernel matrix, K, without degrading the performance. Our experimental results demonstrate that the proposed mechanism *dramatically* reduces the computation time without sacrificing the classification accuracy for artificial and real-life data sets.

1 Introduction

The "Curse of Dimensionality": Even from the infancy of the field of statistical Pattern Recognition (PR), researchers and practitioners have had to wrestle with the so-called "curse of dimensionality". The situation is actually quite ironic : If the patterns to be recognized are represented in a feature space of small dimensions, it is likely that many crucial discriminating characteristics of the classes are ignored. However, if on the other hand, the dimensions of the feature space are large, we encounter this "curse", which brings along the excess

* The second author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. This work was generously supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2005-042-D00265).

baggage of all the related problems associated with learning, training, representation, computation and classification [1], [2]. The "dimensionality reduction" problem involves reducing the dimension of the input patterns and yields the advantages clearly explained in [1] and [2].

The literature reports numerous strategies that have been used to tackle this problem. The most well-known of these is the Principal Components Analysis (PCA) (the details of which are omitted here) 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 the PCA finds components that are efficient for *representation*, the class of Linear Discriminant Analysis (LDA) strategies seek features that are efficient for *discrimination* [1]. LDA methods effectively use the concept of a within-class scatter matrix, S_w, and a between-class scatter matrix, S_b, to maximize a separation criterion, such as $J = tr(S_w^{-1} S_b)$. The advantage of an LDA is that it is non-recursive. Being essentially linear algorithms, neither the PCA nor LDA can effectively classify data which is inherently nonlinear. Consequently, a vast body of research has gone into resolving this limitation, and a detailed review of this is found in [2]. This is exactly the focus of this paper. In this paper, we suggest a novel strategy to enhance the computation for an *entire family* of KFDA's. Rather than invoke the KFDA for the entire data set, we advocate that the data be first reduced into a smaller representative subset using a Prototype Reduction Scheme (PRS) (explained and briefly surveyed presently), and that dimensionality reduction be achieved by invoking a KFDA on *this* reduced data set.

The state-of-the-art in dealing with nonlinear methods include an adaptive method utilizing a rigorous Gaussian distribution assumption [3], a complete PCA plus LDA algorithm [4], two variations on Fisher's linear discriminant [5], Kernel-based PCA (KPCA) [6], Kernel-based FDA (KFDA) (for two classes by Mika *et al.* [7] and for multi-classes by Baudat and Anouar in [8]), Kernel-based PCA plus Fisher LDA (KPCA+LDA) [9], and LDA extensions which use the Weighted Pairwise Fisher Criteria and the Chernoff Criterion [10].

Methods for Handling Nonlinearity: The KPCA (or KFDA) provides an elegant way of dealing with nonlinear problems in an input space $\mathcal{R}^d$ by mapping them to linear ones in a feature space, F. That is, a dot product in space $\mathcal{R}^d$ corresponds to mapping the data into a possibly high-dimensional dot product space F by a nonlinear map $\Phi : \mathcal{R}^d \rightarrow F$, and taking the dot product in the latter space [6]. All of them utilize the *kernel trick* to obtain the kernel PCA components by solving a linear eigenvalue problem similar to that done for the linear PCA. The only difference is that the size of the problem is decided by the *number* of data points, and not by the dimension. In both the KPCA and KFDA, to map the data set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$, (where each $\boldsymbol{x}_i \in \mathcal{R}^d$) into a feature space F, we have to define an $n \times n$ matrix, K, the so-called kernel matrix, (of dimension n) which is analogous to the $d \times d$ covariance matrix of the linear PCA (or LDA).

To solve the KPCA-associated computational problem, a number of methods, such as the techniques proposed by Achlioptas and his co-authors [11], [12], the power method with deflation [6], the method of estimating K with a subset of the data [6], the Sparse Greedy matrix Approximation (SGA) [13], the Nystom method [14], and the sparse kernel PCA method based on the probabilistic feature-space PCA concept [15], have been proposed. In [6], a method of estimating the matrix K from a subset of $n'(< n)$ data points, while still extracting principal components from all the n data points, was considered. Also, in [13], an approximation technique to construct a compressed kernel matrix K' such that the norm of the residual matrix $K - K'$ is minimized, was proposed. Indeed, pioneering to the area of reducing the complexity of kernel-based PCA methods are the works of Achlioptas [12] and his co-authors. Their first category includes the strategy of artificially introducing sparseness into the kernel matrix, which, in turn, is achieved by *physically* setting some randomly-chosen values to *zero*. The other alternative, as suggested in [11], proposes the elimination of the underlying data points themselves. This is the spirit of the strategy we advocate.

Optimizing KFDA: To solve the computational problem for KFDA methods, a number of schemes, such as the efficient leave-one-out cross-validation method [16], the techniques proposed by Xu and his co-authors [17], [18], and the method of using a minimum squared-error cost function and the orthogonal least squares algorithm [19], have been proposed. They are *briefly* (for space limitations) described below, but the details can be found in [26].

Cawley and Talbot [16] showed that the leave-one-out cross-validation of kernel Fisher discriminant classifiers, namely, $f(\boldsymbol{x}_i) = \boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b$, (where $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i)$), can be implemented with a computational complexity of only $O(n^3)$ operations rather than the $O(n^4)$ complexity of a naive implementation, where n is the number of training samples. Xu and his co-authors [17], [18] proposed a reformative kernel Fisher discriminant method (for two and multiple classes respectively) which only computes the kernel matrix, K, between the test pattern and a *part* of the training samples, called the "significant nodes" (assuming that the eigenvectors for larger nonzero eigenvalues led to superior discriminant vectors) which are a few training samples selected from the entire data set. In [19], Billings and Lee suggested that after selecting n_s important terms, $\{\boldsymbol{x}'_i\}_{i=1}^{n_s}$, from all the training patterns, $\{\boldsymbol{x}_j\}_{i=1}^{n}$, using the orthogonal least squared (OLS) algorithm when one has to test the sample $\boldsymbol{z}$, the authors classified it as class ω_1 if $\sum_{i=1}^{n_s} \alpha'_i k(\boldsymbol{z}, \boldsymbol{x}'_i) > c$, where α'_i are the estimated coefficients; Otherwise it is classified as belonging to class ω_2.

Unlike the results mentioned above in [16], [17], [18] and [19], we propose an alternate strategy, akin to the one suggested in [11] for the KPCA family of algorithms – which is a fairly straightforward concept, yielding a *significant* computational advantage. Quite simply put, we propose to solve the computational problem in KFDA by reducing the *size* of the design set without sacrificing the performance, where the latter is achieved by using a Prototype Reduction Scheme (PRS). Thus, the contribution of this paper is that we show that the computational burden of a KFDA can be reduced significantly by not considering the "original" kernel matrix *at all*. Instead, we rather define a reduced-kernel matrix by first preprocessing the training points with a PRS scheme. Further, the PRS scheme does not necessarily have to *select* a reduced set of data points.

Indeed, it can rather *create* a reduced set of prototypes from which, in turn, the reduced-kernel matrix is determined. All of these concepts are novel to the field of designing Kernel-based FDA methods and have been rigorously tested for artificial and real-life data sets.

Prototype Reduction Schemes: Various PRSs[1], which are useful in nearest-neighbour-like classification, have been reported in the literature - two excellent surveys are found in [20], [21]. Bezdek *et al* [20], who composed the second and more recent survey of the field, reported that there are "zillions!" of methods for finding prototypes (see page 1459 of [20]). One of the first of its kind, was a method that led to a smaller prototype set, the Condensed Nearest Neighbor (CNN) rule [22]. Since the development of the CNN, other methods [23] - [25] have been proposed successively, such as the Prototypes for Nearest Neighbor (PNN) classifiers [23] (including a modified Chang's method proposed by Bezdek), Vector Quantization etc. Apart from the above methods, we mention the following: Support Vector Machines (SVM) [24] can also be used as a means of selecting prototype vectors. Observe that these new vectors can be subsequently adjusted by means of an LVQ3-type algorithm. Based on this idea, a new PRS (referred to here as HYB) of hybridizing the SVM and the LVQ3 was introduced in [25]. Based on the philosophy that points near the separating boundary between the classes play more important roles than those which are more interior in the feature space, and that of *selecting* and *adjusting* the reduced prototypes, a new hybrid approach that involved two distinct phases was proposed in [25]. Due to space limitations, the details of other schemes are omitted, but can be found in [26].

2 Optimizing the KFDA with PRSs

The fundamental problem that we encounter when optimizing any KFDA is that of reducing the dimensionality of the training samples. This, in turn, involves four essential phases, namely that of computing the kernel matrix, computing *its* eigenvalues and eigenvectors, extracting the principal components of the kernel matrix from among these eigenvectors, and finally, projecting the samples to be processed onto the reduced basis vectors. We observe, first of all, that all of these phases depend on the size of the data set. In particular, the most time consuming phase involves computing the eigenvalues and eigenvectors of the kernel matrix.

There are a few ways by which the computational burden of the kernel method can be reduced. Most of the reported schemes [11], [12], [13], [14], [15] resort to using the specific properties of the underlying kernel matrix, for example, its sparseness. Our technique is different. The method we propose is by reducing the *size* of the training set. However, we do this, by not significantly reducing the accuracy of the resultant training samples. This is achieved by using a PRS.

The rationale for the proposed method can be conceptually explained using Fig. 1. Fig. 1(a) represents the original training samples presented to the clas-

[1] Our overview is necessarily brief, but additional details can be found in [26].

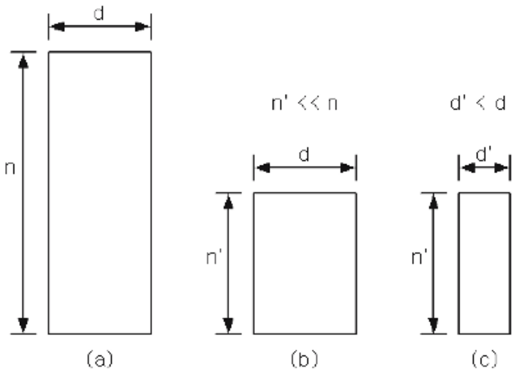

Fig. 1. The rationale for the proposed method. (a) The training samples, where n and d are the number of samples and the dimensionality, respectively. (b) The condensed prototypes extracted from the training samples using a PRS, where $n' \ll n$. (c) The prototype vectors whose dimensionality has been reduced with a KFDA, where $d' \ll d$.

sifier system, where n and d are the number of samples and the dimensionality, respectively. Fig. 1(b) represents the condensed prototypes which are extracted from the training samples using a PRS, where $n' \ll n$. Using the latter data, Fig. 1(c) represents the resultant prototype vectors in which the dimensionality has been reduced by invoking a KFDA, where $d' \ll d$. Observe that since the fundamental problem with *any* kernel-based scheme is that it increases the time complexity from $O(d^3)$ to $O(n^3)$, we can see that the time complexity for the dimensionality reduction from d to d' is sharply decreased from $O(n^3)$ to $O(n'^3)$.

The question now is essentially one of determining which of the training points we should retain. Rather than deciding to discard or retain the training points, we permit the user the choice of either *selecting* some of the training samples using methods such as the CNN, or *creating* a smaller set of samples using the methods such as those advocated in the PNN, VQ, and HYB. This reduced set effectively represents the new "training" set. Additionally, we also permit the user to migrate the resultant set by an LVQ3-type method to further enhance the quality of the reduced samples.

The PRS serves as a preprocessor to the n d-dimensional training samples to yield a subset of n' potentially new points, where $n' << n$. The "kernel" is now computed using this reduced set of points to yield the so-called *reduced-kernel* matrix. The eigenvalues and eigenvectors of *this* matrix are now computed, and the principal components of the kernel matrix are extracted from among *these* eigenvectors of smaller dimension. Notice now that the samples to be tested are projected onto the reduced basis directions represented by *these* vectors.

To investigate the computational advantage gained by resorting to such a PRS preprocessing phase, we observe, first of all, that the time used in determining the reduced prototypes is *fractional* compared to the time required for the expensive matrix-related operations. Once the reduced prototypes are obtained, the eigenvalue/eigenvector computations are significantly smaller since these computations are now done for a much smaller set, and thus for an $n' \times n'$ matrix. The net result of these two reductions is reflected in the time savings we report in a later section in which we discuss the experimental results obtained for artificial and real-life data sets.

3 Experimental Results: Artificial/Real-Life Data Sets

Experimental Data: The proposed method has been rigorously tested and compared with many conventional ones. This was done by performing experiments on both "artificial" and "real-life" data sets.

The data set described as "Random" is generated randomly with a uniform distribution but with irregular decision boundaries. In this case, the points are generated uniformly, and the assignment of the points to the respective classes is achieved by *artificially* assigning them to the region they fall into, as per the manually created "irregular decision boundary". The data set named "Non_normal 2", which has also been employed as a benchmark experimental data set [1], and [25] for numerous experimental set-ups was generated from a mixture of four 8-dimensional Gaussian distributions. The data sets "Iris2", "Ionosphere" (in short, "Iono"), "Sonar", "Arrhythmia" (in short, "Arrhy") and "Adult4", which are real benchmark data sets, are cited from the UCI Machine Learning Repository[2]. Their details can be found in the latter site, and also in [26] and omitted here in the interest of compactness. In the above data sets, the data set for class ω_j was randomly split into two subsets, $T_{j,T}$ and $T_{j,V}$, of equal size. One of them was used for choosing the initial prototypes and training the classifiers, and the other one was used in their validation (or testing). Later, the role of these sets were interchanged.

As in all learning algorithms, choosing the parameters of the PRS and KFDA play an important role in determining the quality of the solution. The parameters for the PRS, the KPCA and the KFDA, are summarized as follows:

1. The kernel function employed is the polynomial $k(x_i, x_j) = (1 + x_i' x_j)^2$.
2. The number of features to be selected is 2 for all the KFDAs.
3. The constant μ is chosen as $\mu = 0.001$ for regularization in KFD, and the fusion coefficient θ in CKFDA is chosen as $\theta = 1.4$.

Selecting Prototype Vectors: In order to evaluate the proposed dimensionality reduction mechanisms, we first selected the prototype vectors from the experimental data sets using the CNN, the PNN and the HYB algorithms. In the HYB, we selected initial prototypes using a SVM algorithm. After this selection, we invoked a phase in which the optimal positions (i.e., with regard to classification) were learned with an LVQ3-type scheme [25]. For the SVM and LVQ3 programs, we utilized two publicly-available software packages[3].

[2] http://www.ics.uci.edu/mlearn/MLRepository.html

[3] These packages can be available from http://www-ai.cs.uni-dortmund.de/SOFTWARE/SVM_LIGHT/svm_light.eng.html and http://cochlea.hut.fi/research/som_lvq_pak.shtml, respectively.

Table 1. The classification accuracies of the proposed computational mechanisms for the artificial and real-life data sets. The details of the entries and how the values were obtained are explained in the text.

Type	Dataset	PRSs	WHL	KPCA	KFD	GDA	KPCA+LDA	CKFDA
Artificial Data	Rand	WHL	96.50	79.50	88.50	88.50	88.50	91.75
		CNN	96.25	59.75	80.50	80.50	80.50	85.25
		PNN	95.75	61.25	83.00	83.00	83.00	89.25
		HYB	89.50	70.50	85.75	85.75	85.75	85.75
	Non_n2	WHL	92.50	92.50	92.50	92.50	92.50	92.50
		CNN	91.90	91.90	91.90	91.90	91.90	91.90
		PNN	92.10	92.10	92.10	92.10	92.10	92.20
		HYB	94.00	94.00	94.00	94.00	94.00	94.10
Real-life Data	Iris2	WHL	92.00	71.00	94.00	94.00	94.00	92.00
		CNN	89.00	63.00	93.00	93.00	93.00	95.00
		PNN	94.00	56.00	89.00	91.00	91.00	89.00
		HYB	94.00	72.00	95.00	92.00	92.00	93.00
	Ionos	WHL	78.65	75.85	76.14	88.64	88.64	83.52
		CNN	81.82	45.17	69.89	88.07	88.07	75.85
		PNN	82.68	43.19	80.11	84.09	84.09	84.94
		HYB	83.24	48.01	80.68	84.94	84.94	83.24
	Sonar	WHL	82.22	52.89	84.14	83.18	83.18	82.21
		CNN	79.81	53.37	77.89	79.81	79.81	77.41
		PNN	82.69	48.56	79.33	81.73	81.73	82.69
		HYB	80.77	50.97	82.21	79.81	79.81	81.73
	Arrhy	WHL	97.57	79.87	99.78	99.78	99.78	99.78
		CNN	96.47	49.78	99.12	99.78	99.99	99.78
		PNN	99.12	53.54	97.57	99.78	99.78	99.33
		HYB	99.12	84.07	99.78	99.33	99.33	99.11
	Adult4	WHL	93.40	91.85	92.97	92.07	92.07	92.73
		CNN	91.58	81.85	83.67	84.40	84.40	85.87
		PNN	89.36	79.35	80.82	81.04	81.04	83.58
		HYB	92.78	59.41	86.41	82.39	82.39	87.32

From the experiments, we can see that the kernel matrix dimensions to be processed in the KFDA computations can be reduced significantly by first employing a PRS. Thus, for the artificial data set "Non_n2" data set, the dimensionality reduced from 500×500 to 63×63 when the HYB method was used as the PRS, and for the real data set "Arrhy", the dimensionality reduced from 226×226 to 8×8 when the PNN method was used as the PRS. Both of these[4] are truly *significant* by any metric of measurement. It should also be mentioned that the reduction rate increased *dramatically* when the size of the data sets was increased. The reduction in the resultant KFDA processing time follows as a natural consequence!

Experimental Results: Table 1 shows the classification accuracies of the proposed computational mechanisms for the data sets. In WHL, the test data sets

[4] The results of the other data sets are omitted here, but can be found in [26].

were classified with the NN rule by utilizing the entire training sets as the code-book vectors. On the other hand, for KPCA, KFD, GDA, KPCA+LDA, and CKFDA classifications, we first chose prototype samples from the training data sets with the CNN, PNN and HYB methods respectively. After selecting the prototype vectors, we reduced their dimensionality using the KPCA, KFD, GDA, KPCA+LDA, and CKFDA methods. Finally, the test data sets were classified with the respective decision rules, where the prototype vectors of reduced dimensionality were utilized as the code-book vectors. The experiments were repeated by exchanging the roles of the data sets, and the two results were then averaged.

Consider the processing times for the "Non_n2" data set. If the entire set of size 500 was processed, the times taken for the KPCA, KFD, GDA, KPCA+LDA and CKFDA are 32.02, 78.44, 138.21, 30.20 and 30.36 seconds, respectively. However, if the same sets were first preprocessed by the CNN, to yield the CNN-KPCA, CNN-KFD, CNN-GDA, CNN-KPCA+LDA and CNN-CKFDA procedures[5], the processing times are 2.54, 3.28, 7.51, 2.53 and 2.55 seconds respectively - which represent a 10-fold to 20-fold reduction ! Notice that the classification accuracies of the method are almost the same as shown in Table 1. Identical comments can also be made about the PNN and HYB schemes[6]. The results of the other data sets are omitted here in the interest of brevity, but is in [26].

4 Conclusions

In this paper, we suggest a computationally superior mechanism to solve the computational problem for KFDA methods. Rather than define the kernel matrix and compute the principal components using the entire data set, we propose that the size of the data be reduced into a smaller prototype subset using a PRS Since the PRS yields a smaller subset of data points that effectively samples the entire space to yield subsets of prototypes, this alleviates the computational burden significantly. The experimental results demonstrate that the proposed schemes can improve the extracting speed of the proposed methods by an order of magnitude, while yielding almost the same classification accuracy.

References

1. K. Fukunaga.: *Introduction to Statistical Pattern Recognition, Second Edition.* Academic Press, San Diego, 1990.
2. F. Camastra.: Data Dimensionality Estimation Methods: A Survey. *Pattern Recognition*, **36** 2945–2954, 2003.
3. R. Lotlikar and R. Kothari.: Adaptive Linear Dimensionality Reduction for Classification. *Pattern Recognition*, **33** 185–194, 2000.
4. J. Yang and J. -Y. Yang.: Why can LDA be performed in PCA transformed Space?. *Pattern Recognition*, **36** 563–566, 2003.
5. T. Cooke.: Two Variations on Fisher's Linear Discriminant for Pattern Recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, **PAMI-24(2)** 268–273, Feb. 2002.

[5] Here, the notation of CNN-KPCA means that the dimensionality reduction is done with KPCA *after* extracting prototypes with the CNN method.

[6] It should be mentioned that such an increase/decrease in accuracy is insignificant.

6. B. Schölkopf, A. J. Smola, and K. -R. Müller.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, **10** 1299–1319, 1998.
7. S. Mika, G. Ratsch, B. Schölkopf, A. Smola, J. Weston, and K. R. Müller.: Fisher Discriminant Analysis with Kernels. *Proc. of IEEE International Workshop Neural Networks for Signal Processing IX*, 41–48, Aug. 1999.
8. G. Baudat and F. Anouar.: Generalized Discriminant Analysis Using a Kernel Approach. *Neural Comput.*, **12** 2385–2404, 2000.
9. J. Yang, A. F. Frangi, J. -Y, Yang and D. Zhang.: KPCA plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, **PAMI-27(2)** 230–244, Feb. 2005.
10. M. Loog and R. P. W. Duin.: Linear dimensionality reduction via a Heteroscedastic extension of LDA: The Chernoff criterion. *IEEE Trans. Pattern Anal. and Machine Intell.*, **PAMI-26(6)** 732–739, June 2004.
11. D. Achlioptas and F. McSherry.: Fast computation of low-rank approximations. *Proc. of the Thirty-Third Annual ACM Symposium on the Theory of Computing*, Hersonissos, Greece, ACM Press, 611–618, 2001.
12. D. Achlioptas, F. McSherry and B. Schölkopf.: Sampling techniques for kernel methods. *Advances in Neural Information Processing Systems*, **14**, MIT Press, Cambridge, MA, 335–342, 2002.
13. A. J. Smola and B. Schölkopf.: Sparse greedy matrix approximation for machine learning. *Proc. of ICML'00*, Bochum, Germany, Morgan Kaufmann, 911–918, 2000.
14. C. Williams and M. Seeger.: Using the Nystrom method to speed up kernel machines. *Advances in Neural Information Processing Systems*, **13**, MIT Press, Cambridge, MA, 2001.
15. M. Tipping.: Sparse kernel principal component analysis. *Advances in Neural Information Processing Systems*, **13**, MIT Press, Cambridge, MA, 633–639, 2001.
16. G. C. Cawley and N. L. C. Talbot.: Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. *Pattern Recognition*, **36** 2585–2592, 2003.
17. Y. Xu, J. -Y. Yang and J. Yang.: A reformative kernel Fisher discriminant analysis. *Pattern Recognition*, **37** 1299–1302, 2004.
18. Y. Xu, J. -Y. Yang, J. Lu and D.-J. Yu.: An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments. *Pattern Recognition*, **37** 2091–2094, 2004.
19. S. A. Billings and K. L. Lee.: Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural Networks*, **15(2)** 263–270, 2002.
20. J. C. Bezdek and L. I. Kuncheva.: Nearest prototype classifier designs: An experimental study. *Int'l. Journal of Intelligent Systems*, **16(12)** 1445–11473, 2001.
21. B. V. Dasarathy.: *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, 1991.
22. P. E. Hart.: The condensed nearest neighbor rule. *IEEE Trans. Inform. Theory*, **IT-14** 515–516, May 1968.
23. C. L. Chang.: Finding prototypes for nearest neighbor classifiers. *IEEE Trans. Computers*, **C-23(11)** 1179–1184, Nov. 1974.
24. C. J. C. Burges.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2(2)** 121–167, 1998.
25. S. -W. Kim and B. J. Oommen.: Enhancing prototype reduction schemes with LVQ3-type algorithms. *Pattern Recognition*, **36(5)** 1083–1093, 2003.
26. S.-W. Kim and B. J. Oommen, "On using prototype reduction schemes to optimize kernel-based Fisher discriminant analysis". *Unabridged version of this paper.*