Adaptive Nonlinear Discriminant Analysis by Regularized Minimum Squared Errors*

Hyunsoo Kim,† Barry L. Drake,‡ and Haesun Park§

February 23, 2005

Abstract: Recently, kernelized nonlinear extensions of Fisher's discriminant analysis, discriminant analysis based on generalized singular value decomposition (LDA/GSVD), and discriminant analysis based on the minimum squared error formulation (MSE) have been introduced for handling undersampled problems and nonlinearly separable data sets. In this paper, an efficient algorithm for adaptive linear and nonlinear kernel discriminant analysis based on regularized MSE, called adaptive KDA/RMSE, is proposed. In adaptive KDE/RMSE, updating and downdating of

*This material is based upon work supported in part by the National Science Foundation Grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 200 Union Street S.E., Minneapolis, MN 55455, USA (hskim@cs.umn.edu).

‡Adaptive Computing Systems Ltd. (bldrake@ieee.org)

§Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 200 Union Street S.E., Minneapolis, MN 55455, USA, and the National Science Foundation, 4201 Wilson Boulevard, Arlington, VA 22230, USA (hpark@cs.umn.edu)

the computationally expensive eigenvalue decomposition (EVD) or singular value decomposition (SVD) is approximated by updating and downdating of the QR decomposition achieving an order of magnitude speed up. This fast algorithm for adaptive kernelized discriminant analysis is designed by utilizing regularization and the relationship between linear and nonlinear discriminant analysis and the MSE. In addition, an efficient algorithm to compute leave-one-out cross validation is also introduced by utilizing downdating of KDA/RMSE.

Keywords: QR decomposition updating and downdating, adaptive classifier, leave-one-out cross validation, linear discriminant analysis, kernel methods, regularization

1 Introduction

In linear discriminant analysis (LDA), a linear transformation for feature extraction is found, which maximizes the between-class scatter and minimizes the within-class scatter [12, 8]. Although LDA is conceptually simple and has been successfully applied in many application areas, it has some limitations: it requires one of the scatter matrices to be nonsingular, and it is applicable only to linearly separable problems. To overcome the nonsingularity restriction, recently, linear discriminant analysis based on the generalized singular value decomposition (LDA/GSVD) [15] has been introduced. Incorporating Mercer's kernel [30], LDA/GSVD has also been extended to kernel discriminant analysis called the KDA/GSVD [21]. Other methods and two-stage approaches for generalization of LDA to undersampled problems have also been studied [6, 33, 32].

There have been several approaches to incremental learning machines, which can effectively compute the updated decision function when data points are appended. For support vector ma-

chines (SVMs) [7, 30, 31], incremental learning has been proposed in order to handle very large data sets efficiently [29], where training a subset of the entire training data set and merging the support vectors found in the iterative training is repeated. This approach is helpful for dealing with very large data sets. Moreover, when new data points are appended, the stored support vectors and the appended data points can be used to obtain updated separating hyperplanes. For many applications where expensive updating of transactions is frequently required, it is desirable to develop adaptive machine learning algorithms, which can effectively compute the updated decision function when data points are appended or deleted.

In numerical linear algebra, updating and downdating of matrix decompositions have been widely studied [13, 14]. Updating and downdating of least squares solutions [3, 9], the QR decomposition [24], adaptive condition estimation [11, 24, 23], and the Cholesky decomposition [25, 4, 10] have been extensively studied. Also, SVD updating and downdating [19] and its effective and efficient approximation by the ULV and URV decompositions [26, 27, 28] have been applied to many important problems in science and engineering.

In this paper, an efficient algorithm for adaptive linear and nonlinear kernel discriminant analysis based on regularized MSE called adaptive KDA/RMSE is proposed. This adaptive classifier avoids the computationally expensive eigenvalue decomposition (EVD) updating, which would be necessary for the design of an adaptive classifier for linear and nonlinear discriminant analysis, i.e., the adaptive KDA/RMSE in its original formulation. In adaptive KDA/RMSE, updating and downdating of the computationally expensive EVD is approximated by updating and downdating of the QR decomposition that is of an order of magnitude faster. An efficient algorithm to compute leave-one-out cross validation is also introduced by utilizing the adaptive KDA/RMSE.

The rest of this paper is organized as follows. A brief overview of a nonlinear discriminant analysis in feature space and based on the minimum squared error formulation [1] is provided in Section 2. In Section 3, kernel discriminant analysis by the minimum squared error formulation (KDA/MSE) is introduced. Using regularization and the relationship between KDA/MSE and KDA/GSVD [21] an efficient adaptive KDA/RMSE is proposed in Section 4. Finally, an efficient leave-one-out cross validation method is introduced by downdating the new QR decomposition-based KDA/RMSE in Section 5.

Throughout this paper it is assumed that a data set $A \in \mathbb{R}^{m \times n}$ with two classes is given,

$$A = \begin{pmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_m^T \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \in \mathbb{R}^{m \times n},$$

where the jth row $\mathbf{a}_j^T$ of the matrix A denotes the jth data item, and the rows of submatrices A_1 and A_2 belong to classes 1 and 2, respectively. In addition, m_i denotes the number of items in class i, $\mathbf{c}_i$ the centroid vector which is the average of all the data in class i, for $1 \leq i \leq 2$, and $\mathbf{c}$ the global centroid vector. Suppose a nonlinear feature mapping ϕ maps the input data to a feature space. The mapped jth data point in the feature space is represented as $\phi(\mathbf{a}_j)^T$, $1 \leq j \leq m$, $\mathbf{c}_i^\phi$ denotes the centroid vector which is the average of all the data in class i in the feature space for $1 \leq i \leq 2$, and $\mathbf{c}^\phi$ the global centroid vector in the feature space. Note that $\mathbf{c}_i^\phi \neq \phi(\mathbf{c}_i)$ and $\mathbf{c}^\phi \neq \phi(\mathbf{c})$, in general, since the mapping ϕ is nonlinear.

2 Linear and Nonlinear Discriminant Analysis by the Minimum Squared Error Formulation

In this Section, kernel discriminant analysis by the minimum squared error formulation in feature space [1] is briefly reviewed. Assume that a feature mapping $\phi(\cdot)$ maps the input data to a higher dimensional feature space:

$$\mathbf{a} \in \mathbb{R}^{n\times 1} \rightarrow \phi(\mathbf{a}) \in \mathbb{R}^{N\times 1}, n < N.$$

Then, for the training data $(\mathbf{a}_i, y_i)$, $1 \leq i \leq m$, in order to obtain the discriminant function,

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} + \beta, \tag{1}$$

for binary classification problems in feature space, a linear system can be built as

$$\begin{bmatrix} 1 & \phi(\mathbf{a}_1)^T \\ \vdots & \vdots \\ 1 & \phi(\mathbf{a}_{m_1})^T \\ 1 & \phi(\mathbf{a}_{m_1+1})^T \\ \vdots & \vdots \\ 1 & \phi(\mathbf{a}_m)^T \end{bmatrix} \begin{bmatrix} \beta \\ \mathbf{w} \end{bmatrix} \cong \begin{bmatrix} y_1 \mathbf{u}_{m_1} \\ y_2 \mathbf{u}_{m_2} \end{bmatrix}, \tag{2}$$

where y_1 and y_2 are the values that indicate the class membership of the $\phi(\mathbf{a}_j), 1 \leq j \leq m$, and $\mathbf{u}_{m_i} \in \mathbb{R}^{m_i \times 1}$ is a column vector with 1's as its elements for $1 \leq i \leq 2$. Various pairs of numbers can be assigned to (y_1, y_2) to discriminate two classes. When N is very large, Eqn. (2) becomes

an underdetermined system and it has either no solution or infinitely many solutions. In general, Eqn. (2) can be formulated as a problem of minimizing the L_2-norm of the error as

$$\min_{\beta,\mathbf{w}} \left\| \begin{bmatrix} \mathbf{u}_{m_1} & \phi(A_1) \\ \mathbf{u}_{m_2} & \phi(A_2) \end{bmatrix} \begin{bmatrix} \beta \\ \mathbf{w} \end{bmatrix} - \begin{bmatrix} y_1\mathbf{u}_{m_1} \\ y_2\mathbf{u}_{m_2} \end{bmatrix} \right\|_2^2 \tag{3}$$

$$= \min_{\beta,\mathbf{w}} \left\| F \begin{bmatrix} \beta \\ \mathbf{w} \end{bmatrix} - \mathbf{y} \right\|_2^2,$$

where

$$\phi(A_1) = \begin{bmatrix} \phi(\mathbf{a}_1)^T \\ \vdots \\ \phi(\mathbf{a}_{m_1})^T \end{bmatrix}, \phi(A_2) = \begin{bmatrix} \phi(\mathbf{a}_{m_1+1})^T \\ \vdots \\ \phi(\mathbf{a}_m)^T \end{bmatrix}, F = \begin{bmatrix} \mathbf{u}_{m_1} & \phi(A_1) \\ \mathbf{u}_{m_2} & \phi(A_2) \end{bmatrix} \text{ and } y = \begin{bmatrix} y_1\mathbf{u}_{m_1} \\ y_2\mathbf{u}_{m_2} \end{bmatrix},$$

Among many possible solutions for Eqn. (3), the minimum norm solution is

$$\begin{bmatrix} \beta \\ \mathbf{w} \end{bmatrix} = F^\dagger \mathbf{y}, \tag{4}$$

where $F^\dagger$ is the pseudoinverse of F.

Now, the relationship between this formulation and kernel discriminant analysis is described. The normal equations for Eqn. (3),

$$F^T F \begin{bmatrix} \beta \\ \mathbf{w} \end{bmatrix} = F^T \mathbf{y},$$

which gives

$$\begin{bmatrix} \mathbf{u}_{m_1}^T & \mathbf{u}_{m_2}^T \\ \phi(A_1)^T & \phi(A_2)^T \end{bmatrix} \begin{bmatrix} \mathbf{u}_{m_1} & \phi(A_1) \\ \mathbf{u}_{m_2} & \phi(A_2) \end{bmatrix} \begin{bmatrix} \beta \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{m_1}^T & \mathbf{u}_{m_2}^T \\ \phi(A_1)^T & \phi(A_2)^T \end{bmatrix} \begin{bmatrix} y_1\mathbf{u}_{m_1} \\ y_2\mathbf{u}_{m_2} \end{bmatrix}. \tag{5}$$

Setting

$$y_1 = m/m_1 \text{ and } y_2 = -m/m_2,$$

we obtain

$$\phi(A_1)^T \mathbf{u}_{m_1} = m_1 \mathbf{c}_1^\phi \text{ and } \phi(A_2)^T \mathbf{u}_{m_2} = m_2 \mathbf{c}_2^\phi.$$

Eqn. (5) gives

$$\beta = -\frac{(m_1 \mathbf{c}_1^\phi + m_2 \mathbf{c}_2^\phi)^T}{m} \mathbf{w} = -(\mathbf{c}^\phi)^T \mathbf{w}$$

And, using the within-class scatter matrix in feature space

$$S_w^\phi = \sum_{i=1,2} \sum_{\mathbf{a} \in \text{class}_i} (\phi(\mathbf{a}) - \mathbf{c}_i^\phi)(\phi(\mathbf{a}) - \mathbf{c}_i^\phi)^T,$$

we have

$$\left[S_w^\phi + \frac{m_1 m_2}{m} (\mathbf{c}_1^\phi - \mathbf{c}_2^\phi)(\mathbf{c}_1^\phi - \mathbf{c}_2^\phi)^T \right] \mathbf{w} = m(\mathbf{c}_1^\phi - \mathbf{c}_2^\phi),$$

which can be simplified to obtain

$$\mathbf{w} = \rho \cdot (S_w^\phi)^{-1} (\mathbf{c}_1^\phi - \mathbf{c}_2^\phi), \tag{6}$$

where ρ is a scalar constant [8, 1]. Eqn. (6) shows that the parameter $\mathbf{w}$ in the discriminant function in Eqn. (1) obtained by the MSE formulation (3) is related to the dimension reducing transformation $\mathbf{w}$ of kernel Fisher discriminant analysis when S_w^ϕ is nonsingular. This shows that kernelized discriminant analysis can be implemented by a minimum squared error formulation for binary class problems. For the extension to multiclass problems see [22].

As shown above, the KDA/MSE solution in Eqn. (3) is related to kernelized Fisher discriminant analysis when the within-class scatter matrix is nonsingular [8]. In general, the feature space obtained by a feature mapping is a very high dimensional space, and the scatter matrcies are likely to be singular. Therefore, kernelization of the classical FDA or LDA, which involves applying the classical FDA or LDA in feature space, may fail without remedies such as regularization. On the other hand, it can be shown that the KDA/MSE solution in Eqn. (3) is mathematicaclly equivalent to Kernelized LDA/GSVD called KDA/GSVD [21, 22], which has a solution regardless of the singularity of the scatter matrices.

The relationship between the LDA/MSE solution and LDA/GSVD for two-class problems as well as multi-class problems is shown in [22]. Also the corresponding relationship between the kernel MSE solution of (3) and KDA/GSVD can be found in [21].

3 Solution for Kernelized Minimum Squared Error Formulation (KDA/MSE)

In this Section, kernelized nonlinear discriminant analysis based on the minimum squared error formulation (KDA/MSE) is introduced. The main idea of the kernel method is that without knowing the nonlinear feature mapping, ϕ, we can work within feature space through kernel functions, as long as the problem formulation depends only on inner products between data points in feature space. This is based on the fact that for any kernel function, $\mathbf{k}$, satisfying Mercer's condition [7],

there exists a mapping ϕ such that

$$< \phi(\mathbf{x}), \phi(\mathbf{y}) >= \mathbf{k}(\mathbf{x}, \mathbf{y}) \tag{7}$$

where $< , >$ is an inner product. For a finite data set $\{\mathbf{a}_1, \cdots, \mathbf{a}_m\}$, a kernel function $\mathbf{k}$ satisfying Mercer's condition can be rephrased as the kernel matrix $K = [\mathbf{k}(\mathbf{a}_i, \mathbf{a}_j)]_{1 \leq i,j \leq m}$ being positive semi-definite [7]. The polynomial kernel

$$\mathbf{k}(\mathbf{x}, \mathbf{y}) = (\gamma_1(\mathbf{x} \cdot \mathbf{y}) + \gamma_2)^d, d > 0 \text{ and } \gamma_1, \gamma_2 \in R \tag{8}$$

and the Gaussian kernel

$$\mathbf{k}(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/\sigma), \sigma \in R \tag{9}$$

are two of the most widely used kernel functions. The feature map ϕ can be either linear or nonlinear depending on the kernel functions used. If the inner product kernel function $\mathbf{k}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$ is used, the feature map is an identity map. Utilizing kernel methods requires neither the feature map nor the feature space be formed explicitly due to the relation (7) once the kernel function $\mathbf{k}$ is known. Since linear discriminant analysis is a special case of kernelized nonlinear discriminant analysis, the discussion in the rest of paper will focus on nonlinear discriminant analysis. The regularization of Fishers' discriminant analysis with a kernel was originally suggested in [18] to overcome ill-posedness. The KDA/GSVD further generalizes kernel Fisher discriminant analysis to multi-class problems, which does not require regularization [21].

Although Eqns. (1) and (2) show how the MSE method can be applied in feature space, it needs to be reformulated in terms of kernel functions when the feature mapping ϕ is unknown. Since $\mathbf{w}$

in Eqn. (1) can be expressed as a linear combination of $\phi(\mathbf{a}_j)$, $j = 1, \cdots, m$ [21], i.e.

$$\mathbf{w} = \Sigma_{j=1}^{m} z_j \phi(\mathbf{a}_j) = \left[\begin{array}{ccc} \phi(\mathbf{a}_1) & \cdots & \phi(\mathbf{a}_m) \end{array} \right] \mathbf{z} \tag{10}$$

where $\mathbf{z} = \left(\begin{array}{ccc} z_1 & \ldots & z_m \end{array} \right)^T$. Applying Eqn. (10) to Eqn. (3), we obtain

$$\begin{aligned} & \min_{\beta, \mathbf{z}} \left\| \left(\begin{array}{cc} \mathbf{u} & \phi(A) \end{array} \right) \left(\begin{array}{c} \beta \\ \phi(A)^T \mathbf{z} \end{array} \right) - \mathbf{y} \right\|_2^2 \\ = & \min_{\beta, \mathbf{z}} \left\| \left(\begin{array}{cc} \mathbf{u} & \phi(A)\phi(A)^T \end{array} \right) \left(\begin{array}{c} \beta \\ \mathbf{z} \end{array} \right) - \mathbf{y} \right\|_2^2 \\ = & \min_{\beta, \mathbf{z}} \left\| \left(\begin{array}{cc} \mathbf{u} & K \end{array} \right) \left(\begin{array}{c} \beta \\ \mathbf{z} \end{array} \right) - \mathbf{y} \right\|_2^2 \end{aligned} \tag{11}$$

where

$$\mathbf{u} = \left[\begin{array}{c} \mathbf{u}_{m_1} \\ \mathbf{u}_{m_2} \end{array} \right], \phi(A) = \left[\begin{array}{c} \phi(A_1) \\ \phi(A_2) \end{array} \right], \mathbf{y} = \left[\begin{array}{c} y_1 \mathbf{u}_{m_1} \\ y_2 \mathbf{u}_{m_2} \end{array} \right], \; K_{i,j} = \mathbf{k}(\mathbf{a}_i, \mathbf{a}_j).$$

Let $G = \left[\begin{array}{cc} \mathbf{u} & K \end{array} \right] \in \mathbb{R}^{m \times (m+1)}$. Then, a data item $\mathbf{x}$ is assigned to the positive class if

$$\left[\begin{array}{ccc} 1 & \mathbf{k}(\mathbf{x}, \mathbf{a}_1) & \cdots \quad \mathbf{k}(\mathbf{x}, \mathbf{a}_n) \end{array} \right] \left[\begin{array}{c} \beta \\ \mathbf{z} \end{array} \right] > 0 \tag{12}$$

Therefore, the decision rule for binary classification is given by $sign(f(\mathbf{x}))$ from KDA/MSE, where

$$f(\mathbf{x}) = \mathbf{k}(\mathbf{x}, A^T) \left[\begin{array}{c} z_1 \\ \vdots \\ z_m \end{array} \right] + \beta \tag{13}$$

and $\mathbf{k}(\mathbf{x}, A^T)$ is an $1 \times m$ row vector, i.e.

$$\mathbf{k}(\mathbf{x}, A^T) = [\mathbf{k}(\mathbf{x}, \mathbf{a}_1), \ldots, \mathbf{k}(\mathbf{x}, \mathbf{a}_m)].$$

The minimum norm solution among all possible solutions that satisfy Eqn. (11) is

$$\begin{pmatrix} \beta \\ \\ \mathbf{z} \end{pmatrix} = G^\dagger \mathbf{y}, \tag{14}$$

where $G^\dagger$ is the pseudoinverse of G. The pseudoinverse $G^\dagger$ can be obtained as

$$G^\dagger = [V_1 \; V_2] \begin{bmatrix} \Sigma_1^{-1} & 0 \\ \\ 0 & 0 \end{bmatrix} [U_1 \; U_2]^T = V_1 \Sigma_1^{-1} U_1^T$$

based on the singular value decomposition (SVD) of G,

$$G = [U_1 \; U_2] \begin{bmatrix} \Sigma_1 & 0 \\ \\ 0 & 0 \end{bmatrix} [V_1 \; V_2]^T = U_1 \Sigma_1 V_1^T,$$

where $U_1 \in \mathbb{R}^{m \times r}$ with $U_1^T U_1 = I_r$, $\Sigma_1 \in \mathbb{R}^{r \times r}$ nonsingular and diagonal with positive diagonal elements in nonincreasing order, $V_1 \in \mathbb{R}^{(m+1) \times r}$ with $V_1^T V_1 = I_r$ and $r = \mathrm{rank}(G)$.

When $\mathrm{rank}(G) = m$, its pseudoinverse can be computed much more efficiently using the QR decomposition of G^T. Let the QR decomposition of G^T be given as

$$G^T = Q \begin{pmatrix} R \\ \\ \mathbf{0}_{1 \times m} \end{pmatrix},$$

where $Q \in \mathbb{R}^{(m+1) \times (m+1)}$ with $Q^T Q = I_{m+1}$ and the upper triangular matrix $R \in \mathbb{R}^{m \times m}$ is nonsingular. Then,

$$G^\dagger = G^T (G G^T)^{-1} = Q \begin{pmatrix} R^{-T} \\ \\ \mathbf{0} \end{pmatrix}. \tag{15}$$

Algorithm 1 $[Q, R]$=QRinsert_col$(Q, R, j, \mathbf{x})$

Given the factors Q and R from the QR decomposition of a matrix A, this algorithm computes the updated Q and R factors after inserting a column vector $\mathbf{x}$ before the jth column of A.

1. Insert $\mathbf{x}$ before the jth column of R.
2. R has nonzeros below the diagonal in the jth column. Determine the Givens rotations to annihilate these nonzeros from bottom to top and multiply them to R from the left.
3. Update Q by multiply the transpose of the Givens rotations from the right.

In many applications, positive definite kernels such as the Gaussian radial basis function have been successfully utilized. Assuming no duplicated data points, positive definite kernels produce a positive definite kernel matrix K. Accordingly, $\mathrm{rank}(G) = rank(\begin{pmatrix} \mathbf{u} & K \end{pmatrix}) = rank(K) = m$ in these cases and $G^{\dagger}$ can be computed via the QR decomposition. This presents an important advantage in designing an adaptive KDA/MSE and an adaptive KDA/GSVD method for binary class problems. When a new data item is added, or an existing data item is removed, the new dimension reducing transformation from KDA/GSVD can be computed by updating the GSVD for the old data. If we utilize the equivalence relationship between MSE and LDA/GSVD [22] and their nonlinear extensions [21] via positive definite kernel functions, then updating the GSVD can be replaced by updating the QR decomposition, which is an order of magnitude faster.

Algorithm 2 $[Q, R]$=QRinsert_row($Q, R, j, \mathbf{x}$)

Given the factors Q and R from the QR decomposition of a matrix A, this algorithm computes the updated Q and R factors after inserting a row vector $\mathbf{x}$ before the jth row of A.

1. $R = [\mathbf{x};\ R]$; $Q = [1\ 0;\ 0\ Q]$;
2. Now, R is upper Hessenberg. Determine the Givens rotations to annihilage the nonzeros below the diagonal of R from top to bottom and multiply them to R from the left.
3. Update Q by multiplying the transpose of the Givens rotations from the right. A row permutation is applied to Q to shuffle the first row to the jth row.

4 Efficient Adaptive KDA by Regularized MSE (KDA/RMSE)

In this Section, an adaptive KDA based on regularized MSE, KDA/RMSE, is proposed, which can efficiently compute the updated solution when data points are appended or removed. In general, a kernel matrix is symmetric positive semidefinite and the kernel matrix for a positive definite kernel becomes positive semidefinite when there are duplicated data points in the training data set.

First, we introduce a KDA based on regularized MSE (KDA/RMSE), which overcomes the potential rank deficiency problem by regularization. For the decision rule for binary classification, the formulation of KDA/RMSE is

$$\min_{\beta,\mathbf{z}} \left\| \begin{bmatrix} \mathbf{u} & K + \lambda I \end{bmatrix} \begin{bmatrix} \beta \\ \mathbf{z} \end{bmatrix} - \mathbf{y} \right\|_2^2, \tag{16}$$

where λ is a regularization parameter, $\lambda > 0$.

The minimum 2-norm solution for Eqn. (16) can be found by computing the QR factorization

Algorithm 3 $[Q, R]$=QRremove_col(Q, R, j)

Given the factors Q and R from the QR decomposition of a matrix A, this algorithm computes the updated Q and R factors after deleting the jth column of A.

1. Remove the jth column of R.
2. R has nonzeros below the diagonal from the jth column to the last column. Determine the Givens rotations to annihilage these nonzeros from left to right and multiply them to R from the left.
3. Update Q by multiplying the transpose of these Givens rotations from the right.

of the matrix G_λ^T where

$$G_\lambda = \begin{bmatrix} \mathbf{u} & K + \lambda I \end{bmatrix} \in \mathbb{R}^{m \times (m+1)}. \tag{17}$$

Let the QR decomposition of G_λ^T be

$$G_\lambda^T = Q \begin{bmatrix} R \\ \\ \mathbf{0}_{m\times 1} \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ \\ \mathbf{0} \end{bmatrix} = Q_1 R,$$

where $Q \in \mathbb{R}^{(m+1)\times(m+1)}$ is an orthogonal matrix, $Q_1 \in \mathbb{R}^{(m+1)\times m}$, $Q_2 \in \mathbb{R}^{(m+1)\times 1}$, and $R \in \mathbb{R}^{m\times m}$ is a nonsingular upper triangular matrix. Then, the solution

$$\begin{bmatrix} \beta \\ \\ \mathbf{z} \end{bmatrix} = G_\lambda^\dagger \mathbf{y}, \quad \text{where} \quad \mathbf{y} = \begin{bmatrix} y_1 \mathbf{u}_{m_1} \\ \\ y_2 \mathbf{u}_{m_2} \end{bmatrix}, \tag{18}$$

of Eqn. (16) can be obtained by applying Eqn. (15) to G_λ, i.e. by solving

$$\mathbf{y} = R^T \mathbf{r}, \tag{19}$$

Algorithm 4 $[Q, R]$=QRremove_row(Q, R, j)

Given the factors Q and R from the QR decomposition of a matrix A, this algorithm computes the updated Q and R factors after deleting the jth row of A.

1. Let the transpose of the$j$th row of Q be the column vector $\mathbf{q}$. Determine a product of Givens rotations to make $\mathbf{q}$ into $[1\ 0\ \ldots\ 0]^T$ from bottom to top. Multiply transposed of these Givens rotations to Q from the right.

2. Apply these Givens rotations to R from the left, which becomes upper Hessenberg.

3. Remove the first column and the jth row of Q. Remove the first row of R.

for $\mathbf{r}$ and computing

$$\begin{bmatrix} \beta \\ \mathbf{z} \end{bmatrix} = Q \begin{bmatrix} \mathbf{r} \\ 0 \end{bmatrix}. \tag{20}$$

The above shows that the KDA solution of Eqn. (11) can be achieved by a QR decomposition, solving a linear system, and a matrix vector multiplication. Moreover, knowing the updated Q^* and R^* when data points are appended or removed, we can efficiently obtain the updated solution β^* and $\mathbf{z}^*$. It is much more efficient to update the QR decomposition of G_λ^T than to compute it from scratch or by updating an SVD [2, 14].

An efficient adaptive kernel discriminant analysis algorithm can be designed by using KDA/RMSE and the QR decomposition updated by Given's rotations. Suppose that we have the QR decomposition for the matrix G_λ, and now we wish to obtain the updated solutions β and $\mathbf{z}$ required after a change in the data set. When a new data point that belongs to class i is added, the matrix G_λ and $\mathbf{y}$ should be modified. Because of the special structure of the kernel matrix $K(A, A^T)$, a new row

as well as a new column needs to be attached to G_λ and a value, i.e. y_1 or y_2 for the corresponding class, to $\mathbf{y}$ needs to be inserted.

For example, when a new data point is added to the positive class, the MSE problem is changed to

$$\min_{\beta', \mathbf{z}'} \left\| G'_\lambda \begin{bmatrix} \beta' \\ \mathbf{z}' \end{bmatrix} - \begin{bmatrix} y_1 \mathbf{u}_{m_1+1} \\ y_2 \mathbf{u}_{m_2} \end{bmatrix} \right\|_2^2,$$

where $\mathbf{z}' \in \mathbb{R}^{(m+1)\times 1}$ and

$$G'_\lambda = \left[\begin{array}{c|c|ccc} 1 & K_{a',a'} + \lambda & K_{a',1} & \cdots & K_{a',m} \\ \hline 1 & K_{1,a'} & K_{1,1} + \lambda & \cdots & K_{1,m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & K_{m_1,a'} & K_{m_1,1} & \cdots & K_{m_1,m} \\ 1 & K_{m_1+1,a'} & K_{m_1+1,1} & \cdots & K_{m_1+1,m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & K_{m,a'} & K+m,1 & \cdots & K_{m,m} + \lambda \end{array}\right],$$

where

$$K_{i,j} = \mathbf{k}(\mathbf{a}_i, \mathbf{a}_j), \quad K_{i,a'} = \mathbf{k}(\mathbf{a}_i, \mathbf{a}'), \quad \text{and} \quad K_{a',j} = \mathbf{k}(\mathbf{a}', \mathbf{a}_j).$$

The value of m_1 is increased by 1 since the new data point belongs to the positive class.

When the kth data point $\mathbf{a}_j$ which belongs to the positive class, is deleted, the problem is to

find the new parameters b' and $\mathbf{z}' \in \mathbb{R}^{(m-1)\times 1}$ for which

$$\min_{\beta',\mathbf{z}'} \left\| G'_\lambda \begin{bmatrix} \beta' \\ \mathbf{z}' \end{bmatrix} - \begin{bmatrix} y_1 \mathbf{u}_{m_1-1} \\ y_2 \mathbf{u}_{m_2} \end{bmatrix} \right\|_2^2,$$

where

$$G'_\lambda = \left[\begin{array}{ccc|cc|ccc} 1 & K_{1,1}+\lambda & \cdots & K_{1,k-1} & K_{1,k+1} & \cdots & K_{1,m} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \hline 1 & K_{k-1,1} & \cdots & K_{k-1,k-1}+\lambda & K_{k-1,k+1} & \cdots & K_{k-1,m} \\ 1 & K_{k+1,1} & \cdots & K_{k+1,k-1} & K_{k+1,k+1}+\lambda & \cdots & K_{k+1,m} \\ \hline \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & K_{m_1,1} & \cdots & K_{m_1,k-1} & K_{m_1,k+1} & \cdots & K_{m_1,m} \\ 1 & K_{m_1+1,1} & \cdots & K_{m_1+1,k-1} & K_{m_1+1,k+1} & \cdots & K_{m_1+1,m} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & K_{m,1} & \cdots & K_{m,k-1} & K_{m,k+1} & \cdots & K_{m,m}+\lambda \end{array}\right].$$

As was shown in the examples, a row and a column are appended to G_λ when a data point is added, and a row and a column are removed from G_λ when a data point is deleted. In adaptive KDA/RMSE, we need to update the QR decomposition of G_λ^T

$$G_\lambda^T = \begin{bmatrix} 1 & \cdots & 1 \\ K_{1,1}+\lambda & \cdots & K_{m,1} \\ \vdots & & \vdots \\ K_{1,m} & \cdots & K_{m,m}+\lambda \end{bmatrix} = Q \begin{bmatrix} R \\ 0 \end{bmatrix}. \tag{21}$$

The updating and downdating procedures are summarized for the adaptive KDA/RMSE in Algorithm 5 and 6, respectively. For details of the QR decomposition updating and downdating, which

is summarized in Algorithms 1-4, see [3, 14]. When a data point is appended or deleted, the proposed adaptive KDA/RMSE can compute the updated solution with computational complexity of $O(m^2)$. This is an order of magnitude faster than adaptive KDA/GSVD, which can be derived based on updating/downdating of the SVDs. When several data points are appended or deleted at the same time, a block updating and downdating QR decomposition [3] method can be applied.

5 Efficient Leave-One-Out Cross Validation by Decremental KDA/RMSE

Kernel discriminant analysis has shown excellent classification performance in many applications. The most commonly used model selection methods are k-fold cross validation and leave-one-out cross validation (LOOCV). LOOCV is used during the training of a classifier to prevent overfitting of a classifier on the training set. The procedure of LOOCV is as follows: given a training set of m data points, the first data point in the training set, $\mathbf{a}_1$, is left out. Then the classifier is trained on the remaining $(m-1)$ data points and tested on $\mathbf{a}_1$ producing a score, s_1, which is either 0 (incorrect) or 1 (correct). Then the first data point $\mathbf{a}_1$ is inserted back into the data set and the next data point, $\mathbf{a}_2$, is left out. A new classifier is trained on the remaining $(m-1)$ data points and tested on $\mathbf{a}_2$, producing a score, s_2. This process is repeated until every data point in the data set has had the opportunity to be left out. The LOOCV rate is defined as the average score of all of the individual classifiers:

$$\text{LOOCV rate} = \sum_{i}^{m} s_i/m.$$

The LOOCV performance is a realistic indicator of performance of a classifier on unseen data and is a widely used statistical technique. The LOOCV is rarely adopted in large-scale applications since it is computationally expensive, though it has been widely studied due to its simplicity. In Algorithm 7, we introduce an efficient way to compute the LOOCV rate by downdating of the QR decomposition in KDA/RMSE. The parameters β and $\mathbf{z}$ are first computed with the entire training data set of m items. Then for testing the effect of leaving out each data point, downdating of the QR decomposition of G_λ in KDA/RMSE is performed to obtain the new parameters. The total LOOCV rate can efficiently be computed by applying the QR decomposition downdating m times to obtain each KDA/RMSE solution.

Algorithm (7) efficiently computes the LOOCV rate by downdating the KDA/RMSE solution introduced in the previous Section. An optimal regularization parameter and kernel parameters can be determined by this efficient LOOCV algorithm.

6 Experimental Results

In this section, we present some test results. The purpose of the first set of experiments presented in Section 6.1 is to illustrate the effect of the values of the regularization parameter λ in adaptive KDA/RMSE. In Section 6.2, the tests show that when there is a change in the data, the decision boundary and prediction accuracy obtained by applying KDA/RMSE from scratch and by applying the adaptive KDA/RMSE are the same. Finally in Section 6.3, we show that the proposed LOOCV algorithm based on downdating of KDA/RMSE is significantly faster than the algorithm that recomputes the solution each time data is removed. The test results were obtained using a Sun

Fire V440 with four 1.1GHz UltraSPARC-IIIi CPUs and 8 GB of RAM, and the algorithms were implemented in MATLAB [17].

6.1 Effect of Regularization in KDA/RMSE

In order to investigate the effect of λ values on KDA/RMSE, we observed the difference of the averages of test set classification errors (Δ) for all $p = 100$ partitions,

$$\Delta = \alpha_{SVD} - \alpha_{\lambda} = \frac{1}{p}\sum_{i}^{p} Err_{SVD}(i) - \frac{1}{p}\sum_{i}^{p} Err_{\lambda}(i),$$

where $Err_{SVD}(i)$ and $Err_{\lambda}(i)$ are the test set classification errors of the ith partition obtained by using a pseudo-inverse solution with the SVD applied within KDA/MSE, and the regularized solution, with the regularization parameter λ, using the QR decomposition, respectively. α_{SVD} is the average value of $Err_{SVD}(i)$, and α_{λ} is the average value of $Err_{\lambda}(i)$ for $1 \leq i \leq 100$. In the first training procedure, all training data points were used for training KDA/MSE with the SVD. In this procedure, the regularization parameter λ was not required since the pseudo-inverse of G can be directly computed via the SVD. In the second procedure, 75% of the training data points were used for training KDA/RMSE and the remaining 25% of the data points were inserted one by one by using adaptive KDA/RMSE. Various λ values were used for this second procedure. Figure 1 shows the difference of averages of test classification errors for the Heart data set [18]. When $10^{-11} \leq \lambda \leq 10^{-9}$, the difference was close to zero. When $\lambda \leq 10^{-12}$, the solutions that came from the second procedure were quite different from those that came from the first procedure. When $\lambda \geq 10^{-11}$, the regularization parameter λ tends to effect the solutions. On the other hand, λ values that are too large provide less accurate solutions since they cause larger perturbations in those solutions. A positive value of Δ indicates that the prediction accuracy of KDA/RMSE is

higher than that of KDA/MSE, which uses the SVD. We observed the maximum value of $\Delta = 0.25$ when $\lambda = 10^{-19}$. By optimizing λ as well as γ in KDA/RMSE, we could build more accurate prediction models than those obtained from KDA/MSE with the SVD. This is the expected result since the regularization parameter λ contributes not only to resolve singularity problems, but also handles noisy data sets. [16].

6.2 Results from KDA/RMSE and Adaptive KDA/RMSE

Using a small artificial classification problem, we now show that the same decision boundaries are computed by the proposed adaptive KDA/RMSE algorithm and KDA/RMSE, which must be reapplied each time a data point is appended or removed. The nonlinearly separable data set consists of 12 two-dimensional data points

$$A = \begin{pmatrix} 2 & 3 & 2 & 8 & 6 & 4 & 9 & 9 & 9 & 6 & 7 & 4 \\ 7 & 6 & 2 & 1 & 4 & 8 & 5 & 9 & 4 & 9 & 4 & 4 \end{pmatrix}^T \in \mathbb{R}^{12\times 2}$$

and class index

$$\mathbf{y}_s = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^T \in \mathbb{R}^{12\times 1}.$$

Figure 2 shows the updated decision boundaries when a data point is removed (left panel) or appended (right panel). For the left panel, the 12th data point is removed. For the right panel, a data point, i.e. $\mathbf{a}' = [9\ 8]$, that belongs to the positive class is inserted. The dash-dotted contour presents a decision boundary of adaptive KDA/RMSE and the dashed contour presents that of the KDA/RMSE where the solution vector is computed from scratch with the entire new set of data points. The contours perfectly match in spite of the different numerical pathways to solve the

Table 1: Comparison of the averages and standard deviations of test set classification errors in %. All experiments were performed by kernel discriminant analysis based on the regularized minimum squared error formulation (KDA/RMSE). For testing adaptive KDA/RMSE, after training 75% of the data points, the remaining 25% of the data points were inserted one by one.

Method	Thyroid	Diabetes	Heart	Titanic
KDA	3.9 ± 2.0	26.3 ± 2.2	16.1 ± 3.5	24.1 ± 2.7
KDA (75%) + adaptive KDA (25%)	3.9 ± 2.0	26.3 ± 2.2	16.1 ± 3.5	24.1 ± 2.7

problems. The radial basis function (RBF) kernel $\mathbf{k}(\mathbf{a}_i^T, \mathbf{a}_j) = \exp(-\gamma \|\mathbf{a}_i - \mathbf{a}_j\|^2)$ with $\gamma = 0.1$ was used.

For the next experiments, four data sets used in Mika *et al.* [18] and Billings and Lee [1] were chosen. The RBF kernel was also used. In Table 1, the averages and standard deviations of test set classification errors for all 100 partitions are presented. The Gaussian RBF kernel parameter γ was chosen based on five-fold cross validation by KDA/RMSE for various regularization parameters λ to obtain the optimal solution for each data set. For instance, the parameters of $\gamma = 2^{-36}$ and $\lambda = 10^{-7}$ were chosen for the Heart data set. For testing adaptive classifiers, after training with 75% of the data points, the remaining 25% of the data points are used to obtain the final optimal classification function by using an incremental strategy where the remaining data points are inserted one by one. The experimental results illustrate that the proposed adaptive KDA/RMSE efficiently produces the same solutions as those obtained by KDA/RMSE, which recomputes the solution from scratch each time data is appended or removed.

6.3 LOOCV Timing for Adaptive KDA/RMSE

The sixth data set consists of drug design data that was used in the 2001 KDD cup data mining competition. The data set can be obtained from http://www.cs.wisc.edu/~dpage/kddcup2001. It consists of 1909 compounds tested for their ability to bind to a target site on thrombin, a key receptor in blood clotting. Of these compounds, 42 are active (bind well) and the others are inactive. Each compound is described by a single feature vector comprised of a class value (A for active, I for inactive) and 139,351 binary features, which describe the three-dimensional properties of the molecule. The purpose of this experiment is to show the speedup of Algorithm 7 for fast computation of LOOCV using adaptive KDA/RMSE; the first 8000 binary features and certain numbers of data points were chosen. Figure 3 shows that the LOOCV based on adaptive KDA/RMSE is very efficient. The test results were obtained using a Sun Fire V440 with four 1.1GHz UltraSPARC-IIIi CPUs and 8 GB of RAM. The algorithms were implemented in MATLAB [17].

7 Conclusion and Discussion

In order to design an adaptive KDA/GSVD, expensive updating and downdating of the GSVD needs to be considered. Since the GSVD can be computed by two SVDs [20, 15], the updating of the SVD [5] can be applied for designing an adaptive KDA/GSVD. Unfortunately, SVD updating schemes require $O(m^3)$ operations, which is the same order of magnitude of computational complexity as recomputing the SVD from scratch, although there are still gains from updating [2]. The generalized URV or generalized ULV decompositions [26, 27, 28] give an approximate generalized GSVD. Though one can use the updating of the generalized URV decomposition, the

computational complexity is still higher than updating the QR decomposition. We designed an efficient adaptive KDA/GSVD utilizing the relationship between KDA/MSE and KDA/GSVD, and the updating and downdating of the QR decomposition, which requires computational complexity of $O(m^2)$. This method is an order of magnitude faster than updating and downdating of the SVD, which requires computational complexity of $O(m^3)$. We also proposed an efficient algorithm to compute the LOOCV rate by adaptive KDA/RMSE. Adaptive KDA/RMSE can be utilized in many applications where data are acquired in an "online" fashion and require fast updating.

Acknowledgments

The authors would like to thank the University of Minnesota Supercomputing Institute (MSI) for providing the computing facilities. The work of Haesun Park has been performed while at the National Science Foundation (NSF) and was partly supported by IR/D from NSF.

References

[1] 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 (2002), pp. 263–270.

[2] A. BJÖRCK, *Numerical Methods for Least Square Problems*, SIAM, Philadelphia, PA, 1996.

[3] A. BJÖRCK, H. PARK, AND L. ELDÉN, *Accurate downdating of least squares solutions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 549–568.

[4] A. W. BOJANCZYK, R. P. BRENT, P. VAN DOOREN, AND F. R. DE HOOG, *A note on downdating the Cholesky factorization*, SIAM J. Sci. and Stat. Comp., 8 (1987), pp. 210–221.

[5] P. BUSINGER, *Updating a singular value decomposition*, BIT, 10 (1970), pp. 376–385.

[6] L. CHEN, H. M. LIAO, M. KO, J. LIN, AND G. YU, *A new LDA-based face recognition system which can solve the small sample size problem*, Pattern Recognition, 33 (2000), pp. 1713–1726.

[7] N. CRISTIANINI AND J. SHAWE-TAYLOR, *Support Vector Machines and other kernel-based learning methods*, University Press, Cambridge, 2000.

[8] R. O. DUDA, P. E. HART, AND D. G. STORK, *Pattern Classification*, Wiley-interscience, New York, 2001.

[9] L. ELDÉN AND H. PARK, *Block downdating of least squares solutions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1018–1034.

[10] ——, *Perturbation analysis for block downdating of a Cholesky decomposition*, Numer. Math., 68 (1994), pp. 457–468.

[11] W. R. FERNG, G. H. GOLUB, AND R. J. PLEMMONS, *Adaptive lanczos methods for recursive condition estimation*, Numerical Algorithms, 1 (1991), pp. 1–20.

[12] K. FUKUNAGA, *Introduction to Statistical Pattern Recognition, second edition*, Academic Press, Boston, 1990.

[13] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[14] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations, third edition*, Johns Hopkins University Press, Baltimore, 1996.

[15] P. HOWLAND, M. JEON, AND H. PARK, *Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 165–179.

[16] H. KIM, *Machine Learning and Bioinformatics*, Ph.D. Thesis, University of Minnesota, Twin Cities, MN, USA, 2004.

[17] MATLAB, *User's Guide*, The MathWorks, Inc., Natick, MA 01760, 1992.

[18] S. MIKA, G. RÄTSCH, J. WESTON, B. SCHÖLKOPF, AND K. R. MÜLLER, *Fisher discriminant analysis with kernels*, in Neural Networks for Signal Processing IX, Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, eds., IEEE, 1999, pp. 41–48.

[19] M. MOONEN, P. VAN DOOREN, AND J. VANDEWALLE, *A singular value decomposition updating algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1015–1038.

[20] C. C. PAIGE AND M. A. SAUNDERS, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[21] C. H. PARK AND H. PARK, *Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition*. SIAM J. Matrix Anal. Appl., to appear.

[22] ——, *An efficient algorithm for LDA utilizing the relationship between LDA and the generalized minimum squared error solution*, Tech. Rep. 04-013, Department of Computer Science and Engineering, University of Minnesota, 2004.

[23] D. J. PIERCE AND R. J. PLEMMONS, *Fast adaptive condition estimation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 274–291.

[24] G. SHROFF AND C. H. BISCHOF, *Adaptive condition estimation for rank-one updates of QR factorizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1264–1278.

[25] G. W. STEWART, *The effects of rounding error on an algorithm for downdating a Cholesky factorization*, J. Inst. Math. Applic., 23 (1979), pp. 203–213.

[26] ——, *Updating a rank-revealing ULV decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 494–499.

[27] ——, *Updating URV decompositions in parallel*, Parallel Computing, 20 (1994), pp. 151–172.

[28] M. STEWART AND P. VAN DOOREN, *Updating a generalized URV decomposition*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 479–500.

[29] N. A. SYED, H. LIU, AND K. K. SUNG, *Incremental learning with support vector machines*, in Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Articial Intelligence (IJCAI-99), Stockholm, Sweden, 1999.

[30] V. VAPNIK, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[31] ——, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[32] J. YANG AND J. Y. YANG, *Why can LDA be performed in PCA transformed space?*, Pattern Recognition, 36 (2003), pp. 563–566.

[33] H. YU AND J. YANG, *A direct LDA algorithm for high-dimensional data with application to face recognition*, Pattern Recognition, 34 (2001), pp. 2067–2070.

Algorithm 5 Incremental KDA/RMSE: Updating a data point

Given a data matrix $A = [A_1; A_2]$ for the training data $(\mathbf{a}_i, y_i)$ with $y_i \in \{-1, +1\}$ for $1 \leq i \leq m$ for which the parameters β and $\mathbf{z}$ for the binary nonlinear decision rule in Eqn. (12) are known, this algorithm computes the new parameters for the classifier when a new data point $\mathbf{a}_{new}$ is added to the class Ω_x. The new parameters are computed by updating KDA/RMSE solution and using the Q and R factors obtained from the QR decomposition of G_λ^T in Eqn. (17) for A. The kernel function and regularization parameter λ are assumed to be given.

1. $\mathbf{k}_r = \mathbf{k}(\mathbf{a}_{new}, A^T) = [\mathbf{k}(\mathbf{a}_{new}, \mathbf{a}_1), \ldots, \mathbf{k}(\mathbf{a}_{new}, \mathbf{a}_m)] \in \mathbb{R}^{1 \times m}$;
2. Determine the first row index j of A for the class Ω_x.
3. $[Q, R]$ = QRinsert_col($Q, R, j, [1\ \mathbf{k}_r]^T$); % updating the QR decomposition after inserting a vector before the jth column.
4. Insert $\mathbf{x}$ to A before the jth row.
5. Insert the corresponding output $y \in \{-1, +1\}$ for the class Ω_x to $\mathbf{y}$ before the jth row.
6. $\mathbf{k}_r(j) = \mathbf{k}_r(j) + \lambda$;
7. $[Q, R]$ = QRinsert_row($Q, R, j + 1, \mathbf{k}_r$); % updating the QR decomposition after inserting a vector before the $(j + 1)$th row.
8. Solve Eqn. (19) and compute the solution by (20).

Algorithm 6 Decremental KDA/RMSE: Removing a data point

Given a data matrix $A = [A_1; A_2]$ for the training data $(\mathbf{a}_i, y_i)$ with $y_i \in \{-1, +1\}$ for $1 \leq i \leq m$ for which the parameters β and $\mathbf{z}$ for the binary nonlinear decision rule in Eqn. (12) are known, this algorithm computes the new parameters for the classifier when the rth data point is deleted. The new parameters are computed by downdating KDA/RMSE solution and using the Q and R factors obtained from the QR decomposition of G_λ^T in Eqn. (17) for A. The kernel function and regularization parameter λ are assumed to be given.

1. Remove the rth row of $\mathbf{y}$.

2. $[Q, R]$ = QRremove_col(Q, R, r); % downdating the QR decomposition after removing the rth column.

3. $[Q, R]$ = QRremove_row($Q, R, r + 1$); % downdating the QR decomposition after removing the $(r + 1)$th row.

4. Solve Eqn. (19) and compute the solution by (20).

Algorithm 7 Efficient Computation of LOOCV by decremental KDA/RMSE

Given a data matrix $A \in \mathbb{R}^{m \times n}$, a kernel functio $\mathbf{k}$, and a regularization parameter λ, this algorithm computes the LOOCV rate for KDA/RMSE.

1. $s = 0$;
2. Compute G_λ of Eqn. (17) and $\mathbf{y}$ of Eqn. 18.
3. Compute the QR decomposition of G_λ and store the matrices Q and R.
4. Compute the solution vector $[\beta;\ \mathbf{z}]$ from Eqns. (19) and (20).
5. For i = 1 : m
   (a) Let $\mathbf{x}$ be the ith data item.
   (b) Compute the downdated solution vector $[\beta^*;\ \mathbf{z}^*]$ by Algorithm 6 using input arguments (Q, R, i).
   (c) Classify the ith data item by $f(\mathbf{x}) = \mathbf{k}(\mathbf{x}, A^T)\mathbf{z}^* + \beta^*$.
   (d) If the classification result is correct, then $s = s + 1$;
6. $loocv\_acc = (s/m) * 100.0$;

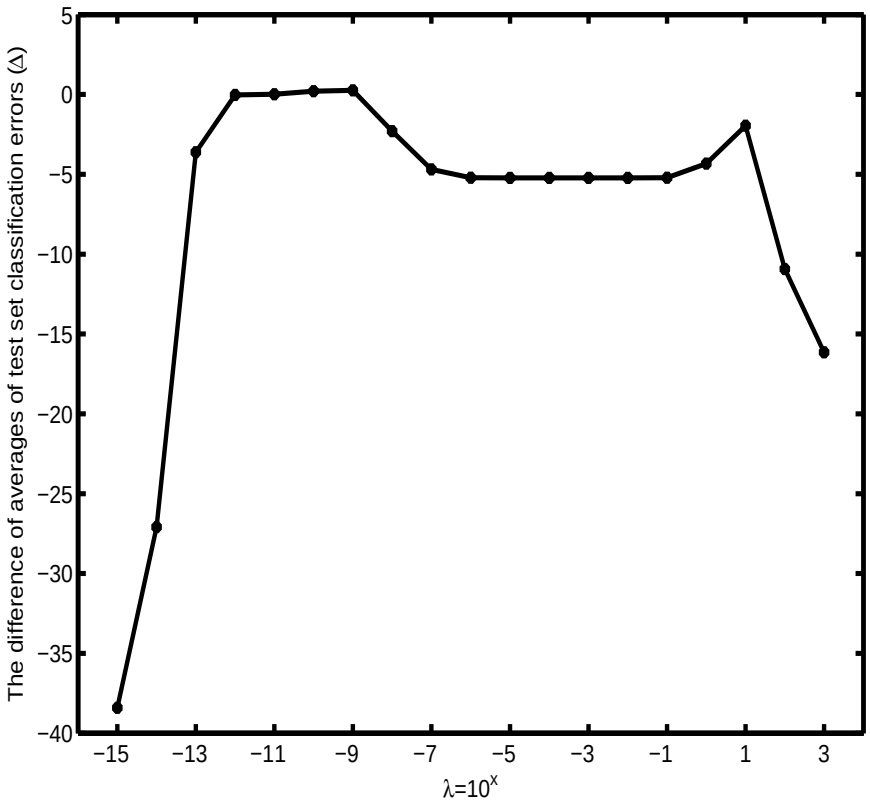

Figure 1: The difference of averages of test set classification errors (Δ) in % for all 100 partitions, which came from two different training procedures for the Heart data set. In the first procedure, all training data points were used for training KDA/MSE with the SVD. In the second procedure, 75% of the training data points were for training KDA/RMSE and the remaining 25% of the data points were inserted one by one by adaptive KDA/RMSE. The kernel parameter $\gamma = 2^{-36}$ was fixed in order to see the effect of λ. The positive Δ values indicate the cases when the prediction accuracy of KDA/RMSE is higher than that of KDA/MSE using the SVD.

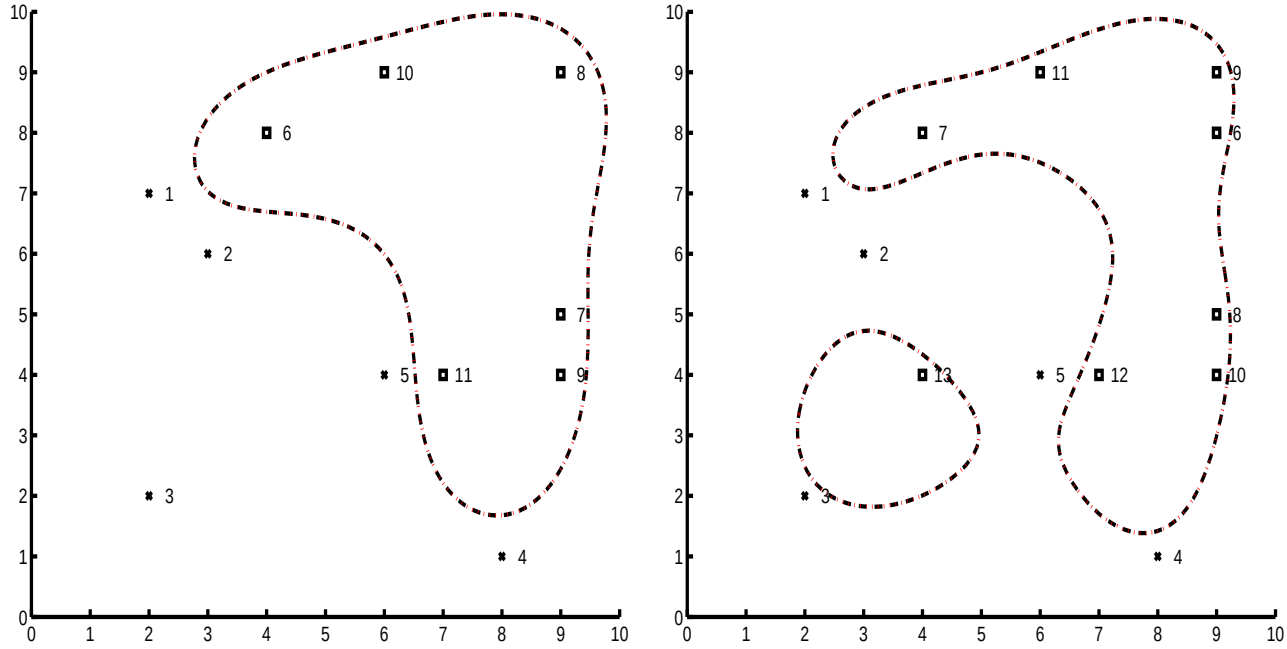

Figure 2: Classification results on an artificial data set of 12 items after deleting the 12th data point (left panel) and appending a data point (right panel). The dash-dotted contour presents a decision boundary of the adaptive KDA/RMSE and the dashed contour presents that of the KDA/RMSE using recomputing solution vector from scratch. The two lines coincide exactly since the results are identical. The radial basis function (RBF) kernel with parameter $\gamma = 0.1$ was used. The regularization parameter was set to $\lambda = 10^{-7}$.

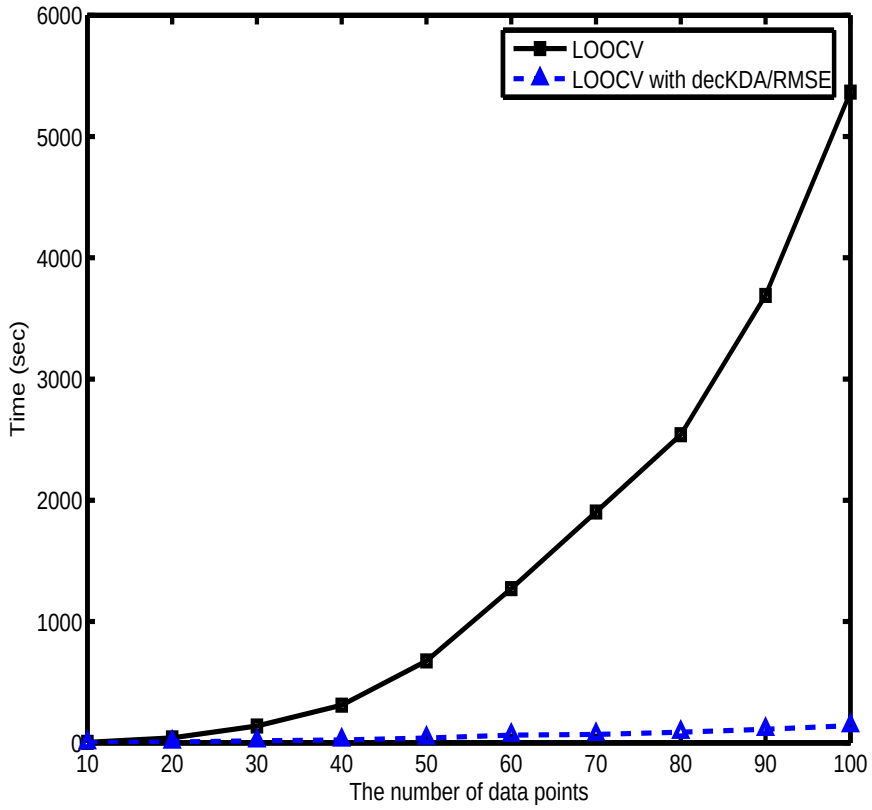

Figure 3: The computation time of leave-one-out cross validation (LOOCV) for different numbers of data points. The solid line represents the computation time of ordinary LOOCV and the dashed line represents that of LOOCV using decremental KDA/RMSE, denoted decKDA/RMSE.