An improved Generalized Discriminant Analysis for Large-scale data set

Weiya Shi, Yue-Fei Guo, Cheng Jin, Xiangyang Xue,
Department of Computer Science and Engineering, Fudan University, Shanghai, China
{wyshi, yfguo, chengjin, xyxue}@fudan.edu.cn

Abstract

In order to overcome the computation and storage problem for large-scale dataset, an efficient iterative method of Generalized Discriminant Analysis is proposed. Because sample vectors cannot explicitly be denoted in kernel space, some mathematical tricks are firstly used to transform the kernel matrix. Then, the columns of transformed matrix are used for iterative algorithm to extract nonlinear discriminant vectors. The proposed method reduces space complexity from $o(m^2)$ to $o(m)$ and its effectiveness is validated from experimental results.

1. Introduction

Generalized discriminant analysis (GDA) [1] is the generalization of Linear Discriminant Analysis (LDA) [2] in kernel space. Its main idea is to map the data set from the input space into high-dimensional (even infinite dimensional) feature space. Thus, the nonlinear components can be extracted using the traditional linear algorithm in the mapping feature space. In practice, the mapping function is neither calculated nor kept explicitly, but realized implicitly via the kernel trick, which the inner product between sample vectors cannot explicitly be denoted in kernel space, so the mapping function is realized implicitly via kernel trick. A possible way of achieving this is to use a kernel function $K(x, y) = \phi(x) \cdot \phi(y)$, where $\phi$ is a mapping from the input space to a feature space of infinite dimension.

But Generalized discriminant analysis is infeasible when faced with large-scale data set. The major problem is that it needs to store the kernel matrix (also called Gram matrix), which takes the space complexity of $o(m^2)$, where $m$ is the number of samples. In addition, it needs the time complexity of $o(m^2)$ to eigen-decompose the Gram matrix. The computation and storage problem impedes its application in large-scale data set.

There are some methods to overcome these disadvantages. It was proposed that QR decomposition could be combined with the kernel discriminant analysis [8]. However, it only finds the nonlinear features in the range space of between-class scatter matrix. Some methods use some representative samples to approximate the whole data set [6], which cannot assure the extraction of optimal discriminant vectors.

In this paper, an efficient iterative method of Generalized Discriminant Analysis is proposed for large-scale data set. It is not necessary to eigen-decompose the matrix like the algorithm of standard GDA. Instead, a new matrix, called Gram-power matrix, is firstly constructed using the original Gram matrix. Then, the diagonal matrix is decomposed into two matrices using Cholesky decomposition. Thus, the column of transformed matrix can be treated as the input sample for iterative algorithm [7] in kernel space. The advantage of the proposed algorithm is that it does not need to store the whole Gram matrix and reduce the space complexity from $o(m^2)$ to $o(m)$. The quick converge of the algorithm can be ensured through theoretical proof in [10]. Experimental results demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows: section 2 gives a short review of the GDA. Then, the proposed method is described in section 3. The experimental evaluation of the proposed method is given in the section 4. Finally we conclude with a discussion.

2. Review of generalized discriminant analysis

Let $X = \{x_{ij}\}_{i=1,...,c; j=1,...,n_i}$ be the data matrix in input space, where $x_{ij}$ is a d-dimensional vector and $n_i$ is the number of data samples in $i$th class and $m = \sum_{i=1}^{c} n_i$. There exists a mapping function $\phi$, which projects the data into high-dimensional (even infinite dimensional) Reproducing Kernel Hilbert Space (RKHS).

$$\phi: \mathbb{R}^d \rightarrow F \ , \ \ \ \ x_{ij} \mapsto \phi(x_{ij})$$ (1)

The whole data matrix can be transformed into $\Phi(X) = \{\phi(x_{ij})\}_{i=1,...,c; j=1,...,n_i}$ by the mapping function $\phi$. In practice, the mapping function $\phi$ does not need to be known explicitly but performed implicitly via kernel trick. A positive definite kernel function $\kappa(\cdot, \cdot)$ is used to calculate the
dot product between mapped sample vectors, where $\kappa(\cdot, \cdot)$ is given by $\kappa(x, y) = \phi(x)^T \phi(y)$. Thus, the entry of the Gram matrix $K$ is defined as $\kappa(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.

In kernel space, the between-class, within-class, and total scatter matrices $S^o_b, S^w_b$, and $S^o_t$ are defined respectively as:

$$
\begin{align*}
S^o_b &= \sum_{i=1}^{c} n_i (\mu_i^o - \mu^o)(\mu_i^o - \mu^o)^T \\
S^w_b &= \sum_{i=1}^{c} \sum_{j=1}^{n} (\phi(x_i) - \mu_i^o)(\phi(x_i) - \mu_i^o)^T \\
S^o_t &= \sum_{i=1}^{c} \sum_{j=1}^{n} (\phi(x_i) - \mu^o)(\phi(x_i) - \mu^o)^T
\end{align*}
$$

(2)

where $\mu_i^o$ is the samples mean of $i$th class and $\mu^o$ is the mean of all samples.

Following the Fisher criterion, GDA also aims to optimize the following problem:

$$
J(\omega) = \frac{\omega^T S^o_w \omega}{\omega^T S^o_t \omega}
$$

(3)

The solutions $\omega$ can be expanded generally using all the mapped sample vectors in the data set $\Phi(X) = \{\phi(x_i)\}_{i=1,...,c,j=1,...,n_i}$ as:

$$
\omega = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \alpha_{ij} \phi(x_i),
$$

(4)

where $\alpha = \{\alpha_{ij}\}_{i=1,...,c,j=1,...,n_i}$ is the span coefficient. By substituting Eq. 2, Eq. 4 into Eq. 3, we can get the following formula of standard Generalized discriminant analysis:

$$
J(\alpha) = \frac{\alpha^T KW K \alpha^T}{\alpha^T \lambda \omega,}
$$

(5)

where the Gram matrix $K = \Phi(X)^T \Phi(X)$, and $W = diag(W_1, ..., W_c)$, $W_i$ is a $c_i \times c_i$ symmetric matrix with all entries equal to $1/n_i$ ($n_i$ is the number of class $c_i$).

In the first phase, the Gram matrix $K$ is eigen-decomposed, which produces $K = U \Gamma U^T$ ($U$ is the matrix of normalized eigenvectors and $\Gamma$ is the diagonal matrix of nonzero eigenvalues). After substituting $K = U \Gamma U^T$ into the Eq. 5, it can be changed into:

$$
J(\beta) = \frac{\beta^T (U^T W U) \beta^T}{\beta^T \lambda \omega, \beta^T}
$$

(6)

Where $\beta = U^T \alpha$.

The second phase is to eigen-decompose the matrix $U^T W U$. Once the eigenvector $\alpha$ has been calculated, we can achieve discriminant vectors using Eq. 4. For a test sample $x$, the nonlinear discriminant feature is:

$$
(\alpha, \phi(x)) = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \alpha_{ij} \phi(x_i) = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \alpha_{ij} \kappa(x_i, x)
$$

(7)

3. Proposed Method

Iterative algorithm generally continuously input each sample vector to achieve the discriminant vectors. But the sample vector in feature space cannot explicitly be denoted because the mapping function is unknown. As a result, it is not easy to use the iterative algorithm in Reproducing Kernel Hilbert Space (RKHS). In order to solve the problem and introduce the proposed method, we need to transform the matrix $K$ and $U^T W U$.

3.1 Transformation of the matrix $K$:

Theorem 1: Matrix $H$ and $H^2$ have same eigenvectors and different eigenvalues.

Proof: Suppose $\omega$ and $\lambda$ is the eigenvector and eigenvalue of $H$, respectively. It follows that:

$$
H\omega = \lambda \omega,
$$

(8)

$$
H^2 \omega = HH\omega = \lambda H\omega = \lambda^2 \omega,
$$

(9)

As a result, the eigenvector and eigenvalue of $H^2$ is $\omega$ and $\lambda^2$, respectively.

Because the kernel matrix $K$ is positive semi-definite [5], we can construct another Gram-power matrix:

$$
G = KK^T
$$

(10)

where $K(x_i)$ is the samples mean of $i$th class and $\kappa^o$ is defined as $K(x_i, ..., K(x_m))$. According to the theorem 1, newly constructing Gram-power matrix $G$ and the Gram matrix $K$ have same eigenvectors, but having different eigenvalues $\lambda_G$ and $\lambda_K$ ($\lambda_G = (\lambda_K)/m$). Thus, when faced with large-scale data set, we do not need to eigen-decompose the kernel matrix $K$, but computing the eigenvector of Gram-power matrix $G$.

In order to achieve the eigenvectors of Gram-power matrix $G$, each column $K(x_i)$ of matrix $K$ can be treated as the input sample vector of iterative algorithm [4] in kernel space. After some iterations, the eigenvectors of $G$ can be quickly calculated. Thus, we only process the "sample" vector $K(x_i)$ (which space complexity is $o(m)$) in each step, and need not to store the whole kernel matrix.
3.2 Transformation of the matrix $U^T WU$:

Lemma 1: Any symmetrical Matrix $W$ can be decomposed into the two matrices, one of which is the transpose of another, using Cholesky decomposition.

Thus, the matrix $U^T WU$ can be represented as:

$$
U^T WU = U^T \eta^T \eta U = (\eta U)^T \eta U
$$

$$
= (\eta_1, ..., \eta_r)^T (U_1, ..., U_n)^T (\eta_1, ..., \eta_r)^T (U_1, ..., U_n)
$$

$$
= (R(x_1), ..., R(x_n))(R(x_1), ..., R(x_n))^T
$$

$$
= \sum_{i=1}^n R(x_i)R(x_i)^T
$$

(11)

where $R(x_i)$ is the rank of symmetrical Matrix $W$, and $n$ is the number of discriminant vectors. We can also treat the column $R(x_i)$ as the 'sample' vector in the kernel space and use it for the iterative algorithm.

3.3 Iterative algorithm in kernel space:

Having transformed the matrix $K$ and $U^T WU$, each column $K(x_i)$ or $R(x_i)$ can be treated as "sample" vector in feature space. Iterative algorithm can be used to compute the discriminant vector. There are some iterative methods to extract discriminant components, such as GHA [4], APEX [3]. But these methods converge slowly. Weng [7] proposed a candid covariance-free incremental principal component analysis (CCIPCA), which uses a well-known statistical concept — efficient estimation. It converges faster than other iterative algorithms, while computation complexity is also low [10]. Therefore we choose the CCIPCA algorithm as the iterative tool for the proposed method.

By definition, eigenvectors $U_G$ and eigenvalues $\lambda_G$ of newly constructing Gram-power matrix $G$ correspond to following equation:

$$
\omega(n) = \lambda_G U_G = GU_G,
$$

(12)

where $\omega(n)$ is the $n$th step estimate of eigenvector. After the $\omega(n)$ estimated, we can easily calculate the eigenvector $U_G = \omega/||\omega||$ and eigenvalue $\lambda_G = ||\omega||$. The formula is derived as follows:

$$
\omega_1(n) = \frac{1}{\pi} GU_G = \frac{1}{\pi} \sum_{t=1}^n K_i(x_t) K_i(x_t)^T \frac{\omega_1(t-1)}{||\omega_1(t-1)||}
$$

$$
= \frac{1}{\pi} \sum_{t=1}^{n-1} K_i(x_t) K_i(x_t)^T \frac{\omega_1(t-1)}{||\omega_1(t-1)||}
$$

$$
+ \frac{1}{\pi} K_i(x_n) K_i(x_n)^T \frac{\omega_1(n-1)}{||\omega_1(n-1)||}
$$

$$
= \frac{1}{\pi} \omega_1(n-1) + \frac{1}{\pi} K_i(x_n) K_i(x_n)^T \frac{\omega_1(n-1)}{||\omega_1(n-1)||}
$$

(13)

where $i$ denotes the $i^{th}$ kernel principal components and $K_i(x_i)$ is the input sample of $t$ instance.

The higher order eigenvector can be calculated similarly by using the residual data vector, which is subtracted from data its projection on the estimated low order eigenvector ($K_i(x_i) = K(x_i)$):

$$
K_{i+1}(x_n) = K_i(x_n) - K_i(x_n) \frac{\omega_1(n)}{||\omega_1(n)||},
$$

(14)

By iteratively calculating, we can get the approximate eigenvectors and eigenvalues.

The algorithm is shown in Algorithm 1:

Algorithm 1 Computation process of the proposed method

1. initially the first $k$ eigenvector using first $k$ samples.
2. for $t = 1 : m$ do following steps:
3. for $i = 1 : k$ do following steps:
4. for each input data $x_i$ , calculate the correspond column $K_i(x_i)$ representing the input vector for the algorithm.
5. using Eq. 13 Eq. 14 to calculate the first $k$ principal components
6. go to step 3
7. go to step 2 for some iterations $p$
8. output the eigenvector $U_G$ and eigenvalue $\lambda_G$.

The algorithm of the second Phase is the same as the first phase except that $K(x_i)$ is substituted with $R(x_i)$.

3.4 Complexity analysis of proposed methods:

In whole process, we do not need to eigen-decompose the Gram matrix, which space complexity is $O(m^2)$ and time complexity is $O(m^3)$. Instead, we process the 'sample' $K(x_i)$, $R(x_i)$ stepwise, which space complexity is $O(m)$ and time complexity is $O(npn)$, where $m$ is the sample number, $n$ is the number of extracted discriminant vectors and $p$ is the number of iterative. It only needs some iteration to achieve the discriminant vectors.

4. Experiment Results

To demonstrate the effective of the proposed method, we perform the experiments on iris dataset using the standard GDA (using the original Matlab source code [1] available from 1) and our method, respectively. The experiment results are given in Fig. 1. It illustrates the projection of iris data on the first 2 discriminant axis. From the result, the proposed method can get almost similar performance with standard GDA.

In addition, we use USPS data set to validate the feasibility of the proposed method when faced with the large-scale

1Available at http://www.kernel-machines.org/code/GDA.m
In this paper, an efficient iterative algorithm for Generalized discriminant analysis is proposed to solve the problem of large-scale data set. The ‘sample’ vector in kernel space is obtained using some mathematic trick. Then the iterative algorithm is used to extract the nonlinear discriminant vector. The method needs not to store the Gram matrix. The space and time complexity reduce to \( o(m) \) and \( o(kpm) \), respectively. More importantly, its effectiveness still exists where the standard GDA cannot succeed.

5. Conclusions

Acknowledgment

This work was supported in part by MoE research Fund under contract 104075, MoST Support Program under contract 2007BAH09B03 and the National High Technology Research and Development Program of China under contract 2007AA01Z176.

References