2016 15th IEEE International Conference on Machine Learning and Applications

Linear Discriminant Analysis for Large-Scale data : Application on Text and Image data

Nassara ELHADJI ILLE GADO, Edith GRALL-MAËS, Malika KHAROUF
University of Champagne, University of Technology of Troyes
Charles Delaunay Institute (ICD)/LM2S, 12 rue Marie Curie, 10000 Troyes, France
e-mail : firstname.familyname@utt.fr

***Abstract*—Linear Discriminant Analysis (LDA) is a technique which is frequently used to extract discriminative features that preserve the class separability. LDA involves matrices eigen decomposition which can be computationally expensive in both time and memory, in particular when the number of samples and the number of features are large. This is the case for text and image data sets where the dimension can reach in order of hundreds of thousands or more.**

In this paper, we propose an efficient algorithm Fast-LDA to handle large scale data for discriminant analysis. The proposed approach uses a feature extraction method based on random projection to reduce the dimensionality and then perform LDA in the reduced space.

By reducing data dimension, we reduce the complexity of data analysis. The accuracy and the computational time of the proposed approach are provided for a wide variety of real image and text data sets. The results show the relevance of the proposed method compared to other methods.

I. Introduction

Learning large scale data sets is a main concern in the research topics due to curse of dimensionality. In machine learning area, it has become a challenge to handle high dimensional data or modern massive data sets especially to limit the computing time and keeping good performance of an algorithm. One of the most common solutions to learn large data is to reduce the original dimension to find an almost optimal space, where algorithms can be easily developed for any specific application. Dimension reduction strategies consist on reducing the data space by eliminating irrelevant and redundant information in the data by removing the impact of noisy/less-relevant features and improve the robustness of the results. Many methods have been proposed for dimension reduction. The most popular techniques used are principal component analysis (PCA)[1] and random projection (RP)[2].

This paper deals with the problem of supervised classification of large scale data. The linear discriminant analysis (LDA)[3] is a well-known method, that searches for the project axes on which data points of different classes are far from each other while data points of the same class are close to each other. The optimal transformation of LDA can be computed by applying an eigendecomposition, which could be very expensive in both time and memory for high dimensional data. Various approaches have been proposed to outperform LDA in high dimension including PCA+LDA [4], LDA/QR [5](QR for QR decomposition), RP+LDA [6], [7] to name a few.

We propose a method that combines two parts. The first step is a projection stage that consists of finding a representation of the original data by using fast approximate singular value decomposition [8]. The second step is a classical linear discriminant analysis [9].

The remainder of this paper is organized as follows : in section II, we give a brief description of LDA, and Fast Approximate-SVD methods. In section III, we describe the proposed approach. We provide in section IV the numerical results on three real data sets. Finally we conclude in section V.

Notations. In the following, capital characters denote matrices and lower case symbols represent vectors. The transpose operator is denoted by $(.)'$. The $n \times n$ identity matrix is denoted by I_n.

II. LDA and Fast Approximate-SVD Basics

A. Classical linear discriminant analysis

We give a brief LDA basics. Consider the following supervised multi-class classification problem : we dispose of a set of N labelled data belonging to K classes $\{C_1, C_2, ..., C_K\}$ with class sizes $\{N_1, N_2, ..., N_K\}$, where $N_1 + N_2 + \cdots + N_K = N$. $X = \{x_1, x_2, ..., x_N\}$, $x_i \in \mathbb{R}^{1\times d}$, is the observed sample and $y_i \in \{1, 2, \cdots, K\}_{i=1,...,N}$ is the given class membership for x_i. The goal of LDA is to build a classifier based on the training set $X \in \mathbb{R}^{N\times d}$ to predict the class label of a new unlabelled set $X_u = \{x_1^u, x_2^u, ..., x_{N_u}^u\}$. The LDA objective is to seek a projection matrix W that maximizes the following ratio

$$J(W) = \underset{W}{\operatorname{argmax}} \ \frac{\det(W' S_b W)}{\det(W' S_w W)} \tag{1}$$

such that the similarity within data S_w is minimized and the dissimilarity between classes S_b is maximized [10]. The matrices S_b and S_w are the between class scatter and the within class scatter defined by (2). The optimal discrimination projection W can be obtained by computing the eigenvectors of the matrix $S_w^{-1} S_b$ [11]. Since the rank of S_b is bounded by $K-1$ [12], there are at most $K-1$ eigenvectors corresponding

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.125

to non zeros eigenvalues.

$$S_b = \frac{1}{N}\sum_{i=1}^{K} N_i (m_i - m)'(m_i - m),$$

$$S_w = \frac{1}{N}\sum_{i=1}^{K}\sum_{x_j \in C_i} (x_j - m_i)'(x_j - m_i), \quad (2)$$

$m = \frac{1}{N}\sum_{i=1}^{N}(x_i)$ is the mean of the training data set and m_i is the mean of the data in the class C_i. The time complexity and the memory requirement increase with N and d. Then, when N and/or d is large, it is difficult to perform LDA.

B. Fast Approximate SVD

Approximate SVD is a process of finding a rank-r approximation as forcing the original data matrix to provide a shrunken description of itself. The problem is used for mathematical modeling and data compression. Let $X \in \mathbb{R}^{N\times d}$ be the data matrix, and let the SVD of X be of the form :

$$X = U\Sigma V' \quad (3)$$

where, $U \in \mathbb{R}^{N\times N}$, $V \in \mathbb{R}^{d\times d}$ and $\Sigma \in \mathbb{R}^{N\times d}$. The matrices U and V are orthogonal. Σ is a semi-diagonal matrix with non-negative real numbers entries $\sigma_1 \geq \ldots \geq \sigma_s > 0$ (singular values) where $s \leq min\{N, d\}$.

Giving a value of r ($r < d$), the truncated form X_r of X is defined by :

$$X_r = \sum\nolimits_{i=1}^{r} u_i v_i' \sigma_i = U_r \Sigma_r V_r'. \quad (4)$$

where only the first r column vectors of U and V and the $r \times r$ sub-matrix are selected. The form X_r in (4) is mathematically guaranteed to be the optimal r-approximation of X [8]. Due to the orthogonality of U_r and V_r, the matrix XV_rV_r' (resp. $U_rU_r'X$) has rank at most r and approximate X. The time complexity of (4) is $O(Nd\ min\{N, d\})$ which makes it infeasible if $min\{N, d\}$ is large.

To speed up the calculation of X_r we use fast approximate SVD algorithm recently used in [8]. The principle of fast approximate SVD algorithm, which combines random projection and approximate SVD methods, is the following. Consider the subspace spanned by a random projection $Z = X \times R$, where R is an $d \times p$ random matrix (saying Gaussian). In [13], the authors shows that by projecting X onto the row space of Z, and then finding the best rank-r approximation to this new space (i.e. the truncated SVD), a good approximation onto the best rank-r approximation of X itself is obtained. Thus the algorithm of fast approximate SVD takes as input the matrix X, an integer r which has to be larger than 2 and smaller than the rank of X, and an integer p larger than r. The error in the approximation is directly linked to p (details about the error bound can be found in [8]). The algorithm is the following:

1) Generate an $d \times p$ random matrix $R \sim \mathcal{N}(0, I_p)$,
2) Compute the matrix $Z = XR$,
3) Orthonormalize Z to obtain Q of size $N \times p$,
4) Set G (of size $d \times r$) as the top r right singular vectors of $Q'X$.

Then G can be used as a projection matrix for dimension reduction.

III. The proposed approach

The proposed approach seeks a space of dimension r from the original space on which we can efficiently perform LDA. If $X_r = XGG'$ is a low rank approximation of some matrix X, $\tilde{X} = XG$ can be considered as a reduced form of X which contains the final reconstructed extracted features and then it can be used to perform LDA.

In fast approximate SVD, the fourth step of the algorithm needs the eigen decomposition of a matrix $Q'X$ of size $p \times d$ which can be computationally expensive if d is large. The rank of $Q'X$ is at most equal to d (since $p < d$ for dimension reduction), so we do not need to compute all the eigenvectors. Here we use an efficient SVD algorithm which allows to find the matrix G much faster.

From the equation (3) and for any matrix X, it can be easily seen that $XX' = U\Sigma^2U'$. Thus, the left singular vectors of X are the eigenvectors of XX'. Therefore, it is simply sufficient to compute the eigen decomposition of XX' and then retrieve V from $\Sigma^{-1}U'X = V'$. In the same way, the top right singular vectors of $B = Q'X$, *i.e* G, can be efficiently computed using this trick.

This stage is very important for computational complexity reason in the fast approximate SVD algorithm and allows to achieve a significant time saving (see section II-B). We add this efficient SVD form to the fast approximate SVD and call this algorithm Fast Efficient-SVD which is presented as follows :

1) Generate an $d \times p$ random matrix $R \sim \mathcal{N}(0, I_p)$,
2) Compute the matrix $Z = XR$,
3) Orthonormalize Z to obtain Q of size $n \times p$,
4) Compute $B = Q'X$ of size $p \times d$,
5) Compute the eigenvectors of $T = BB'$ such as $T = H\Delta H'$,
6) Compute $G = (\Delta^{-1}H'B)'$,
7) Set G (of size $d \times r$) as the top r right singular vectors.

The Fast Efficient-SVD algorithm provides G such that XGG' is a low rank approximation of X. The matrix $\widetilde{X} = XG$ is a new representation of X in the reduced space. In the new space, according to $\widetilde{X}$, the covariance matrix $\widetilde{S}$ can be written as

$$\begin{aligned}\tilde{S} &= \frac{1}{N}(\widetilde{X} - \widetilde{m})'(\widetilde{X} - \widetilde{m}) \\ &= \frac{1}{N}(XG - mG)'(XG - mG) \\ &= \frac{1}{N}G'(X - m)'(X - m)G \\ &= G'SG \end{aligned} \quad (5)$$

Similarly we get :

$$\widetilde{S}_w = G'S_wG \quad \text{and} \quad \widetilde{S}_b = G'S_bG \quad (6)$$

The new LDA objective is given by :

$$J(\widetilde{W}) = \frac{\det(\widetilde{W}'\widetilde{S}_b\widetilde{W})}{\det(\widetilde{W}'\widetilde{S}_w\widetilde{W})},$$

where

$$\widetilde{W}'\widetilde{S}_b\widetilde{W} = \widetilde{W}'G'S_bG\widetilde{W} = (\widetilde{W}'G')\ S_b\ (G\widetilde{W}) = W'S_bW$$

with $W = G\widetilde{W}$. The optimal projection matrix $\widetilde{W}$ is the eigenvectors corresponding to the largest eigenvalues of $\widetilde{S}_w^{-1}\widetilde{S}_b$. The obtained matrix $\widetilde{W}$ is a good approximation of W as far as $\widetilde{X}$ is a good approximation of X.

In conclusion, our proposed Fast-LDA method is obtained by combining the two following approaches : (1) feature extraction by Fast Efficient SVD and (2) applying classical LDA in the reduced feature space. Note that the so called Fast-LDA algorithm can achieve the linear discriminant analysis with very large matrix. The algorithm 1 gives the main steps of Fast-LDA.

Algorithm 1 *Fast LDA algorithm*

INPUTS : X, Y, p, r and μ
OUTPUT : $\widetilde{W}$

1) Compute $G = \textit{Fast-Efficient-SVD}(X, p, r)$,
2) Project X using G to obtain $\tilde{X} = XG$,
3) Calculate $\widetilde{S}_w$ and $\widetilde{S}_b$ from $\tilde{X}$,
4) Determine $\widetilde{W}$ by solving $\widetilde{S}_b\widetilde{W} = \lambda\widetilde{S}_w\widetilde{W}$
5) Return $\widetilde{W}$.

If the scatter matrix $\widetilde{S}_w$ is singular, we perform a regularized process to solve the singularity problem, *i.e*, when the eigendecomposition of $\widetilde{S}_w^{-1}\widetilde{S}_b$ states that $\widetilde{S}_w$ is singular, we compute $(\widetilde{S}_w + \mu I_p)^{-1}\widetilde{S}_b$ where μ is a regularized term. Note that $(\widetilde{S}_w + \mu I_p)^{-1}$ involves to add a diagonal term to S_w to make sure that very small eigenvalues are bounded away from zero, which ensures the numerical stability when computing the inverse of $\widetilde{S}_w$.

IV. Experimental Results

In this section, the performances of Fast-LDA algorithm based on experiments on real data for documents and image classification are given. All experiments have been performed on P4 2.7GHz Windows7 machine with 16GB memory. We have used Matlab routine for programming.

A. Data sets

We evaluate the effectiveness of the proposed algorithm Fast-LDA with small and large dimensional data sets. Experiment have been set-up on COIL20 image data, TDT2 and Reuters21578 documents data. All data sets can be download at http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html. The first data set has smaller number of features whereas the last two data sets have very large number of features. The statistics of data sets are listed in Table I.

COIL20: : This data set contains 1440 sample images of 20 different subjects. The size of each image is (32×32) pixels, with 256 grey scale levels per pixel. Thus, each subject is represented by a 1024-dimensional vector.

Reuters21578: The corpus contains 8293 documents in 65 categories with 18933 distinct terms.

TABLE I
Statistics of data sets

data	size (n)	dim (d)	# of classes
T2TD	9394	36771	30
Reuters21578	8293	18933	65
COIL20	1440	1024	20

TDT2: (Nist Topic Detection and Tracking corpus) This subset is composed of 9394 documents in 30 categories with 36771 features.

B. Compared methods

In order to access the relevance of the proposed **Fast-LDA** method, we have compared its performance with three other methods which are listed below:

LDA/QR [5] which is a variant of LDA. It uses the QR decomposition of the centered data matrix.

NovRP is a random projection for LDA presented in [6]. It uses random map with a regularisation form.

SRDA is the spectral regression discriminant analysis method presented in [14]. We downloaded the code of SRDA directly at the web page of the author. We have applied the iterative solution with LSQR on document data sets and applied normal equations solution on COIL20 data. We set $\alpha = 1$ and LSQR iteration number equal to 15.

C. Experiments and results

For the proposed method, we set $\mu = 1$ and $p = 200$ on T2TD and Reuters2158 data. As the original dimension of COIL20 is relatively small, we set $p = 50$. We set $r = \frac{4}{5}p$ for each data set. A random subset with TN=[5% 10% 20% 30% 40% 50%] samples per each i-th class was selected with labels to form a training set of size $N = TN(\%) \times N_i \times K$ and the rest was used as a testing set. For all experiments, we averaged the results over 20 iterations and retained the mean value.

The classification accuracy and the run time of individual approach on each dataset are reported from table II to VII. We report the running time spent to find the final projection space W (for training) for all methods.

One of the striking observations from tables III and V is that our proposed approach Fast-LDA has very fast run times whenever we increase the train size. More precisely in high dimensionality, with a large amount of samples in the training set, Fast-LDA is very efficient in computation time. Experiments indicate that running our algorithm with a relative small number of features (equal to r) achieves almost the same separation results as SRDA for large dimension (see table II and IV).

In the case of small dimension (COIL20 dataset), LDA/QR achieves the best run time (see table VII) whereas the classification results of Fast-LDA achieves better accuracy (see table VI).

TABLE II
ACCURACY RATE ON T2TD (MEAN ± STD %)

Train Size	NovRP	SRDA	LDA/QR	Fast-LDA
5%	65.14±1.37	86.84±0.47	86.60±0.82	**86.95±0.66**
10%	78.06±0.89	92.85±0.30	92.46±0.47	**93.30±0.40**
20%	83.46±0.60	**95.70±0.20**	94.70±0.27	95.41±0.25
30%	84.68±0.61	**96.58±0.21**	95.44 ±0.24	96.00±0.28
40%	85.22±0.72	**96.85±0.16**	95.65±0.18	96.20±0.24
50%	85.73±0.59	**97.10±0.20**	95.68 ±0.28	96.26±0.23

TABLE III
COMPUTATIONAL TIME ON T2TD (S)

Train Size	NovRP	SRDA	LDA/QR	Fast-LDA
5%	1.30	0.47	**0.37**	0.61
10%	1.31	0.61	**0.63**	0.68
20%	1.39	0.85	1.18	**0.83**
30%	1.41	1.06	1.69	**0.94**
40%	1.45	1.21	2.18	**1.05**
50%	1.48	1.38	2.75	**1.18**

TABLE IV
ACCURACY RATE ON REUTERS21578 (MEAN ± STD %)

Train Size	NovRP	SRDA	LDA/QR	Fast-LDA
5%	32.08±6.88	74.02±0.71	67.82±0.92	**75.14±1.27**
10%	63.34±0.95	80.24±0.46	72.53±0.87	**83.48±0.70**
20%	70.21±0.82	85.44±0.32	78.44±1.02	**88.16±0.25**
30%	72.36±0.73	88.00±0.37	82.10±0.86	**89.02±0.44**
40%	73.03±0.96	**89.66±0.31**	84.47±0.83	89.62±0.52
50%	73.71±0.68	**90.86±0.36**	86.38±0.51	90.05±0.50

TABLE V
COMPUTATIONAL TIME ON REUTERS21578 (S)

Train Size	NovRP	SRDA	LDA/QR	Fast-LDA
5%	0.69	0.33	**0.20**	0.35
10%	0.71	0.49	**0.38**	0.39
20%	0.72	0.67	0.70	**0.43**
30%	0.78	0.87	1.09	**0.52**
40%	0.77	0.96	1.33	**0.57**
50%	0.77	1.21	1.71	**0.61**

TABLE VI
ACCURACY RATE ON COIL20 (MEAN ± STD %)

Train Size	NovRP	SRDA	LDA/QR	Fast-LDA
5%	66.04±2.49	76.79±2.50	76.08±2.93	**79.43±2.94**
10%	79.57±2.08	82.80±2.04	81.52±1.71	**87.08±1.95**
20%	87.92±1.88	88.22±1.58	86.22±2.86	**92.06±1.38**
30%	91.13±1.06	91.26±1.10	89.49±1.93	**94.06±0.83**
40%	91.86±1.00	92.52±1.23	91.07±2.24	**95.08±0.98**
50%	91.85±1.28	92.85±0.88	91.80±2.26	**94.42±1.10**

V. CONCLUSION

In this paper we present a new method based on LDA which can be stably and efficiently done in both high and small dimensional spaces. The proposed approach consists of finding an approximation of the original space in a lower rank. It uses efficient approximate SVD to get the projection space before performing classical LDA. Experiments on three real word data have been done and the results have been compared with three other methods. Experimental results are quite encouraging with respect to the performance of the proposed approach.

TABLE VII
COMPUTATIONAL TIME ON COIL20 ($\times 10^{-2}$S)

Train Size	NovRP	SRDA	LDA/QR	Fast-LDA
5%	1.30	**0.32**	0.33	0.80
10%	1.32	0.56	**0.44**	0.90
20%	1.33	1.39	**0.71**	1.16
30%	1.49	2.91	**0.96**	1.46
40%	1.70	5.02	**1.41**	1.90
50%	1.71	6.84	**1.63**	2.11

ACKNOWLEDGMENT

This work is supported by the region of Champagne Ardenne, France on the APERUL project (Machine Learning).

REFERENCES

[1] Elhadji Ille Gado, N., Grall-Maës, E., & Kharouf, M. Linear KernelPCA and K-Means Clustering Using New Estimated Eigenvectors of the Sample Covariance Matrix. In IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 386-389.December 2015.

[2] Achlioptas, D. , Database-friendly random projections, in ACM Symposium on the Principles of Database Systems, pp. 274-281, 2001.

[3] Yu, H., & Yang, J. A direct LDA algorithm for high-dimensional data-with application to face recognition. Pattern recognition, pp. 2067-2070. 2001.

[4] Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D. L., & Weng, J. Discriminant analysis of principal components for face recognition. In Face Recognition pp. 73-85.1998.

[5] Ye, J., & Li, Q. LDA/QR: an efficient and effective dimension reduction algorithm and its theoretical foundation. Pattern recognition,pp. 851-854. 2004.

[6] Liu, H., & Chen, W. S. A novel random projection model for Linear Discriminant Analysis based face recognition. In Wavelet Analysis and Pattern Recognition (ICWAPR) pp. 112-117. 2009.

[7] Arriaga, R. I., & Vempala, S. An algorithmic theory of learning, robust concepts and random projection. In Proceedings of the 40th Annual IEEE Symposium on Foundation of Computer Science, pp. 616-623.1999.

[8] Boutsidis, C., Zouzias, A., Mahoney, M. W., & Drineas, P. Randomized dimensionality reduction for-means clustering. Information Theory, IEEE Transactions, pp. 1045-1062. 2015.

[9] Friedman, J. H. Regularized discriminant analysis. Journal of the American statistical association, pp. 165-175.1989.

[10] Welling, M. Fisher linear discriminant analysis. Department of Computer Science, University of Toronto, 3. 2005.

[11] Chen, L. F., Liao, H. Y. M., Ko, M. T., Lin, J. C., & Yu, G. J. A new LDA-based face recognition system which can solve the small sample size problem. Pattern recognition, pp. 1713-1726. 2000.

[12] Cai, D., He, X., & Han, J. Training linear discriminant analysis in linear time. IEEE 24th International Conference (ICDE), pp. 209-217. April, 2008.

[13] Vempala, S.S. The random projection method. DIMACS series in discrete mathematics and theoretical computer science,vol 65. American Mathematical Society, Providence RI. 2004.

[14] Cai, D., He, X., & Han, J. SRDA: An efficient algorithm for large-scale discriminant analysis. IEEE transactions on knowledge and data engineering, pp. 1-12, 2008.