Linear Discriminant Analysis based on Fast Approximate SVD

Nassara Elhadji Ille Gado, Edith Grall-Maës and Malika Kharouf
University of Champagne / University of Technology of Troyes,
Charles Delaunay Institute(ICD) UMR 6281 UTT-CNRS / LM2S, Troyes, France
{nassara.elhadji_ille_gado, edith.grall, malika.kharouf}@utt.fr

Keywords: LDA, Fast SVD, Dimension Reduction, Large Scale Data.

Abstract: We present an approach for performing linear discriminant analysis (LDA) in the contemporary challenging context of high dimensionality. The projection matrix of LDA is usually obtained by simultaneously maximizing the between-class covariance and minimizing the within-class covariance. However it involves matrix eigendecomposition which is computationally expensive in both time and memory requirement when the number of samples and the number of features are large. To deal with this complexity, we propose to use a recent dimension reduction method. The technique is based on fast approximate singular value decomposition (SVD) which has deep connections with low-rank approximation of the data matrix. The proposed approach, appSVD+LDA, consists of two stages. The first stage leads to a set of artificial features based on the original data. The second stage is the classical LDA. The foundation of our approach is presented and its performances in term of accuracy and computation time in comparison with some state-of-the-art techniques are provided for different real data sets.

1 INTRODUCTION

Linear Discriminant Analysis (LDA) is a well-known supervised technique for feature extraction (Friedman, 1989), (Duda et al., 2012), (Welling, 2005). It has been widely used in many applications such as face recognition (Chen et al., 2005), handwritten code classification (Hastie et al., 2001), text classification (Moulin et al., 2014). The traditional LDA seeks a projection matrix so that data points in different classes are far from each other while those in the same class are close to each other, thus achieving maximum discrimination. To find such optimal projection matrix, LDA involves eigendecomposition of the scatter matrices. For face recognition and documents classification for example, the intrinsic structure of samples can make a scatter matrix singular since the data sets are from a very high-dimensional space. In high dimensional context, the singularity problem and eigendecomposition complexity of the scatter matrices make LDA infeasible.

Many approaches have been proposed to outperform LDA in high dimension (Yu and Yang, 2001) (Ye and Li, 2004) and (Ye et al., 2005). A common way to deal with the curse of dimensionality is to determine an intermediate subspace where optimization problems can be solved efficiently with much smaller size matrices. Dimension reduction strategies consist in eliminating irrelevant information. The most popular techniques proposed for dimension reduction with large scale data sets are principal component analysis (PCA)(Lee et al., 2012) and random projection (RP) (Achlioptas, 2003), (Cardoso and Wichert, 2012).

In this paper, we use a dimension reduction strategy which uses fast approximate singular value decomposition (SVD) (Menon and Elkan, 2011). This technique was also used in (Boutsidis et al., 2015). The principle is to reconstruct some d-dimensional feature space onto its best rank-k approximation for some $k \ll d$. After dimension reduction, it becomes practically easy to handle data in the new reduced feature space. Hence, the proposed appSVD+LDA approach deals with a multi-class supervised classification problem. It consists of outperforming the traditional LDA in a new artificial subspace constructed by fast approximate SVD.

The remainder of this paper is organized as follows: in section 2, we give a brief description of LDA and fast approximate SVD methods. In section 3, we describe the proposed approach appSVD+LDA. In section 4 numerical results supporting the performance of the proposed approach compared to some state-of-the-art methods are presented. Finally in section 5 we conclude the paper.

Gado, N., Grall-Maës, E. and Kharouf, M.
Linear Discriminant Analysis based on Fast Approximate SVD.
DOI: 10.5220/0006148603590365
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 359-365
ISBN: 978-989-758-222-6

2 A BRIEF REVIEW OF LDA AND FAST APPROXIMATE-SVD

2.1 Classical Linear Discriminant Analysis (LDA)

In this section, 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 $\{C_1, C_2, \ldots, C_K\}$ with class size $\{N_1, N_2, \ldots, N_K\}$, where $N_1 + N_2 + \ldots + N_K = N$. $X = \{x_1, x_2, \ldots, x_N\}$ where $x_i \in \mathbb{R}^{1\times d}$ is the observed sample and $\{y_i\}_{i=1,\ldots,N}$, $y_i \in 1 \ldots K$ is the given class membership for x_i. The goal 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 $X_u = \{x^u_1, x^u_2, \ldots, x^u_{N_u}\}$. The LDA objective function is to seek a projection matrix W such that the data points in the new space which belong to the same class are very close while data points in different classes are far from each other (Welling, 2005). W maximizes the following ratio

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

S_b is the between class scatter matrix and S_w is the within class scatter matrix defined by

$$S_b = \sum_{i=1}^{K} N_i (m_i - m)^T (m_i - m),$$
$$S_w = \sum_{i=1}^{K} \sum_{x_j \in C_i} (x_j - m_i)^T (x_j - m_i), \tag{2}$$

where $m = \frac{1}{N}\sum_{i=1}^{N}(x_i)$ is the total sample mean vector, m_i is the mean vector of the i-th class.

The optimal discriminative projection matrix W can be obtained by computing the eigenvectors of the matrix $S_w^{-1} S_b$ (Chen et al., 2005). Since the rank of S_b is bounded by $K-1$, there are at most $K-1$ eigenvectors corresponding to non zeros eigenvalues. The time complexity and the memory requirement increase with N and d. Then, when N and d are large (or d is large), it is difficult to perform LDA.

2.2 Fast Approximate-SVD

Low-rank approximation or approximate SVD is a minimization problem, in which the cost function measures the fit between a given matrix (the data) and an approximating matrix (the optimization variable), subject to a constraint that the approximating matrix has a reduced rank. The problem aims to find a low-rank matrix X_k which approximates the matrix X in some lower rank such as $min_{X_k} \ \| X - X_k \|_F \ \ s.t.\ rank(X_k) = k$ where F indicates the Frobenius norm.

Approximate SVD can be seen as a process of finding a rank-k approximation as forcing the original 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 :

$$X = U\Sigma V^T \tag{3}$$

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 $k \leq min\{N, d\}$ and by using (3), the truncated form X_k of X is defined by :

$$X_k = \sum_{i=1}^{k} u_i {v_i}^T \sigma_i = U_k \Sigma_k {V_k}^T, \tag{4}$$

where only the first k column vectors of U, V and the $k \times k$ sub-matrix are selected. The form X_k in (4) is mathematically guaranteed to be the optimal approximation of X (Boutsidis et al., 2015). Due to the orthogonality of U_k, V_k, the matrix $XV_k{V_k}^T$(resp. $U_k{U_k}^T X$) has rank at most equal to k and approximates X. The computation complexity of (4) is $O(Nd \ min\{N, d\})$ which makes it infeasible if $min\{N, d\}$ is large.

To speed up the computation of the best rank-k approximation of X, it is possible to use a fast approximate SVD algorithm. This algorithm, recently used in (Boutsidis et al., 2015), uses random projection. The principle is the following (Menon and Elkan, 2011) : we consider the subspace spanned by a random projection $Y = X \times R$ where R is a $d \times p$ random matrix. It is shown that by projecting X onto the column space of Y, and then finding the best rank-k approximation to this new space (i.e. the truncated SVD), we get a good approximation to the best rank-k approximation of X itself. Thus the algorithm of fast approximate SVD takes as input the matrix X and integers k and p such that $2 < k < rank(X)$ and $k \leq p \ll d$. The error in the approximation is directly linked to p (details about the error bound can be found in (Boutsidis et al., 2015)). The fast approximate SVD (Fast-AppSVD) algorithm is the following:

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

Then G can be used as a projection matrix.

3 THE PROPOSED APPROACH

The proposed approach proceeds in two steps. Firstly, we perform a feature selection by applying the fast approximate SVD described in the previous section. k-dimensional space obtained in the first step. The proposed approach allows to perform the linear discriminant analysis with very large matrix. The algorithm 1 gives the main steps of our method.

Algorithm 1: appSVD+LDA algorithm.

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

1. Compute $G = \textit{Fast-AppSVD}(X, p, k)$,
2. Project X using G to obtain $\tilde{X} = XG$,
3. Calculate $\widetilde{S}_w$ and $\widetilde{S}_b$ from $\tilde{X}$,
4. Find $\widetilde{W}$ as the eigenvectors of $(\widetilde{S}_w)^{-1}\widetilde{S}_b$ if $\widetilde{S}_w$ is not singular and of $(\widetilde{S}_w + \mu I_p)^{-1}\widetilde{S}_b$ else,
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*, we compute $(\widetilde{S}_w + \mu I_p)^{-1}\widetilde{S}_b$ where μ is a regularized term. Note that $(\widetilde{S}_w + \mu I)^{-1}$ involves to add a diagonal term to $\widetilde{S}_w$ to make sure that very small eigenvalues are bounded away from zero, which ensures the numerical stability when computing the inverse of $\widetilde{S}_w$.

It can be demonstrated that the projection matrix $\widetilde{W}$ is a good approximation of W. The data covariance matrix in the d- original space is given by

$$S = \frac{1}{N}(X-m)^T(X-m).$$

The Fast-AppSVD algorithm provides G such that XGG^T is a low rank approximation of X. The matrix $\tilde{X} = XG$ is a new representation of the original data matrix in the reduced feature space. In the new space, the covariance matrix $\tilde{S}$ can be written as

$$\begin{aligned}\tilde{S} &= \frac{1}{N}(\widetilde{X}-\widetilde{m})^T(\widetilde{X}-\widetilde{m})\\ &= \frac{1}{N}(XG-mG)^T(XG-mG)\\ &= \frac{1}{N}G^T(X-m)^T(X-m)G = G^TSG \qquad (5)\end{aligned}$$

Similarly we get :

$$\widetilde{S}_w = G^TS_wG \quad \text{and} \quad \widetilde{S}_b = G^TS_bG \qquad (6)$$

Then

$$\widetilde{W}^T\widetilde{S}_b\widetilde{W} = \widetilde{W}^TG^TS_bG\widetilde{W} = W^TS_bW$$

with $W = G\widetilde{W}$. The new LDA objective function can be rewritten as follows:

$$J(\widetilde{W}) = \frac{\det(\widetilde{W}^T\widetilde{S}_b\widetilde{W})}{\det(\widetilde{W}^T\widetilde{S}_w\widetilde{W})} = \frac{\det(W^TS_bW)}{\det(W^TS_wW)}. \qquad (7)$$

The optimal projection matrix $\widetilde{W}$ (for simplicity we do not use $\widetilde{W}^*$ for the optimal value) is formed by the largest eigenvalues of $\widetilde{S}_w^{-1}\widetilde{S}_b$.

Then the obtained projection matrix $\widetilde{W}$ should be a good approximation of W as far as $\widetilde{X}$ is a good approximation of X.

4 EXPERIMENTAL RESULTS

In this section, the performances of the proposed algorithm appSVD+LDA are given. The experiments are based on real data sets including face recognition and text classification which can be download at http://www.cad.zju.edu.cn/home/dengcai/Data/data.html. All the experiments have been performed on P4 2.7GHz Windows7 machine with 16GB memory. We have used Matlab routine for programming.

4.1 Data Sets

Two images data sets ORL, COIL20 and two texts data sets TDT2, Reuters21578 have been used in our experiments. The image data have been normalized to have L2-norm equal to 1. For text data, each document have been represented as a term-frequency vector and have been normalized to have L2-norm equal to 1. The statistics of these data sets are listed in Table 1.

COIL20. This data set contains 1440 sample images of 20 different subjects. The size of each image is (32×32) pixels.

ORL. This data set contains 10 different poses of 40 distinct subjects with 4096-dimension (64×64 pixels). The images were taken at different times, ranged from full right profile to full left profile.

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

Reuters21578. These data were originally collected and labeled by *Carnegie Group, Inc. and Reuters, Ltd.* The corpus contains 8293 documents in 65 categories with 18933 distinct terms.

4.2 Experiments

For COIL20, TDT2 and Reuters21578 data sets, a subset TN = [10%, 30%, 50%] of samples per class

Table 1: Statistics of data sets and value of the chosen parameter p.

Statistics of data sets				size of
data sets	samples (N)	dim (d)	# of classes	dim-Red (p)
COIL20	1440	1024	20	20
ORL	400	4096	40	50
Reuters21578	8293	18933	65	80
TDT2	9394	36771	30	80

with labels was selected at random to form the training set. For ORL data, we randomly selected TN = [2, 4, 6] samples per class for training. The rest of samples were used for testing.

We set the regularized parameter $\mu = 0.5$ and $k = p$ for fast approximate SVD on the assumption that $K-1 \leqslant k \leqslant p \ll d$. Table 1 shows for each data set the dimension p that we chose for the intermediate space. Since $K-1$ directions can be generated by LDA, we finally retain $K-1$ vectors of W and then classify the transformed data in the new space of dimension $K-1$.

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

- Direct LDA (DLDA) (Friedman, 1989) which solves the LDA problem in the original space.
- LDA/QR (Ye and Li, 2004) which is a variant of LDA that needs to solve the QR decomposition of a small size matrix.
- NovRP (Liu and Chen, 2009) which is an approach that uses sparse random projection as dimension reduction before performing LDA. The parameters μ and p have been set in the same way as our approach.

4.3 Performance

The experimental results are given from Table 2 to 9 for all data sets highlight above. In these tables the results are averaged over 20 random splits for each TN(%) and report the mean as well as the standard deviation. As the running time is nearly constant we just report the mean value.

Tables 2 and 3 show the performance results on COIL20 data. DLDA achieves the best accuracy in this case whereas its running time is significantly the highest. appSVD+LDA presents a quite good accuracy performance and its running time is nearly 100 times smaller than that of DLDA. The running time of NovRP is the most efficient in this case whereas its accuracy is the lowest one. For ORL data, experimental results are displayed on Tables 4 and 5. As can be seen, appSVD+LDA presents the best accuracy (for 4 and 6 samples) and a low running time. As the dimension in this case is relatively large, the computation time for DLDA is very large (see Table 5).

Reuters21578 and TDT2 are very large data sets. As DLDA needs memory to store the centered and scatter matrices in the original features space, it is infeasible to apply DLDA in these cases. Tables 6 to 9 display only the performance results for NovRP, LDA/QR and appSVD+LDA. The NovRP method gives the most efficient time (see Tables 7 and 9) whereas its accuracy is by far the lowest. It can be seen that the chosen value of p is widely sufficient for appSVD+LDA to recover nearly 86% of accuracy for Reuters21578 and 95% for TDT2 and the computational time is quite small (see Tables 7 and 9). In the whole results appSVD+LDA significantly outperforms LDA in running time and its accuracy performance let believe in its effectiveness and efficiency compared to other methods.

4.4 Parameter Tunning

There are three essential parameters in the proposed method which are μ, p and k. μ is used for the regularization process of the scatter matrix. k is the dimension of the new feature space where LDA is performed. p is the dimension size of the intermediate subspace where the original features are randomly mapped. A sensitive way of the proposed appSVD+LDA is the choice of p. This parameter should guarantee a minimum distortion between data points after random map. In the final dimensional space each point is represented as a k feature vector that leads to a faster classification process. In our experiments, we chose $k = p$. To illustrate the impact of this parameter, we take various values of p. The accuracy and the training time as a function of p averaged over 20 random splits are plotted on figures 1 and 2. The methods DLDA and LDA/QR do not depend on p contrary to appSVD+LDA and NovRP. In figure 1 (right), as the training time of DLDA is widely high, we have not plotted it. It can be seen that the accuracy of the proposed method is good for small values of p ($p = 80$) and it increases slowly with p

Table 2: Accuracy on COIL20 (Mean ± Std-Dev %).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
10%	**85.88**± **1.78**	73.05 ± 2.55	80.88±1.83	84.37 ± 2.22
30%	**94.14**± **1.02**	79.65 ± 2.59	88.28 ± 2.45	90.43 ±0.97
50%	**95.42** ± **0.89**	81.42 ± 2.04	90.37± 1.84	91.89 ± 1.25

Table 3: Computational time on COIL20 (s).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
10%	2.152	**0.006**	0.009	0.017
30%	2.189	**0.007**	0.030	0.019
50%	2.242	**0.008**	0.050	0.022

Table 4: Accuracy on ORL (Mean ± Std-Dev %).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
2× 40	69.70±3.45	46.52± 3.07	**74.50**± **2.11**	67.41±2.79
4× 40	84.65±2.29	75.75±2.50	85.90±2.85	**89.35**±**1.88**
6× 40	90.12±2.50	84.81±2.93	90.69±1.49	**92.94**±**1.87**

Table 5: Computational time on ORL (s).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
2×40	93.223	**0.041**	0.065	0.225
4× 40	93.340	**0.042**	0.104	0.228
6× 40	93.584	**0.045**	0.145	0.235

Table 6: Accuracy on Reuters21578 (Mean ± Std-Dev %).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
10%	–	48.77±1.55	75.57±0.73	**83.27**±**0.76**
30%	–	48.75±1.99	83.17± 0.62	**86.72**±**0.58**
50%	–	46.81±1.81	86.52±0.44	**86.53**±**0.55**

Table 7: Computational time on Reuters21578 (s).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
10%	–	**0.271**	3.038	1.873
30%	–	**0.282**	10.190	2.286
50%	–	**0.296**	19.101	2.494

Table 8: Accuracy on TDT2 (Mean ± Std-Dev %).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
10%	–	58.92± 1.40	92.45± 0.70	**94.07**± **0.91**
30%	–	62.68±1.28	95.29± 0.23	**95.11**±**0.66**
50%	–	63.33±1.22	**95.74**±**0.28**	95.23±0.77

Table 9: Computational time on TDT2 (s).

TN	DLDA	NovRP	LDA/QR	appSVD+LDA
10%	–	**0.539**	4.066	3.891
30%	–	**0.574**	11.618	4.552
50%	–	**0.582**	18.591	4.627

while the computation time increases quickly with p. For NovRP, the context is opposite, *i.e*, the accuracy increases quickly with p whereas the time increases slowly. For Reuters21578, the best accuracy is obtained with appSVD+LDA for any value of p in the considered range between 80 and 250.

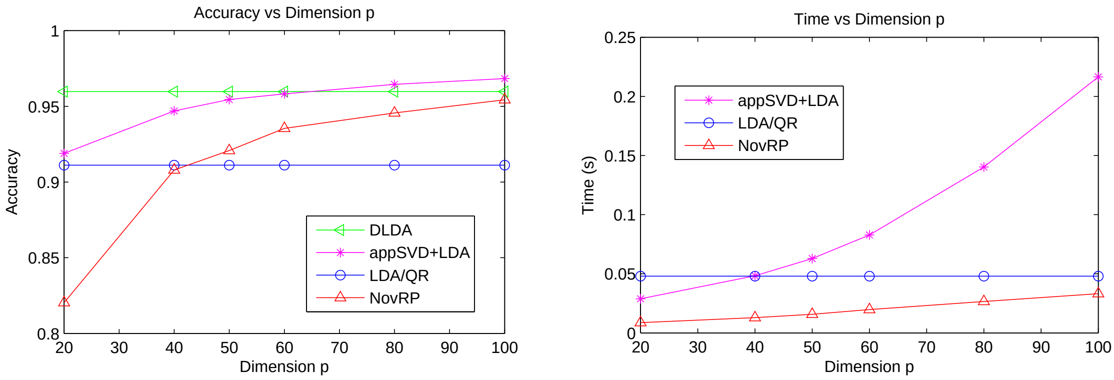

Figure 1: Accuracy vs Dimension p (left) and Time vs Dimension p (right) on COIL20 data set for TN=50%.

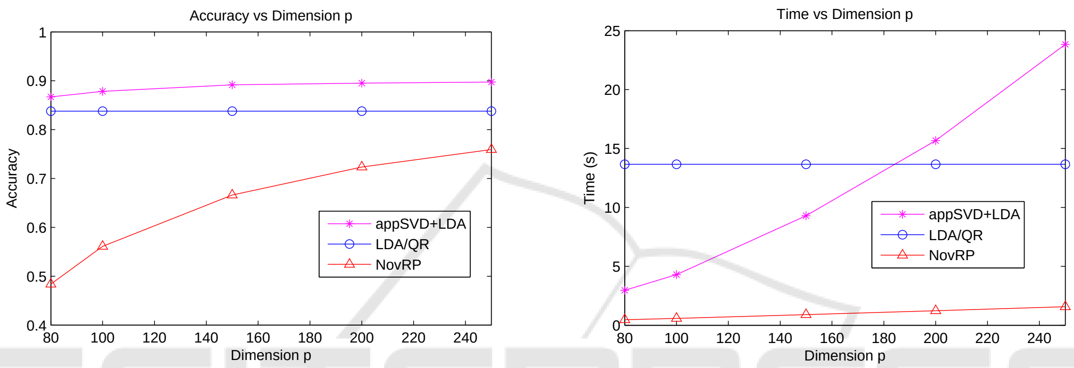

Figure 2: Accuracy vs Dimension p (left) and Time vs Dimension p (right) on Reuters21578 data set for TN=30%.

5 CONCLUSION

This work provides a novel approach to tackle the problem encountered when performing LDA with large scale data sets. It consists of looking for an approximation of the original space in a lower rank. It combines the fast approximate singular value decomposition and LDA. We show by experiments on real world data sets the effectiveness and the efficiency of the proposed method appSVD+LDA. As can be seen, appSVD+LDA outperforms direct LDA in terms of computational time and achieves significant performance in comparison to other state-of-the-art methods. appSVD+LDA allows to classify large scale data by just holding a small features size (k). For example, on Reuters21578 data set where the original feature space is $d = 18933$, it achieves more than 86% accuracy in nearly two seconds whereas it is infeasible to perform direct LDA in this case. The performance results displayed by appSVD+LDA are very encouraging for learning LDA both in small and high dimensional spaces.

ACKNOWLEDGEMENTS

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

REFERENCES

Achlioptas, D. (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687.

Boutsidis, C., Zouzias, A., Mahoney, M. W., and Drineas, P. (2015). Randomized dimensionality reduction for-means clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062.

Cardoso, Â. and Wichert, A. (2012). Iterative random projections for high-dimensional data clustering. *Pattern Recognition Letters*, 33(13):1749–1755.

Chen, L., Man, H., and Nefian, A. V. (2005). Face recognition based on multi-class mapping of fisher scores. *Pattern Recognition*, 38(6):799–811.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Lee, Y. K., Lee, E. R., and Park, B. U. (2012). Principal component analysis in very high-dimensional spaces. *Statistica Sinica*, pages 933–956.

Liu, H. and Chen, W.-S. (2009). A novel random projection model for linear discriminant analysis based face recognition. In *2009 International Conference on Wavelet Analysis and Pattern Recognition*, pages 112–117. IEEE.

Menon, A. K. and Elkan, C. (2011). Fast algorithms for approximating the singular value decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):13.

Moulin, C., Largeron, C., Ducottet, C., Géry, M., and Barat, C. (2014). Fisher linear discriminant analysis for text-image combination in multimedia information retrieval. *Pattern Recognition*, 47(1):260–269.

Welling, M. (2005). Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto*, 3:1–4.

Ye, J. and Li, Q. (2004). Lda/qr: an efficient and effective dimension reduction algorithm and its theoretical foundation. *Pattern recognition*, 37(4):851–854.

Ye, J., Li, Q., Xiong, H., Park, H., Janardan, R., and Kumar, V. (2005). Idr/qr: an incremental dimension reduction algorithm via qr decomposition. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1208–1222.

Yu, H. and Yang, J. (2001). A direct lda algorithm for high-dimensional data with application to face recognition. *Pattern recognition*, 34(10):2067–2070.