Optimal Conjugate Gradient Algorithm for Generalization of Linear Discriminant Analysis Based on L$_1$ Norm

Kanishka Tyagi[1], Nojun Kwak[2] and Michael Manry [1]

[1]*Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Texas, U.S.A.*

[2]*Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea*

Keywords: Linear Discriminant Analysis, L_1 Norm, Dimension Reduction, Conjugate Gradient, Learning Factor.

Abstract: This paper analyzes a linear discriminant subspace technique from an L_1 point of view. We propose an efficient and optimal algorithm that addresses several major issues with prior work based on, not only the L_1 based LDA algorithm but also its L_2 counterpart. This includes algorithm implementation, effect of outliers and optimality of parameters used. The key idea is to use conjugate gradient to optimize the L_1 cost function and to find an learning factor during the update of the weight vector in the subspace. Experimental results on UCI datasets reveal that the present method is a significant improvement over the previous work. Mathematical treatment for the proposed algorithm and calculations for learning factor are the main subject of this paper.

1 INTRODUCTION

Dimensionality reduction and object classification has received considerable attention from the pattern recognition community in the past decades (Duda et al., 2012), (Theodoridis and Koutroumbas, 2009). The goal of dimensionality reduction in classification is to remove less useful elements from the input vectors. Some of the conventional methods employed for dimensionality reduction and object classification are principal component analysis (PCA) (Fukunaga, 1990) (Turk and Pentland,), independent component analysis (ICA) (Bell and sejnowski, 1995) (Cao et al., 2003) (Kwak and Choi, 2003) (Kwon and Lee, 2004) and linear discriminant analysis (LDA) (Fisher, 1936).

The linear discriminant analysis) of Fisher (Fisher, 1936) is a classical supervised subspace analysis technique. By minimizing the within-class scatter and maximizing the between-class scatter, LDA seeks discriminative features. LDA is a classic dimensionality reduction method that preserves as much of the class discriminatory information as possible.

Since the conventional LDA discriminates data in a least square sense (L_2 norm), it is prone to problems common to any methods utilizing L_2 optimization. It is well known that conventional subspace analysis techniques based on L_2 norm minimization is more sensitive to the presence of outliers as its effect is magnified due to a large norm. To alleviate this problem, Koren et., al. (Koren and Carmel, 2008) proposed an optimal weighting approach that assigns small weights to the outliers. However, an optimal weighting parameter is difficult to determine. (Li et al., 2010) proposes a similar rotational invariant L_1 approach but in author's opinion, the method has complex implementation. Another problem of the conventional LDA arises from the assumption that the data distribution is Gaussian. It means that if there are several separate clusters in a class (i.e., multi modal) then the data is not uniquely modeled by a Gaussian distribution. As a result the approaches based on L_2 norm will fail (Fukunaga, 1990). To overcome the difficulty of multi-modal data distribution with LDA, it can be combined with the unsupervised dimension reduction algorithms called locality preserving projection method (LPP) to form a local Fisher discriminant analysis (LFDA), which effectively combines the ideas of LDA and LPP (Sugiyama, 2007).

In this paper, an efficient L_1 norm based-LDA algorithm (Oh and Kwak, 2013), which is motivated by the work in (Kwak, 2008), has been improved by using an iterative algorithm based on a modified optimal conjugate gradient (OCG) to solve an L_1 norm LDA problem (called OCG-LDA hereafter). We also introduce a learning factor scheme for updating the weight vector or projection vectors in the OCG algorithm. The OCG algorithm iteratively converges to an optimal solution similar to iterative recursive least squares (IRLS) algorithm.

Tyagi K., Kwak N. and Manry M..
Optimal Conjugate Gradient Algorithm for Generalization of Linear Discriminant Analysis Based on L1 Norm.
DOI: 10.5220/0004825402070212
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 207-212
ISBN: 978-989-758-018-5

The remainder of this paper is structured as follows. Section 2 reviews the conventional LDA method (based on L_2 norm). In Section 3, we propose the L_1 LDA based on OCG algorithm including its mathematical treatment, theoretical justification, proof for suboptimal learning factor and algorithmic description of the complete methodology. Experimental results are presented in Section 4. Section 5 discusses the results and finally Section 6 and 7 summarizes the proposed work and presents some future exploration respectively.

2 CONVENTIONAL L_2 LDA: A REVIEW

Linear discriminant analysis is a subspace learning approach that leads to supervised dimensionality reduction. It is based on the work of Fisher (Fisher, 1936) and can be considered as an optimal feature generation process (Theodoridis and Koutroumbas, 2009). It tries to find a transformation that maximizes the ratio of the between-class scatter matrix S_B and the within-class scatter matrix S_W (Fukunaga, 1990) which are defined as

$$\begin{aligned} S_B &= \sum_{i=1}^{C} N_i(m_i - m)(m_i - m)^T \\ S_W &= \sum_{i=1}^{C}\sum_{j=1}^{N_i}(x_j^i - m_i)(x_j^i - m_i)^T \end{aligned} \tag{1}$$

where x_i^j is the i-th sample of class j, m_j is the mean of class j, C is the number of classes, N_i is the number of samples in class i and m represents the mean of all samples. This is formulated to find M projection vectors $\{w_i\}_{i=1}^{M}$ that maximize the Fisher's criterion with respect to $W = [w_1, \cdots, w_M]$, as follows:

$$W_{LDA} = \underset{W}{\text{argmax}} \frac{|W^T S_B W|}{|W^T S_W W|}. \tag{2}$$

Equation (2) is the generalized Rayleigh quotient (Duda et al., 2012), which, as known from linear algebra, is maximized if W is chosen such that

$$\begin{aligned} S_B w_i &= \lambda_i S_W w_i \\ \lambda_1 &\geq \lambda_2 \cdots \geq \lambda_m \end{aligned} \tag{3}$$

Then the linear projection of $\{w_i\}_{i=1}^{M}$ can be obtained. Here λ_i is the i-th largest eigenvalue of $S_w^{-1}S_B$ and $w_i \in \Re^{d\times m}$. Viewing LDA as dimension reduction technique, LDA is performed by mapping each vector x in d-dimensional space to a vector y in the M dimensional space ($M < d$) linearly. The linear projection is such that the lower dimensional projection is closer for same class and farther for different classes. However it is well known that if the L_p ($p < 2$) norm is used instead of L_2 norm, outliers are suppressed and the method performs better (Oh and Kwak, 2013). In our present investigation, we consider $p = 1$. The details are presented in next section.

3 PROPOSED PARADIGM: OCG-LDA

It is well known in the literature that algorithms based on the L_1 norm are less sensitive to outliers as compared to their L_2 counterparts (Claerbout and Muir, 1973) (J. A. Scales and Lines, 1988).

3.1 L_1 Norm based LDA

We formulate an L_1-norm maximization problem to design an L_1 based LDA. Motivated from the basic L_1 theory, we solve the following L_1-norm maximization problem (Oh and Kwak, 2013) with the constraint $||w||_2$=1.

$$F_1(w) = \frac{\sum_{i=1}^{C} N_i|w^T(m_i - m)|}{\sum_{i=1}^{C}\sum_{j=1}^{N_i}|w^T(x_j^i - m_i)|} \tag{4}$$

In order to maximize the objective function in (4), we need to consider its non-convexity owing to the absolute value function involved. The singularity due to the non-convexity of $F_1(w)$ makes it difficult to calculate its gradient vector and direction vector. In order to circumvent this problem, we use $sgn(\cdot)$ function as follows,

$$sgn(i) = \begin{cases} 1 & \text{if } i > 0 \\ 0 & \text{if } i = 0 \\ -1 & \text{if } i < 0. \end{cases} \tag{5}$$

modify equation (4) as,

$$F_1(w) = \frac{\sum_{i=1}^{C} N_i sgn(w^T a_i)\cdot(w^T a_i)}{\sum_{i=1}^{C}\sum_{j=1}^{N_i} sgn(w^T b_j^i)\cdot(w^T b_j^i)}. \tag{6}$$

Here,

$$\begin{aligned} a_i &= m_i - m \\ b_j^i &= x_j^i - m_i. \end{aligned} \tag{7}$$

3.2 Mathematical Treatment

We now take the gradient of (4) with respect to w.

$$g(w) = \nabla_w F_1(w) = \frac{(A\cdot B) - (C\cdot D)}{B^2} \tag{8}$$

where A,B,C and D are defined as,

$$A = \sum_{i=1}^{C} N_i \left[sgn(w^T a_i) \cdot a_i\right]$$

$$B = \sum_{i=1}^{C} \sum_{j=1}^{N_i} \left[sgn(w^T b_j^i) \cdot (w^T b_j^i)\right]$$

$$C = \sum_{i=1}^{C} N_i \left[sgn(w^T a_i) \cdot (w^T a_i)\right]$$

$$D = \sum_{i=1}^{C} \sum_{j=1}^{N_i} \left[sgn(w^T b_j^i) \cdot b_j^i\right]$$

The above gradient is well defined when $w^T b_j^i \neq 0$ for all $i = 1, \cdots, C$ and $j = 1, \cdots, N_i$. However, the A and D terms in (8) are not well defined at the singular points where $w^T b_j^i = 0$ because 0^0 is hard to define. To avoid this problem, we add a singularity check step before computing the gradient vector in later development.

3.3 Theoretical Treatment: Why Conjugate Gradient ?

The specific objective of the present work is to improve upon the existing approach of (Oh and Kwak, 2013). In order to do that, we first replace the steepest descent algorithm with a much better conjugate gradient approach. To have a better intuitiveness for the learning factor, we update the weight vector as follows:

$$w(t+1) \leftarrow w(t) + z_1 \cdot v(t) \tag{9}$$

where $v(t)$ is the direction vector and z_1 is the learning factor, as discussed in the next subsection.

The reason as to why steepest descent is slow is due to its straight line search strategy. Since the steepest descent works on the gradient direction and goes along a straight line search, it does not stop till the descent line is parallel to the contour line of the cost function surface. This poses a serious problem and eventually leads to slow minimization. What if we want to stop and change the gradient direction before it becomes parallel? In steepest descent it is difficult to make such a stop and therefore it will follow a zigzag pattern (Haykin, 2009). In general, overshooting and undershooting are inherent problems with the steepest descent approach. One way to overcome this is to use an learning factor so as to update the weight vector with gradient information in a more intelligent way. However what if instead of a line search we do a plane search? This leads to the conjugate gradient (CG) method where an arbitrary combination of two vectors forms a hyperplane. The CG algorithm (Chong and Stanislaw, 2013) solves the equation in exactly n steps where n is the number of unknowns. However for non L_2 functions,(non -quadratic functions)a plane search is hard to deal with.

The conjugate gradient algorithm is related to Krylov subspace (Olavi, 1993) iteration methods. Our motivation for using CG comes from its easy modification for non quadratic problems. From the Hestense-Stiefel formula (to overcome line search problem) and Polak Ribiere formula, we conclude that major techniques exist to go around the two major problems of line search and calculation of Hessian matrix (Chong and Stanislaw, 2013).

Having a strong reason to use CG algorithm, we now present a brief description of the learning factor, weight updation scheme and proposed algorithm in the subsequent section.

3.4 Deriving the Learning Factor

Let F_1^{new} be the estimated new value of F_1. Applying Taylor series on F_1, we get

$$F_1^{new} = F_1^{old} + g_n \cdot \Delta w \tag{10}$$

where gradient $g = dF_1/dw$. Now we have $\Delta w = z_1 \cdot v$. Substituting the value of Δw in (10) we get,

$$F_1^{new} = F_1^{old} + z_1 \cdot ||v||^T g \tag{11}$$

On simplifying, we get,

$$F_1^{new} - F_1^{old} = z_1 \cdot ||v||^T g \tag{12}$$

If the desired value of F_1^{new}= (1+z) $\cdot$ F_1^{old}. Here z is the desired fractional increment set to an initial value as 0.01.

$$F_1^{new} - F_1^{old} = z \cdot F_1^{old} \tag{13}$$

Therefore we have

$$z_1 = \frac{z \cdot F_1^{old}}{||v||^T g} \tag{14}$$

3.5 Deriving the Weight Updation Scheme

We now elaborate the weight updation scheme

1. **Initialization**
   Set $F_1^{old} = \varepsilon$ (small number). Choose z so as to fractionally increase the value of $-F_1$ by 1 % in each iteration and let $z = 0.01$.
2. **Weight updates**
   For each iteration,
   - Calculate the gradient from (8) and z_1 as in section 3.4. Update the weight vector as in (9)
   - Using (4), calculate the value of cost function $-F_1$.

```
If $F_1 > F_1^{old}$
    $F_1^{old} \leftarrow F_1$
    Increment z as z $\leftarrow 1.1 * z$
    Save weight as w $\leftarrow w_{old}$ $(Forwardstep)$
Else
    Decrement z as $z \leftarrow 0.5 \cdot z$
    Decrement $z_1$ as $z_1 \leftarrow 0.5 \cdot z_1$
    Read old weight vector $w_{old}$
    Save weight as w $\leftarrow w_{old}$ (Back step)
    Update the weight vector as in (9)
    Break
End if
end iteration
```

In the above pseudocode, for each weight vector, the iterations are performed quite a number of time until we get an optimal value of weight. We now propose the formal algorithm that incorporates the above expressions.

3.6 Proposed Algorithm

Since every update of the weight vector leads to the maximization of $F_1(w)$, setting an initial $w(0)$ is highly important. In our investigation, we choose the initial vector $w(0)$ to be the solution of the conventional L_2-LDA. Alternatively other techniques can also be tried like re-run the OCG algorithm several times with different initial $w(0)$ and choosing the best one. We now present the proposed OCG-LDA algorithm as follows:

1. **Initialization**
   Set $t = 0$ and $w(0)$ s.t. $||w(0)||_2 = 1$, $v(-1) = 0$, $X_{Den} = 1$. Here v is the direction vector.
2. **Singularity Check**
   If $w(t)^T b_j^i = 0$ for some i and j, $w(t) \leftarrow \frac{w(t)+\delta}{||w(t)+\delta||_2}$. Here, δ is a random vector with a small magnitude.
3. **Gradient Calculation**
   Compute equation (8) to obtain the gradient vector **g**.
4. **Gradient Energy**
   Compute gradient energy $X_{Num} = ||g(w)||_2^2$.
5. **Coefficients for Direction Vector**
   Compute $B_1 = \frac{X_{Num}}{X_{Den}}$.
6. **Direction Vector**
   Compute $v(t) = g(w) + B_1 \cdot v(t-1)$
7. **learning Factor and Weight Updation**
   Calculate F_1(w) in (4) and compute z_{opt}. Update weight vector as $w(t+1) \leftarrow w(t) + z_{opt} \cdot v(t)$. (See Section 3.5 for details)
8. **Update for Next Iteration**
   Replace X_{Den} with the value of X_{Num}.
   Set $t \leftarrow t+1$.
9. **Convergence Check**
   If $||w(t) - w(t-1)|| \geq \varepsilon$, then go to Step 2.
   Else $w^* = w(t)$ and stop iteration.

Figure 1 explains the learning factor $z_1$and weight updation calculation. We take the forward step by incrementing z_1 until the value of $-F_1$ is lower than the previous iteration. In case the value of $-F_1$ is greater for the current iteration, we take a back step to decrement z_1 by half and thereby backtrack to reading the old weight vector. It should be noted here that the learning factor z obtained by using the above procedure is optimal in a sense that its better estimate than a heuristic approach. In Figure 1 (1), (2) are forward step and (3), (4) are back step.

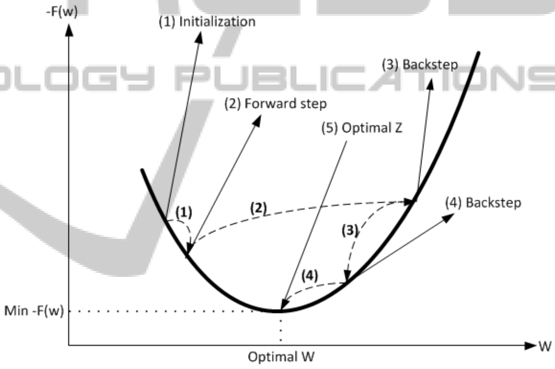

Figure 1: Calculating learning factor z_1.

4 EXPERIMENT AND RESULTS

We now apply OCG-LDA to different datasets from UCI machine learning repositories and compare their performance with other algorithms. We have compared the proposed methodology with two subspace learning method (L_2-LDA, SD-LDA) and two L_1 based least square methods. The L_2-LDA we used, is solved by eigenvalue decomposition. The IRLS $(Ji, 2006)$ and L_1 regularized least square (L_1-RLS) (Boyd,2007) have been chosen primarily for there less time complexity. Table 1 shows the numbers of variables, classes, and instances of each data set which is used in this paper.

In Table 2, we present the classification performances for OCG-LDA and it's comparison with other iterative algorithms. We also make the observation that using a direction vector information rather than the gradient vector leads to a faster performance. Not

Table 1: UCI data set.

Data Set	No of variables	No of instances
Australian	14	690
Heart Disease	13	297
Bupa	6	345
Pima	8	768
Sonar	60	208
Balance	4	625
Waveform	21	4999

Table 2: Classification (%) for UCI data set.

Dataset ↓	L_2 LDA	SD- LDA	L_1 IRLS	L_1 RLS	OCG-LDA
Australian	76.6790	81.8834	80.0032	81.3002	**82.0324**
Heart Disease	75.7580	80.8506	79.2130	75.7721	**81.3059**
Bupa	54.2182	65.8655	62.0581	62.8991	**67.8591**
Pima	65.8751	71.0919	**72.6371**	67.8161	72.4526
Sonar	65.3810	76.8333	**78.4051**	72.0712	78.1261
Balance	88.9683	90.0794	87.1943	87.5292	**92.0957**
Waveform	52.9113	56.0905	51.3703	51.7708	**58.8654**

only that, varying learning factor rather than a fixed value leads to a better results than L_1 LDA with steepest descent.

5 DISCUSSION

The result obtained in previous section, generates an interesting set of findings. We observe from Table 2 that OCG-LDA is comparable in classification rate to other learning algorithms. Except the sonar dataset, the classification rate for OCG-LDA is better either marginally or considerably than other subspace or least square algorithms. The average time (in sec) is also comparable to most of the other L_1 based algorithms. Lower time complexity of L_1 algorithms is a clear advantage over its L_2 counterpart. The average number of iterations also bolster our argument about the clear advantage of using OCG-LDA over other L_1 based algorithms and clearly the L_2 based SD-LDA algorithm.

To understand the importance of the findings we see that the weight vector in the updating scheme (17) depends on the input feature vector. Now since we have optimal weight updating and conjugate gradient scheme for optimizing the weight vectors, the number of input feature vector clearly affects the overall optimization scheme. This explains poor performance of OCG-LDA for sonar dataset. As the cost function minimization involves absolute value and not a square error, the time taken by L_1 based approach is considerably lower than it's L_2 method (L_2-LDA).

Table 3: Average time (s) and average number of iterations.

Dataset ↓	L_2 LDA	SD- LDA	L_1 IRLS	L_1 RLS	OCG-LDA
Australian	0.5507 (-)	1.6717 (100)	0.7164 (3)	0.7670 (4)	**0.6121** (3)
Heart Disease	0.1240 (-)	0.1268 (15)	0.1324 (5)	0.1244 (3)	**0.1242** (4)
Bupa	0.1575 (-)	0.4132 (47)	0.3472 (9)	0.3120 (3)	**0.2742** (5)
Pima	0.6621 (-)	1.1014 (58)	0.8274 (4)	0.9524 (3)	**0.7825** (4)
Sonar	0.0783 (-)	0.5850 (100)	**0.4765** (10)	0.5101 (4)	0.4956 (5)
Balance	0.4493 (-)	1.0717 (71)	0.7582 (2)	0.6823 (1)	**0.6248** (1)
Waveform	26.503 (-)	4.3259 (11)	2.4462 (2)	**1.3170** (5)	1.5295 (2)

An important implication of using an learning factor and a second order optimizing scheme is that the average time and the average number of iterations are considerably better or comparable to other L_1 and L_2 based algorithms. Updating the weight vector in L_1 subspace and guiding it with a learning factor serves two-fold purpose. Firstly, the problem of overshooting or zigzagging of the weight vector is eliminated as is typical in any gradient based iterative approach. Secondly, the use of direction vector facilitates in a much faster and efficient algorithm. What's interesting to note here is that the combination of efficient scheme for obtaining learning factor and optimization using CG algorithms leads to an algorithm that is much better than SD-LDA. Not only that, OCG-LDA is comparable to other complicated algorithms based on L_1 regularization or iterative lease square scheme.

In order to relate the finding to those of similar studies, our motivation came from improving the SD-LDA by developing a better and efficient subspace algorithm. The results in Table 2 and 3 clearly implies that we have achieved our goal. The questions raised in (Oh and Kwak, 2013) about the use of an optimal weight updation scheme, in (Duda et al., 2012) about overcoming the effect of outliers and comparison to the regularized least square and iterative least square algorithms serves as motivation for our study. Another aspect was to see the Gaussian distribution effect on data that is prominent in L-2 norms.

6 CONCLUSIONS

The proposed OCG-LDA algorithm uses a conjugate gradient optimization scheme to improve the existing SD-LDA subspace algorithm. The conjugate gradient is chosen due to its advantages over first degree optimization scheme like steepest descent and easy implementation. We test the OCG-LDA algorithm for various UCI datasets to demonstrate its classification performance, average time and average iterations. OCG-LDA clearly outperforms the L-2 LDA and SD-LDA but has comparable performance with the L-1 version of least square algorithms. However the proposed methodology is simple and easy to implement and is a good alternative to other algorithms in building a robust model for classification. As from No Free Lunch theorem, no single classification algorithm can outperform any other algorithm when performance is analyzed over many classification dataset. In conclusion, OCG-LDA can be used as a basic classifier unit in a multi stage classification scheme.

7 FUTURE WORK

The OCG-LDA methodology is an evident advancement in the L_1 family of LDA subspace algorithms. As a part of future direction, a multiple optimal learning factor scheme based on the Gaussian Newton approximation (Malalur and Manry, 2010) can be investigated. Recently, the author (Cai et al., 2011) have proposed an efficient partial Hessian calculation that does not involves inversion and is successfully applied on Radial basis function neural networks. Therefore a study can be conducted to foray into the second order algorithms using regularization parameter.

REFERENCES

Bell, A. and sejnowski, T. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7.

Cai, X., K.Tyagi, and Manry, M. (2011). An optimal construction and training of second order rbf network for approximation and illumination invariant image segmentation.

Cao, L., Chua, K., Chong, W., Lee, H., and Gu, Q. (2003). A comparison of pca, kpca and ica for dimensionality reduction in support vector machine. *Neurocomputing*, 55.

Chong, E. and Stanislaw, Z. (2013). *An introduction to optimization*. Wiley, USA, 3rd edition.

Claerbout, J. F. and Muir, F. (1973). Robust modeling with erratic data.

Duda, R., Hart, P., and Stork, D. (2012). *Pattern Classification*. Wiley-interscience, USA, 2nd edition.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2).

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, USA, 2nd edition.

Haykin, S. (2009). *Neural networks and learning machines*. Prentice Hall, USA, 3rd edition.

J. A. Scales, A. Gersztenkorn, S. T. and Lines, L. R. (1988). Robust optimization methods in geophysical inverse theory.

Ji, J. (2006). Cgg method for robust inversion and its application to velocity-stack inversion.

Koren, Y. and Carmel, L. (2008). Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):459–470.

Kwak, N. (2008). Principal component analysis based on l-1 norm maximization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(9):1672–1680.

Kwak, N. and Choi, C.-H. (2003). Feature extraction based on ica for binary classification problems. *IEEE Trans. on Knowledge and Data Engineering*, 15(6):1374–1388.

Kwon, O.-W. and Lee, T.-W. (2004). Phoneme recognition using ica-based feature extraction and transformation. *Signal Processing*, 84(6).

Li, X., Hu, W., Wang, H., and Zhang, Z. (2010). Linear discriminant analysis using rotational invariant l1 norm. *Neurocomputing*, 73.

Malalur, S. S. and Manry, M. T. (2010). Multiple optimal learning factors for feed-forward networks.

Oh, J. and Kwak, N. (2013). Generalization of linear discriminant analysis using lp-norm. *Pattern Recognition Letters*, 34(6):679–685.

Olavi, N. (1993). *Convergence of iterations for linear equations*. Birkhauser, USA, 3rd edition.

Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061.

Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition*. Academic Press, USA, 4th edition.

Turk, M. and Pentland, A. Face recognition using eigenfaces.