Linear classifier design under heteroscedasticity in Linear Discriminant Analysis

Gyamfi, KS, Brusey, J, Hunt, A & Gaura, E

Published PDF deposited in Coventry University’s Repository

Original citation:

http://dx.doi.org/10.1016/j.eswa.2017.02.039

DOI 10.1016/j.eswa.2017.02.039
ISSN 0957-4174

Publisher: Elsevier

Under a CC-BY-NC-ND 4.0 licence (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.
Linear classifier design under heteroscedasticity in Linear Discriminant Analysis

Kojo Sarfo Gyamfi*, James Brusey, Andrew Hunt, Elena Gaura

Faculty of Engineering and Computing, Coventry University, Coventry, CV1 5FB, United Kingdom

ARTICLE INFO

Article history:
Received 12 November 2016
Revised 23 February 2017
Accepted 24 February 2017
Available online 24 February 2017

Keywords:
LDA
Heteroscedasticity
Bayes error
Linear classifier

ABSTRACT

Under normality and homoscedasticity assumptions, Linear Discriminant Analysis (LDA) is known to be optimal in terms of minimising the Bayes error for binary classification. In the heteroscedastic case, LDA is not guaranteed to minimise this error. Assuming heteroscedasticity, we derive a linear classifier, the Gaussian Linear Discriminant (GLD), that directly minimises the Bayes error for binary classification. In addition, we also propose a local neighbourhood search (LNS) algorithm to obtain a more robust classifier if the data is known to have a non-normal distribution. We evaluate the proposed classifiers on two artificial and ten real-world datasets that cut across a wide range of application areas including handwriting recognition, medical diagnosis and remote sensing, and then compare our algorithm against existing LDA approaches and other linear classifiers. The GLD is shown to outperform the original LDA procedure in terms of the classification accuracy under heteroscedasticity. While it compares favourably with other existing heteroscedastic LDA approaches, the GLD requires as much as 60 times lower training time on some datasets. Our comparison with the support vector machine (SVM) also shows that, the GLD, together with the LNS, requires as much as 150 times lower training time to achieve an equivalent classification accuracy on some of the datasets. Thus, our algorithms can provide a cheap and reliable option for classification in a lot of expert systems.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In many applications one encounters the need to classify a given object under one of a number of distinct groups or classes based on a set of features known as the feature vector. A typical example is the task of classifying a machine part under one of a number of health states. Other applications that involve classification include face detection, object recognition, medical diagnosis, credit card fraud prediction and machine fault diagnosis.

A common treatment of such classification problems is to model the conditional density functions of the feature vector (Ng & Jordan, 2002). Then, the most likely class to which a feature vector belongs can be chosen as the class that maximises the a posteriori probability of the feature vector. This is known as the maximum a posteriori (MAP) decision rule.

Let K be the number of classes, C_k be the kth class, x be a feature vector and D_k be training samples belonging to the kth class (k ∈ {1, 2, …, K}). The MAP decision rule for the classification task is then to choose the most likely class of x, C^*(x) given as:

\[ C^*(x) = \arg \max_{k \in \{1, 2, \ldots, K\}} p(C_k|x), \quad k \in \{1, 2, \ldots, K\} \]

(1)

We assume for the moment that there are only K = 2 classes, i.e. binary classification (we consider multi-class classification in a later section). Then, using Bayes’ rule, the two posterior probabilities can be expressed as:

\[ p(C_k|x) = \frac{p(x|C_k) \times p(C_k)}{p(x)} \quad k \in \{1, 2\} \]

(2)

It is often the case that the prior probabilities p(C_1) and p(C_2) are known, or else they may be estimable from the relative frequencies of D_1 and D_2 in D where D = D_1 ꚃ D_2. Let these priors be given by π_1 and π_2 respectively for class C_1 and C_2. Then, the likelihood ratio defined as:

\[ \lambda(x) = \frac{p(x|C_1)}{p(x|C_2)} \]

(3)

is compared against a threshold defined as τ = π_2/π_1 so that one decides on class C_1 if \( \lambda(x) \geq \tau \) and class C_2 otherwise.

Linear Discriminant Analysis (LDA) proceeds from here with two basic assumptions (Izenman, 2009, Chapter 8):

1. The conditional probabilities p(x|C_1) and p(x|C_2) have multivariate normal distributions.
2. The two classes have equal covariance matrices, an assumption known as homoscedasticity.

Let $\mathbf{x}_1$, $\Sigma_1$ be the mean and covariance matrix of $\mathcal{D}_1$ and $\mathbf{x}_2$, $\Sigma_2$ be the mean and covariance of $\mathcal{D}_2$ respectively. Then, for $k \in \{1, 2\}$,

$$p(x|C_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp \left[ -\frac{1}{2} (x - \bar{x}_k)^T \Sigma_k^{-1} (x - \bar{x}_k) \right]$$

where $d$ is the dimensionality of $\mathcal{X}$, which is the feature space of $x$. Given the above definitions of the conditional probabilities, one may obtain a log-likelihood ratio given as:

$$\ln \lambda(x) = \frac{1}{2} \ln \det \Sigma_2 - \frac{1}{2} \ln \det \Sigma_1 + \frac{1}{2} \left( (x - \bar{x}_2)^T \Sigma_2^{-1} (x - \bar{x}_2) - (x - \bar{x}_1)^T \Sigma_1^{-1} (x - \bar{x}_1) \right)$$

(5)

which is then compared against $\ln \tau$ so that $C_1$ is chosen if $\ln \lambda(x) \geq \ln \tau$, and $C_2$ otherwise. Thus, the decision rule for classifying a vector $x$ under class $C_1$ can be rewritten as:

$$(x - \bar{x}_2)^T \Sigma_2^{-1} (x - \bar{x}_2) - (x - \bar{x}_1)^T \Sigma_1^{-1} (x - \bar{x}_1) \geq \ln \frac{\tau^2 \det \Sigma_1}{\det \Sigma_2}$$

(6)

In general, this result is a quadratic discriminant. However, a linear classifier is often desired for the following reasons:

1. A linear classifier is robust against noise since it tends not to overfit (Mika, Ratsch, Weston, Scholkopf, & Mullers, 1999).
2. A linear classifier has relatively shorter training and testing times (Yuan, Ho, & Lim, 2012).
3. Many linear classifiers allow for a transformation of the original feature space into a higher dimensional feature space using the kernel trick for better classification in the case of a non-linear decision boundary (Bishop, 2006, Chapter 6).

By calling on the assumption of homoscedasticity, i.e. $\Sigma_1 = \Sigma_2 = \Sigma$, the original quadratic discriminant given by (6) for classifying a given vector $x$ decomposes into the following linear decision rule:

$$x^T \Sigma^{-1} (\bar{x}_1 - \bar{x}_2) \leq \frac{1}{2} \ln \tau + \frac{1}{2} (\bar{x}_1^T \Sigma^{-1} \bar{x}_1 - \bar{x}_2^T \Sigma^{-1} \bar{x}_2)$$

(7)

Here, $\Sigma^{-1} (\bar{x}_1 - \bar{x}_2)$ is a vector of weights denoted by $w$ and $\ln \tau + \frac{1}{2} (\bar{x}_1^T \Sigma^{-1} \bar{x}_1 - \bar{x}_2^T \Sigma^{-1} \bar{x}_2)$ is a threshold denoted by $w_0$. This linear classifier is also known as Fishers Linear Discriminant. If only the weight vector $w$ is required for dimensionality reduction, $w$ may be obtained by maximising Fishers criterion (Fisher, 1936), given by:

$$S = \frac{w^T (\bar{x}_1 - \bar{x}_2) (\bar{x}_1 - \bar{x}_2)^T w}{w^T \Sigma w}$$

(8)

where $\Sigma = n_1 \Sigma_1 + n_2 \Sigma_2$ and $n_1, n_2$ are the cardinalities of $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively.

LDA is the optimal Bayes’ classifier for binary classification if the normality and homoscedasticity assumptions hold (Hamsici & Martinez, 2008) (Izenman, 2009, Chapter 8). It demands only the computation of the dot product between $w$ and $x$, which is a relatively computationally inexpensive operation.

As a supervised learning algorithm, LDA is performed either for dimensionality reduction (usually followed by classification) (Barber, 2012, Chapter 16; Buturovic, 1994; Duin & Loog, 2004; Sengur, 2008), or directly for the purpose of statistical classification (Fukunaga, 2013, Chapter 4; Izenman, 2009; Mika et al., 1999). LDA has been applied to several problems such as medical diagnosis e.g. Coomans, Jonckheer, Massart, Broeckaert, and Bloks (1978); Polat, Günes, and Arslan (2008); Sengur (2008); Sharma and Palival (2008), face and object recognition e.g. Chen, Liao, Ko, Lin, and Yu (2000); Liu, Chen, Tan, and Zhang (2007); Song, Zhang, Wang, Liu, and Tao (2007); Yu and Yang (2001) and credit card fraud prediction e.g. Mahmoudi and Duman (2015). The widespread use of LDA in these areas is not because the datasets necessarily satisfy the normality and homoscedasticity assumptions, but mainly due to the robustness of LDA against noise, being a linear model (Mika et al., 1999). Since the linear Support Vector Machine (SVM) can be quite expensive to train, especially for large values of $K$ or $n (n = n_1 + n_2)$, LDA is often relied upon (Harilharan, Malik, & Ramanan, 2012).

Yet, practical implementation of LDA is not without problems. Of note is the small sample size (SSS) problem that LDA faces with high-dimensional data and much smaller training data (Lu, Plataniotis, & Venetsanopoulos, 2003; Sharma & Palival, 2015). When $d \gg n$, the scatter matrix $\Sigma_x$ is not invertible, as it is not full-rank. Since the decision rule as given by (7) requires the computation of the inverse of $\Sigma_x$, the singularity of $\Sigma_x$ makes the solution infeasible. In works by, for example, Liu et al. (2007); Palival and Sharma (2012), this problem is overcome by taking the Moore–Penrose pseudo-inverse of the scatter matrix, rather than the ordinary matrix inverse. Sharma and Palival (2008) use a gradient descent approach where one starts from an initial solution of $w$ and moves in the negative direction of the gradient of Fisher’s criterion (8). This method avoids the computation of an inverse altogether. Another approach to solving the SSS problem involves adding a scalar multiple of the identity matrix to the scatter matrix to make the resulting matrix non-singular, a method known as regularised discriminant analysis (Friedman, 1989; Lu et al., 2003).

However, for a given dataset that does not satisfy the homoscedasticity or normality assumption, one would expect that modifications to the original LDA procedure accounting for these violations would yield an improved performance. One such modification, in the case of a non-normal distribution, is the mixture discriminant analysis (Hastie & Tibshirani, 1996; Ju, Kolaczyk, & Gopal, 2003; McLachlan, 2004) in which a non-normal distribution is modelled as a mixture of Gaussians. However, the parameters of the mixture components or even the number of mixture components, are usually not known a priori. Other non-parametric approaches to LDA that remove the normality assumption involve using local neighbourhood structures (Cai, He, Zhou, Han, & Bao, 2007; Fukunaga & Mantock, 1983; Li, Lin, & Tang, 2009) to construct a similarity matrix instead of the scatter matrix $\Sigma_x$ used in LDA. However, these approaches aim at linear dimensionality reduction, rather than linear classification. Another modification, in the case of a non-linear decision boundary between $\mathcal{D}_1$ and $\mathcal{D}_2$, is the Kernel Fisher Discriminant (KFD) (Mika et al., 1999; Polat et al., 2008; Zhao, Sun, Yu, Liu, & Ye, 2009). KFD maps the original feature space $\mathcal{X}$ into some other space $\mathcal{Y}$ (usually higher dimensional) via the kernel trick (Mika et al., 1999). While the main utility of the kernel is to guarantee linear separability in the transformed space, the kernel may also be employed to transform non-normal data into one that is near-normal.

Our proposed method differs from the above approaches in that we primarily consider violation of the homoscedasticity assumption, and do not address the SSS problem. We seek to provide a linear approximation to the quadratic boundary given by (6) under heteroscedasticity without any kernel transformation; we note that several heteroscedastic LDA approaches have been proposed to this effect. Nevertheless, for reasons which we highlight in the next section, our contributions in this paper are stated explicitly as
follows:
1. We propose a novel linear classifier, which we term the Gaussian Linear Discriminant (GLD), that directly minimises the Bayes error under heteroscedasticity via an efficient optimisation procedure. This is presented in Section 3.
2. We propose a local neighbourhood search method to provide a more robust classifier if the data has a non-normal distribution (Section 4).

2. Related work

Under the heteroscedasticity assumption, many LDA approaches have been proposed among which we mention (Fukunaga, 2013, Chapter 4; Decell & Mayekar, 1977; Decell Jr & Marani, 1976; Duin & Loog, 2004; Loog & Duin, 2002; Malina, 1981; McLachlan, 2004; Zhang & Liu, 2008). As it is known that Fisher’s criterion (whose maximisation is equivalent to the LDA derivation described in the Introduction section) only takes into account the difference in the projected class means, existing heteroscedastic LDA approaches tend to obtain a generalisation on Fisher’s criterion. In the work of Loog and Duin (2002), for instance, a directed distance matrix (DDM) known as the Chernoff distance, which takes into account the difference in covariance matrices between the two classes as well as the projected class means, is maximised instead of Fisher’s criterion (8). The same idea employing the Chernoff criterion is used by Duin and Loog (2004). A wider class of Bregman divergences including the Bhattacharyya distance (Decell Jr & Marani, 1976) and the Kullback-Leibler divergence (Decell & Mayekar, 1977) have also been used for heteroscedastic LDA, as Fisher’s criterion can be considered a special case of these measures when the covariance matrices of the classes are equal.

However, most of these approaches aim at linear dimensionality reduction, which involves finding a linear transformation that transforms the original data into one of reduced dimensionality, while at the same time maximising the discriminatory information between the classes. Our focus with this paper, however, is not on dimensionality reduction, but on obtaining a Bayes optimal linear classifier for binary classification assuming that the covariance matrices are not equal. As far as we know, the closest work to ours in this regard are the works by Anderson and Bahadur (1962); Fukunaga (2013); Marks and Dunn (1974); Peterson and Mattson (1966).

Obtaining the Bayes optimal linear classifier involves minimising the probability of misclassification $p_e$ as given by:

$$p_e = \tau_1 \Pr(y < w_0 | C_1) + \tau_2 \Pr(y \geq w_0 | C_2)$$

where $y = \mathbf{w}^T \mathbf{x}$. Unfortunately, there is no closed-form solution to the minimisation of (9) (Anderson & Bahadur, 1962). Thus, an iterative procedure is inevitable in order to obtain the Bayes optimal linear classifier.

In the work of Marks and Dunn (1974), for example, the iterative procedure described is to solve for $\mathbf{w}$ and $w_0$ as given by

$$\mathbf{w} = \left[ s \Sigma_1 + (1 - s) \Sigma_2 \right]^{-1} (\bar{x}_1 - \bar{x}_2)$$

$$w_0 = \mu_1 - s \sigma_1^2 = \mu_2 + s \sigma_2^2$$

(10)

by obtaining the optimal values of $s_1$ and $s_2$ via systematic trial and error. We denote this heteroscedastic LDA procedure by R-HLD-2, for the reason that the two parameters $s_1$ and $s_2$ are chosen at random.

Anderson and Bahadur (1962) make the observation that if the weight vector $\mathbf{w}$ and the threshold $w_0$ are both multiplied by the same positive scalar, the decision boundary remains unchanged. Therefore, by multiplying (10) through by the scalar $s_1 + s_2$, $\mathbf{w}$ and $w_0$ can be put in the form of:

$$\mathbf{w} = \left[ s \Sigma_2 + (1 - s) \Sigma_1 \right]^{-1} (\bar{x}_1 - \bar{x}_2)$$

$$w_0 = \mu_1 - (1 - s) \sigma_1^2 = \mu_2 + s \sigma_2^2$$

(11)

Still, the optimal value of $s$ has to be chosen by systematic trial and error. We denote this heteroscedastic LDA approach by R-HLD-1, for the reason that only one parameter $s$ is chosen at random. As we show in the next section, $s$ is unbounded. Therefore, the difficulty faced by this approach is that $s$ has to be chosen from the interval $(-\infty, \infty)$, so that the probability of finding the optimal $s$ for a given dataset is low, without extensive trial and error to limit the choice of $s$ to some finite interval $[a, b]$.

To avoid the unguided trial and error procedure in Anderson and Bahadur (1962); Marks and Dunn (1974), (Peterson & Mattson, 1966) and Fukunaga (2013, Chapter 4) propose a theoretical approach described below:

1. Change $s$ from 0 to 1 with small step increments $\Delta s$.
2. Evaluate $\mathbf{w}$ as given by:

$$\mathbf{w} = \left[ s \Sigma_1 + (1 - s) \Sigma_2 \right]^{-1} (\bar{x}_1 - \bar{x}_2)$$

(12)

3. Evaluate $w_0$ as given by:

$$w_0 = \frac{s \mu_2 \sigma_2^2 + (1 - s) \mu_1 \sigma_1^2}{s \sigma_1^2 + (1 - s) \sigma_2^2}$$

(13)

4. Compute the probability of misclassification $p_e$.
5. Choose $\mathbf{w}$ and $w_0$ that minimise $p_e$.

We refer to this procedure as C-HLD, for the reason that the optimal $s$ is constrained in the interval $[0, 1]$.

However, we highlight two main problems with the above C-HLD procedure:

1. There is no obvious choice of the step rate $\Delta s$. Too small a value of $\Delta s$ will demand too many matrix inversions in Step 2, as there will be too many $s$ values. On the other hand, if $\Delta s$ is too large, the optimal $s$ may not be refined enough, and the $\mathbf{w}$ obtained may not be optimal. Specifically, the change in $\mathbf{w}$ that results from a small change in $s$ is given as:

$$d\mathbf{w} = (s \Sigma_2 + (1 - s) \Sigma_1)^{-1} (\Sigma_1 - \Sigma_2)$$

$$\times (s \bar{x}_1 - \bar{x}_2)ds$$

(14)

which can affect the classification accuracy.

2. The solution obtained this way is only locally optimal as $s$ is bounded in the interval $[0, 1]$. As we show in the next section, $s$ is actually unbounded. When there is a class imbalance (Xue & Titterington, 2008), the optimal $s$ may be found outside the interval $[0, 1]$, which can lead to poor classification accuracy.

Our proposed algorithm, which is described in the next section, unlike the trial and error approach by Anderson and Bahadur (1962); Marks and Dunn (1974), has a principled optimisation procedure, and unlike Fukunaga (2013); Peterson and Mattson (1966) do not encounter the problem of choosing an inappropriate $\Delta s$, nor restricts $s$ to the interval $[0, 1]$. Consequently, our proposed algorithm achieves a far lower training time than the C-HLD, R-HLD-1 and R-HLD-2, for roughly the same classification accuracy.

3. Gaussian linear discriminant

Let $\mathbf{w} \in \mathbb{R}^d$ be a vector of weights, and $w_0 \in \mathbb{R}$, a threshold such that:

$$C^* (\mathbf{x}) = \begin{cases} C_1 & \text{if } y = \mathbf{w}^T \mathbf{x} \geq w_0 \\ C_2 & \text{if } y = \mathbf{w}^T \mathbf{x} < w_0 \end{cases}$$

(15)

Since $\mathbf{x}$ is assumed to have a multivariate normal distribution in classes $C_1$ and $C_2$, $y$ has a mean of $\mu_1$ and a variance of $\sigma_1^2$ for
class \( c_1 \) and a mean of \( \mu_2 \) and a variance of \( \sigma_2^2 \) for class \( c_2 \) given as:

\[
\mu_1 = w^T \hat{x}_1 \quad \mu_2 = w^T \hat{x}_2 \quad \sigma_1^2 = w^T \Sigma_1 w \quad \sigma_2^2 = w^T \Sigma_2 w
\]  

With reference to the Bayes error of (9), the individual misclassification probabilities can be expressed as:

\[
p(y < w_0 | c_1) = \int_{-\infty}^{w_0} \frac{1}{\sqrt{2\pi}\sigma_1} \exp \left[ -\frac{(\xi - \mu_1)^2}{2\sigma_1^2} \right] d\xi = 1 - Q\left( \frac{w_0 - \mu_1}{\sigma_1} \right)
\]  

and

\[
p(y \geq w_0 | c_2) = \int_{w_0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp \left[ -\frac{(\xi - \mu_2)^2}{2\sigma_2^2} \right] d\xi = Q\left( \frac{w_0 - \mu_2}{\sigma_2} \right)
\]  

where \( Q(\cdot) \) is the Q-function. Therefore, the Bayes error to be minimized may be rewritten as:

\[
p_e = \pi_1 \left[ 1 - Q(z_1) \right] + \pi_2 \left[ Q(z_2) \right]
\]  

where

\[
z_1 = \frac{w_0 - \mu_1}{\sigma_1} \quad \text{and} \quad z_2 = \frac{w_0 - \mu_2}{\sigma_2}
\]  

Our aim is to find a local minimum of \( p_e \). A necessary condition is for the gradient of \( p_e \) to be zero, i.e.,

\[
\nabla p_e(w, w_0) = \left[ \frac{\partial p_e}{\partial w}, \frac{\partial p_e}{\partial w_0} \right] = 0
\]  

From (9), it can be shown that:

\[
\frac{\partial p_e}{\partial w} = \pi_1 \left( \frac{1}{\sqrt{2\pi}} e^{-z_1^2/2} \frac{\partial z_1}{\partial w} \right) - \pi_2 \left( \frac{1}{\sqrt{2\pi}} e^{-z_2^2/2} \frac{\partial z_2}{\partial w} \right)
\]  

From (20), however, we obtain the following:

\[
\frac{\partial z_1}{\partial w} = -\frac{\sigma_1}{\sigma_1^2} \hat{x}_1 - \frac{\sigma_2}{\sigma_1^2} \Sigma_1 w \quad \text{and} \quad \frac{\partial z_2}{\partial w} = -\frac{\sigma_2}{\sigma_2^2} \hat{x}_2 - \frac{\sigma_2}{\sigma_2^2} \Sigma_2 w
\]  

Therefore,

\[
\frac{\partial p_e}{\partial w} = \frac{1}{\sqrt{2\pi}} \left[ -\frac{\pi_1 e^{-z_1^2/2}}{\sigma_1^2} \left( \frac{\sigma_1}{\sigma_1^2} \hat{x}_1 + \frac{\pi_1 z_1}{\sigma_1^2} \Sigma_1 w \right) \right]
\]  

\[
+ \frac{\pi_2 e^{-z_2^2/2}}{\sigma_2^2} \left( \frac{\sigma_2}{\sigma_2^2} \hat{x}_2 + \frac{\pi_2 z_2}{\sigma_2^2} \Sigma_2 w \right)
\]  

It can similarly be shown from (9) that:

\[
\frac{\partial p_e}{\partial w_0} = \pi_1 \left( \frac{1}{\sqrt{2\pi}} e^{-z_1^2/2} \frac{\partial z_1}{\partial w_0} \right) - \pi_2 \left( \frac{1}{\sqrt{2\pi}} e^{-z_2^2/2} \frac{\partial z_2}{\partial w_0} \right)
\]  

Again, from (20),

\[
\frac{\partial z_1}{\partial w_0} = \frac{1}{\sigma_1} \quad \text{and} \quad \frac{\partial z_2}{\partial w_0} = \frac{1}{\sigma_2}
\]  

Therefore,

\[
\frac{\partial p_e}{\partial w_0} = \pi_1 \left( \frac{1}{\sqrt{2\pi}} e^{-z_1^2/2} \right) - \pi_2 \left( \frac{1}{\sqrt{2\pi}} e^{-z_2^2/2} \right)
\]  

Substituting (29) into (28) yields:

\[
\left( \frac{z_2}{\sigma_2} \Sigma_2 - \frac{z_1}{\sigma_1} \Sigma_1 \right) w = \left( \hat{x}_1 - \hat{x}_2 \right)
\]  

Then the vector \( w \) can be given by:

\[
w = \left( \frac{z_2}{\sigma_2} \Sigma_2 - \frac{z_1}{\sigma_1} \Sigma_1 \right)^{-1} \left( \hat{x}_1 - \hat{x}_2 \right)
\]  

It will be noted however that (31) is still in terms of \( w_0 \), so that an explicit representation of \( w_0 \) in terms of \( w \) is needed from (29) to substitute in \( z_1 \) and \( z_2 \) in (31). This is where our approach most significantly differs from Fukunaga (2013). Solving for \( w_0 \) from (29) results in the following quadratic:

\[
\frac{z_2^2}{\sigma_2^2} - \frac{z_1^2}{\sigma_1^2} - \ln \left( \frac{\tau \sigma_1}{\sigma_2} \right) = 0
\]  

which can be simplified to:

\[
\left( \frac{w_0 - \mu_1}{\sigma_2} \right)^2 - \left( \frac{w_0 - \mu_1}{\sigma_1} \right)^2 - 2 \ln \left( \frac{\tau \sigma_1}{\sigma_2} \right) = 0
\]  

where \( \tau \) is given as before as \( \tau = \pi_2/\pi_1 \). If \( \tau \) is defined and not equal to zero, and \( \sigma_1^2 \neq \sigma_2^2 \), (33) can be shown to have the following solutions:

\[
w_0 = \frac{\mu_2 \sigma_1^2 - \mu_1 \sigma_2^2 \pm \sigma_1 \sigma_2 \sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2) \ln \left( \frac{\tau \sigma_1}{\sigma_2} \right)}}{\sigma_1^2 - \sigma_2^2}
\]  

Nevertheless, since there are two solutions to \( w_0 \) in (34), a choice has to be made as to which of them is substituted into (31). To eliminate one of the solutions, we consider the second-order partial derivative of \( p_e \) with respect to \( w_0 \) evaluated at \( w_0 \) as given by (34), and determine under what condition it is greater than or equal to zero. This is a second-order necessary condition for \( p_e \) to be a local minimum. From (27), it can be shown that:

\[
\frac{\partial^2 p_e}{\partial w_0^2} = \pi_1 \left( -\frac{z_1}{\sigma_1^2} e^{-z_1^2/2} \right) + \pi_2 \left( \frac{z_2}{\sigma_2^2} e^{-z_2^2/2} \right)
\]  

We denote this second-order derivative by \( h \). We then consider all possibilities of \( z_1 \) and \( z_2 \) (which are the variables in (35) that depend on \( w_0 \)) under three cases, and analyse the sign of \( h \) in each.

**Case 1**

\( z_2 \leq 0 \) and \( z_1 \geq 0 \): then \( h \) is trivially non-negative.

**Case 2**

\( z_2 \geq 0 \) and \( z_1 \leq 0 \): then \( h \) is trivially non-negative.

**Case 3**

\( z_2 > 0 \) and \( z_1 > 0 \): then \( h \) is non-negative if and only if

\[
\ln \left( \frac{\pi_2 z_2}{\sigma_2^2} \right) - \frac{z_2^2}{2} \geq \ln \left( \frac{\pi_1 z_1}{\sigma_1^2} \right) - \frac{z_1^2}{2}
\]  

i.e.,

\[
\ln \frac{z_2}{\sigma_2^2} / \frac{z_1}{\sigma_1^2} \geq \frac{z_2^2}{2} - \frac{z_1^2}{2} - \ln \left( \frac{\tau \sigma_1}{\sigma_2} \right)
\]  

It will be noted that the right-hand side of the inequality (37) is identically zero, as can be seen from (32). Therefore, the condition under which \( h \) is greater than or equal to zero is when:

\[
\frac{z_2}{\sigma_2^2} \geq \frac{z_1}{\sigma_1^2}
\]  

Note also that Case 2 necessarily satisfies (38) so that we consider (38) as the general inequality for the non-negativity of \( h \) for all cases, and thus for \( w_0 \) to be a local minimum.
Now, when one considers the two solutions of $w_0$ in (35), only the solution given by:

$$w_0 = \frac{\mu_2 \sigma_1^2 - \mu_1 \sigma_2^2 + \sigma_1 \sigma_2 \sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2)}}{\sigma_1^2 - \sigma_2^2}$$

(39)

satisfies the inequality of (38), i.e., only this choice of $w_0$ corresponds to a local minimum. The proof of this is given in the appendix. We may then substitute this expression of $w_0$ into (31) so that (31) is in terms of $w$ only. Even so, $w$ has to be solved for iteratively. This is because (31) has no closed-form solution since $\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$ are themselves functions of $w$. As the iterative procedure requires an initial choice of $w$, we use Fisher’s choice of the weight vector as given by:

$$w = (n_1 \Sigma_1 + n_2 \Sigma_2)^{-1} (\bar{x}_1 - \bar{x}_2)$$

(40)

as our initial solution. Again, we mention that $n_1$ and $n_2$ are the cardinalities of $D_1$ and $D_2$. After a number of such iterative updates, the optimal $w_0$ is then solved for from (39). This algorithm, known as the Gaussian Linear Discriminant (GLD), is described in detail in Algorithm 1.

**Algorithm 1 GLD.**

1. Input: $D_1$ and $D_2$
2. Evaluate $\bar{x}_1$, $\bar{x}_2$, $\Sigma_1$, $\Sigma_2$
3. Initialise $w = (n_1 \Sigma_1 + n_2 \Sigma_2)^{-1} (\bar{x}_1 - \bar{x}_2)$
4. Evaluate $\mu_1$, $\mu_2$, $\sigma_1^2$, $\sigma_2^2$, $z_1$, $z_2$.
5. While Stopping criteria are not satisfied do
6. Solve for $w_0$ from (39)
7. Evaluate $z_1$, $z_2$.
8. Evaluate the Bayes error $p_e$
9. Update $w$ as $w = (\sigma^2_2 \Sigma_2 - \sigma^2_1 \Sigma_1)^{-1} (\bar{x}_1 - \bar{x}_2)$
10. Evaluate $\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$.
11. End while

Note that by multiplying both $w$ of (31) and $w_0$ proportionally by $c = (\sigma_1 z_2 - \sigma_2 z_1)/\sigma_1 \sigma_2$ (due to (38), $c$ is non-negative and hence the discrimination criterion given by (15) is not changed), the GLD may be viewed in terms of the optimal solution of (11), where

$$s = -\sigma_2 z_1/(\sigma_1 z_2 - \sigma_2 z_1).$$

(41)

which is unbounded given the inequality of (38). However, unlike Anderson and Bahadur (1962); Marks and Dunn (1974), $s$ is not chosen by systematic trial and error, and unlike Fukunaga (2013), $s$ is not varied between 0 and 1 at small step increments. Instead, since $s$ is a function of $w$ and $w_0$, our algorithm may be interpreted as obtaining increasingly refined values of $s$ by improving upon $w$ and $w_0$ starting from Fisher’s solution, as is described in Algorithm 1.

**3.2. Multiclass classification**

Suppose now that there are $K > 2$ classes in the dataset $\mathcal{D}$, then the classification problem may be reduced to a number of binary classification problems. The two main approaches usually taken for this reduction are the One-vs-All (OvA) and One-vs-One (OvO) strategies (Bishop, 2006; Hsu & Lin, 2002).

**3.2.1. One-vs-All (OvA)**

In OvA, one trains a classifier to discriminate between one class and all other classes. Thus, there are $K$ different classifiers. An unknown vector $x$ is then tested on all $K$ classifiers so that the class corresponding to the classifier with the highest discriminant score is chosen. However, with respect to the proposed GLD algorithm, this is an ill-suited approach. This is because the collection of all other classes on one side of the discriminant will not necessarily have a normal distribution, and could in fact be multimodal, if the means are well-separated. Since our algorithm is built on strong normality assumptions of the data on each side of the discriminant, the GLD, as has been formulated, is expected to perform poorly.

**3.2.2. One-vs-One**

In OvO, a classifier is trained to discriminate between every pair of classes in the dataset, ignoring the other $K - 2$ classes. Thus, there are $K(K - 1)/2$ unique classifiers that may be constructed. Again, an unknown vector $x$ is tested on all $K(K - 1)/2$ classifiers. The predicted classes for all the classifiers are then tallied so that the class that occurs most frequently is chosen. This is equivalent to a majority vote decision. In a lot of cases, however, there is no clear-cut winner, as more than one class may have the highest number of votes. In such a case, the most likely class is often chosen randomly between those most frequently occurring classes. The GLD provides a more appropriate means for breaking such ties, by making use of the minimised Bayes error $p_e$ for each classifier. Specifically, one may instead use a weighted voting system, where the count of every predicted class is weighted by $1 - p_e$, since $p_e$ provides an appropriate measure of uncertainty associated with each classifier output. Thus, the decision rule is reduced to choosing the maximum weighted vote among the $K$ classes.

Note that even though the GLD minimises the Bayes error for each classifier, the overall Bayes error for a multiclass problem may not be minimised by using multiple binary classifiers.

**4. Non-normal distributions**

So far, the fundamental assumption that has been used to derive the GLD is that the data in each class has a normal distribution. Thus, for an unknown non-normal distribution, the linear classifier we have obtained does not minimise the Bayes error for that unknown distribution. We argue, however, that if this unknown distribution is nearly-normal (Mudholkar & Hutson, 2000), then a more robust linear classifier may be found in some neighborhood of the GLD. For this reason, we use a local neighbourhood search algorithm to explore the region in $\mathbb{R}^{d+1}$ around the GLD to obtain the classifier that minimises the number of misclassifications on the training dataset. We do this by perturbing each of the $d + 1$ vector elements in the optimal $w = [w_0, w']^T$ obtained from the GLD procedure by a small amount $\delta w$. After every perturbation, the resulting classifier is evaluated on the test dataset.
This procedure is repeated as described in Algorithm 2 until the stopping criterion is satisfied.

The algorithm is terminated after a certain maximum number of iterations \( R \) is reached. Additionally, one may perform an early termination if after a predefined number of iterations \( i_{\text{max}} \), there is no improvement in the minimum number of classifications on the training dataset that has been found in the search.

5. Experimental validation

We validate our proposed algorithm on two artificial datasets denoted by D1 and D2, as well as on ten real-world datasets taken from the University of California, Irvine (UCI) Machine Learning Repository. These datasets are shown in Table 1, and cut across a wide range of applications including handwriting recognition, medical diagnosis, remote sensing and spam filtering. D1 and D2 are normally distributed with different covariance matrices. For D1, we generate 1000 samples for class \( C_1 \) and 2000 samples for class \( C_2 \) using the following Gaussian parameters:

\[
\mathbf{x}_1 = [3.86, 3.10, 0.84, 0.84, 1.64, 1.08, 0.26, 0.01]^T
\]

\[
\Sigma_1 = \begin{bmatrix}
0.25 & 0.75 & 1.25 & 1.75 \\
0.25 & 0.75 & 1.25 & 1.75 \\
1.25 & 1.25 & 1.75 & 2.73 \\
1.75 & 1.75 & 2.73 & 3.78
\end{bmatrix}
\]

\[
\mathbf{x}_2 = [-1.5, -0.75, 0.75, 1.5]^T
\]

For D2, we generate 2000 samples for class \( C_1 \) and 4000 samples for class \( C_2 \) using the following Gaussian parameters:

\[
\mathbf{x}_2 = [-1.5, -0.75, 0.75, 1.5]^T
\]
element by 10% of its absolute value, i.e. $\delta \tilde{w}_i = 0.1 |\tilde{w}_i|$, and we run for $R=1000$ iterations, terminating prematurely if $r_{\text{max}} = 0.1R$. We use a step size of $\Delta s = 0.001$ for the C-HLD algorithm, and run 1000 trials for R-HLD-1 and R-HLD-2. All the parameters used in the experiments are optimised via cross-validation. Note that if the sample covariance matrix is singular, we use the Moore–Penrose pseudo-inverse.

6. Results and discussion

For real-world datasets, the covariance matrices of the classes are rarely equal, therefore the homoscedastic assumption in LDA does not hold. Our results in Table 2 confirm that LDA does not minimise the Bayes error under heteroscedasticity, as none of the datasets used has equal covariance matrices. With the exception of datasets (d), (j) and (l), where LDA achieves an equal Bayes error as the other heteroscedastic LDA approaches, LDA is outperformed by the GLD on all remaining datasets in terms of minimising the Bayes error. It will be noted that the other three heteroscedastic LDA approaches algorithms achieve a performance comparable to the GLD on all the datasets in terms of the Bayes error. However, R-HLD-1 and R-HLD-2 require a lot of trials (1000 in our experiments) in order to obtain the optimal parameters $s$ and $s_1$, $s_2$ respectively, while C-HLD requires a step size of $\Delta s = 0.001$ which translates to 1000 trials. Consequently, the training time for these algorithms far exceed that of the GLD, as can be seen in Table 4. For example, the gain in training time of the GLD over C-HLD, R-HLD-1 and R-HLD-2 is over 62 folds for dataset (g), and about 20 folds for dataset (l). Moreover, since C-HLD, R-HLD-1 and R-HLD-2 all require matrix inversions, performing a matrix inversion for each of the 1000 trials can be a computationally demanding task especially for high-dimensional data, which have large covariance matrices. Instead, since the GLD follows a principled optimisation procedure, the number of matrix inversions required is far lower. For example, on dataset (f), which has a dimensionality of 47, the GLD requires over 60 times less time to train than the other heteroscedastic LDA approaches.

It is conceivable that the minimisation of the Bayes error would translate into a good performance in terms of the classification accuracy, if the normality assumption of LDA holds. For this reason, it can be seen in Table 3 that the GLD achieves the best classification accuracy on datasets (a) and (b), which are generated from known normal distributions. Thus, the proposed GLD algorithm is particularly suited for applications with datasets that tend to be normally distributed in each class e.g. in machine fault diagnosis, or accelerometer-based human activity recognition (Ojetola, Gaura, & Brusey, 2015), as it also requires far less training time than the existing heteroscedastic LDA approaches.

However, for datasets (c) through to (l), the classes do not have any known normal distribution. Therefore, minimising the Bayes error under the normality assumption would not necessarily result in a classifier that has the best classification accuracy, even if the difference in covariance matrices has been accounted for. For this reason, it is not surprising that LDA achieves a superior classification accuracy than C-HLD, R-HLD-1, R-HLD-2 and the GLD on datasets (c) and (h) as can be seen in Table 3. However, by searching around the neighbourhood of the GLD, the local neighbourhood search (LNS) algorithm is able to account for the non-normality and obtain a more robust classifier. Thus, the GLD, together with the LNS procedure, achieves a higher classification accuracy than all the LDA approaches on all the real-world datasets (i.e. (c)–(l)) with the exception of dataset (f) which has the GLD showing superior classification accuracy.

While the SVM outperforms the LDA approaches on half of the datasets, its training time can be rather long for large datasets. For instance, for dataset (d) which has 58000 elements, the SVM takes about 1.3 h to train whereas the GLD with LNS, which achieves the best classification accuracy on this dataset, takes 43 s to train, representing over 100 fold savings in computational time over the SVM. Similar patterns can be seen in other datasets like (i), where the GLD with LNS achieves a superior classification accuracy with over 150 times shorter training time than the SVM. This suggests that for such large datasets, the GLD with local neighbourhood search is a low-complexity alternative to the SVM, as it requires far less computational time than the SVM.

We, however, make note of two weaknesses of our proposed algorithms have. For the GLD, the procedure as described in Algorithm 1, may converge to a saddle point, instead of a local minimum. Even if it were to converge to a local minimum, there is no guarantee that is the global optimum solution due to the fact that the objective function $p_c$ is known to be non-convex (Anderson & Bahadur, 1962). Also, since the local neighbourhood search involves evaluating the misclassification rate on the training set for every perturbation, the procedure does not scale well with large amounts of training data. Because of this, it is important to have a good initial solution like the GLD, so that an early termination may be performed if there is no improvement after some number of iterations.

7. Conclusion

In this paper, we have presented the Gaussian Linear Discriminant (GLD), a novel and computationally efficient method for obtaining a linear discriminant for heteroscedastic Linear Discriminant Analysis (LDA) for the purpose of binary classification. Our algorithm minimises the Bayes error via an iterative optimisation procedure that uses Fisher's Linear Discriminant as the initial so-

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Average training time (s).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>LDA</td>
</tr>
<tr>
<td>(a)</td>
<td>0.001</td>
</tr>
<tr>
<td>(b)</td>
<td>0.001</td>
</tr>
<tr>
<td>(c)</td>
<td>0.001</td>
</tr>
<tr>
<td>(d)</td>
<td>0.037</td>
</tr>
<tr>
<td>(e)</td>
<td>0.036</td>
</tr>
<tr>
<td>(f)</td>
<td>3.307</td>
</tr>
<tr>
<td>(g)</td>
<td>0.128</td>
</tr>
<tr>
<td>(h)</td>
<td>0.101</td>
</tr>
<tr>
<td>(i)</td>
<td>0.017</td>
</tr>
<tr>
<td>(j)</td>
<td>0.638</td>
</tr>
<tr>
<td>(k)</td>
<td>0.304</td>
</tr>
<tr>
<td>(l)</td>
<td>0.835</td>
</tr>
</tbody>
</table>

This table shows the average training times on the test datasets for LDA, C-HLD, R-HLD-1, R-HLD-2, GLD, GLD+LNS and SVM. Best values are in bold.
lution. Moreover, the GLD does not require any parameter adjustments. We have also proposed a local neighbourhood search method by which a more robust linear classifier may be obtained for non-normal distributions. Our experimental results on two artificial and ten real world applications show that when the covariance matrices of the classes are unequal, LDA is unable to minimise the Bayes error. Thus, under heteroscedasticity, our proposed algorithm achieves superior classification accuracy to the LDA for normally distributed classes. While the proposed GLD algorithm compares favourably with other heteroscedastic LDA approaches, the GLD requires a far less training time. Moreover, the GLD, together with the LNS, has been shown to be particularly robust, comparing favourably with the SVM, but requiring far less training time on our datasets. Thus, for expert systems like machine fault diagnosis or human activity monitoring that require linear classification, the proposed algorithms provide a low-complexity, high-accuracy solution.

While this work has focused on linear classification, on-going work is focused on modifying the GLD procedure for the purpose of linear dimensionality reduction. Moreover, it is of particular interest to us to be able to derive the Bayes error for some known non-normal distributions. An alternative to this is to be able to obtain a kernel that implicitly transforms some data of a known non-normal distribution into a feature space where the classes are normally distributed. Finally, like all local search algorithms, the performance and complexity of the LNS procedure depends on the choice of the initial solution. Therefore, further work that explores the use initial solutions (including the heteroscedastic LDA approaches discussed) other than the GLD for the LNS procedure is being done.

**Appendix A**

**Theorem 1.** Let \( w_0^1 \) and \( w_0^2 \) be the two distinct solutions of (34), then \( w_0^1 \) and \( w_0^2 \) cannot both satisfy (38) given that \( \sigma_1 \neq \sigma_2 \).

**Proof.** Let

\[
\beta = \sqrt{\frac{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2)}{\ln \frac{\sigma_1}{\sigma_2}}} \tag{A.1}
\]

and let

\[
w_0^1 = \frac{\mu_2 \sigma_1^2 - \mu_1 \sigma_2^2 + \sigma_1 \sigma_2 \beta}{\sigma_1^2 - \sigma_2^2} \tag{A.2}
\]

Then

\[
z_2 = \frac{(\mu_2 - \mu_1) \sigma_2 + \beta \sigma_1}{\sigma_2 (\sigma_1^2 - \sigma_2^2)}, \quad z_1 = \frac{(\mu_2 - \mu_1) \sigma_1 + \beta \sigma_2}{\sigma_1 (\sigma_1^2 - \sigma_2^2)} \tag{A.3}
\]

Suppose that \( w_0^1 \) satisfies (38), then

\[
\frac{(\mu_2 - \mu_1) \sigma_2 + \beta \sigma_1}{\sigma_2 (\sigma_1^2 - \sigma_2^2)} \geq \frac{(\mu_2 - \mu_1) \sigma_1 + \beta \sigma_2}{\sigma_1 (\sigma_1^2 - \sigma_2^2)} \tag{A.4}
\]

i.e.,

\[
\frac{\beta \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{\beta \sigma_1^2}{\sigma_1^2 - \sigma_2^2} \tag{A.5}
\]

This implies that \( \sigma_2^2/(\sigma_1^2 - \sigma_2^2) > \sigma_1^2/(\sigma_1^2 - \sigma_2^2) \) since \( \beta \) is a positive scalar.

Consider now \( w_0^2 \) given as:

\[
w_0^2 = \frac{\mu_2 \sigma_1^2 - \mu_1 \sigma_2^2 - \sigma_1 \sigma_2 \beta}{\sigma_1^2 - \sigma_2^2} \tag{A.6}
\]

Then

\[
z_2 = \frac{(\mu_2 - \mu_1) \sigma_2 - \beta \sigma_1}{\sigma_2 (\sigma_1^2 - \sigma_2^2)}, \quad z_1 = \frac{(\mu_2 - \mu_1) \sigma_1 - \beta \sigma_2}{\sigma_1 (\sigma_1^2 - \sigma_2^2)} \tag{A.7}
\]

In order for (38) to be satisfied, it can be shown, similar to (A.5), that

\[
\frac{\beta \sigma_1^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{\beta \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \tag{A.8}
\]

which can be simplified to give \( 1 \geq 0 \). Since this conclusion is false, only \( w_0^1 \) satisfies (26). □

**References**


Loog, M., & Duin, R. P. (2002). Non-iterative heteroscedastic linear dimension reduction for two-class data. In Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR) (pp. 508–517). Springer.


