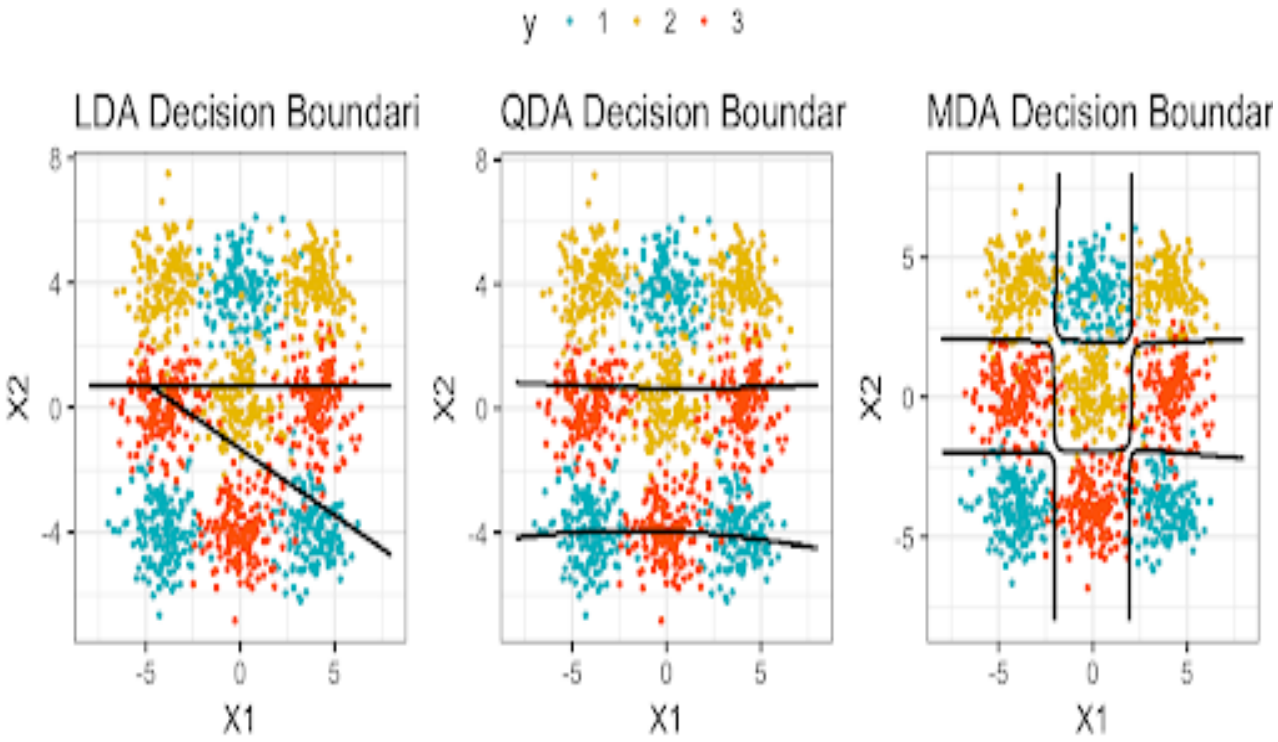
CS 383 – Final Paper

Quadratic Discriminant Analysis

Ridwan Olawin



Abstract

Classification in machine learning is a supervised learning approach in which a computer program learns from the data given to it and makes new observations or classifications. There are a variety of classification algorithms out there but the one factor that stands out amongst them is their individual accuracy and efficiency. This has led to scientists making assumptions about data before implementing the algorithm to get the best classifier possible. One of those methods led to the creation of generative classifiers which uses data to make further assumptions about how to classify itself. In this paper, we explore in detail Quadratic Discriminant Analysis (QDA), a generative classifier, its advantages, and disadvantages over similar algorithms to itself.

Background

Generative classifiers are typically used to generate models of joint probability distribution. This means there exists an input and target variable (X and y). The solution would be a distribution that could generate new input variables with their respective targets. The general name of the model to be extended in this work is called a Gaussian Discriminant Analysis (GDA) model. For a model of this category, a major assumption here is that the class conditional densities are normally distributed.

$$P(\mathbf{x} \mid t = c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c),$$

Where υ is the class-specific mean vector and Σ is the class-specific covariance matrix. Using bayes classifier, we can calculate the class posterior:

$$\overbrace{P(t = c|\mathbf{x}, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}^{\text{class posterior}} = \frac{\overbrace{P(\mathbf{x} \mid t = c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}^{\text{class-conditional density}} \overbrace{P(t = c)}^{\text{class prior}}}{\sum_{k=1}^{K} P(\mathbf{x} \mid t = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) P(t = k)}.$$

Equation 1

We can then classify **x** into class

$$\hat{h}(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\, P(t = c|\mathbf{x}, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c).$$

Equation 2

Given the above conditions, we are typically trying to reduce the error rate:

Error rate = L(h) = P(h(x) != y)

- y is the actual and h(x) is the predicted value of our target

The bayes classifier depends on unknown quantities so we need to use data to find some approximation to the rule. This resulted in a concept known as **Gaussian Maximum Likelihood Classification (GMLC).**

Related Work

Gaussian Maximum Likelihood Classification

This method assumes that each class (target) has a gaussian distribution and the estimates the distribution from the data, then classifies each new observation to the class with maximum likelihood. There are two main methods that fall under this classification known as Linear Discriminant Analysis (LDA) and QDA. It is important to understand in detail the background of LDA before exploring QDA.

Methodology

Linear Discriminant Analysis

LDA can be used to perform supervised dimensionality reduction (as learned in class). In this case however, it is used to model the class conditional distribution of data P(x|y=k) for each class k.

$$P(x|y=k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k)\right)$$

Given this conditional, we simply want a way to draw the decision boundary between our classes:

{ x | P(y=1|x=x) == P(y=0|x=x) }

We are simply trying to solve for

P(y=1|x=x) == P(y=0|x=x)

A major assumption in LDA is that the class covariances are the same. Figures 1 and 2 show the derivation of the decision boundary.

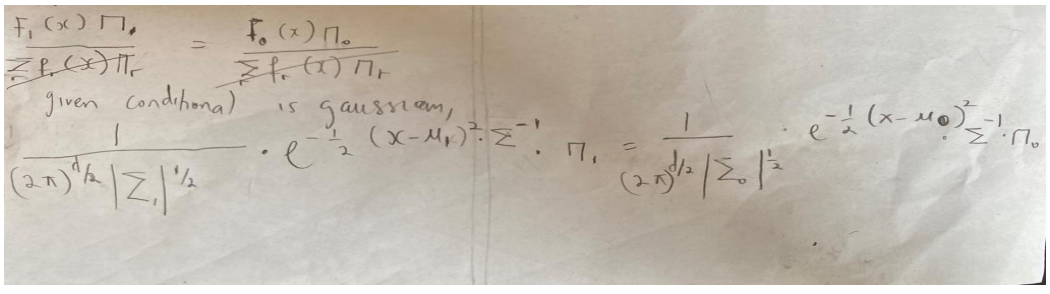


Figure 1

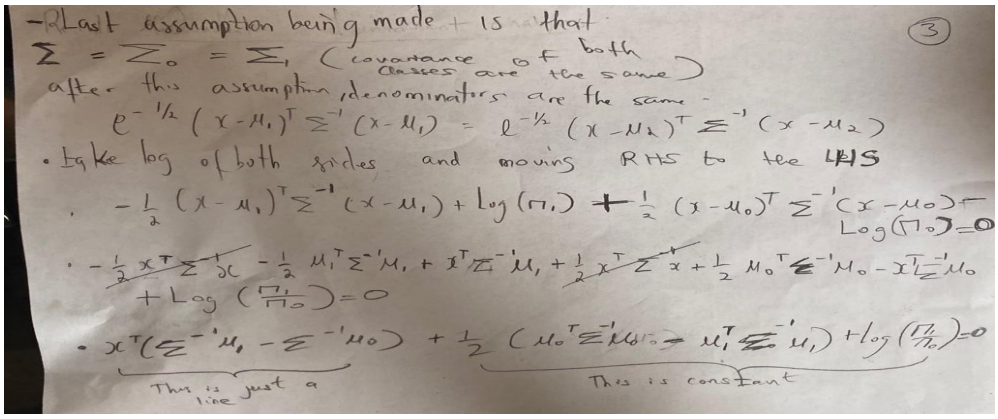


Figure 2

From the derivation, we can see the equation ending in the form:

x.T*B + a = 0; which simply shows the form **mx + c,** the equation of a line.

The resulting graph looks of the form:

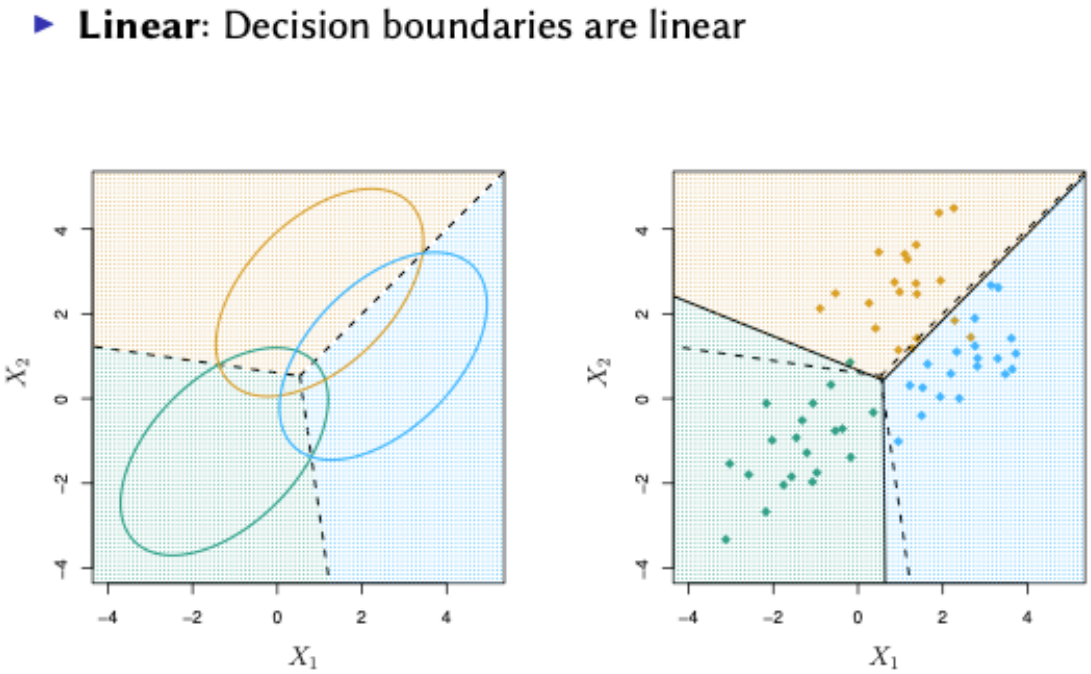


Figure 3

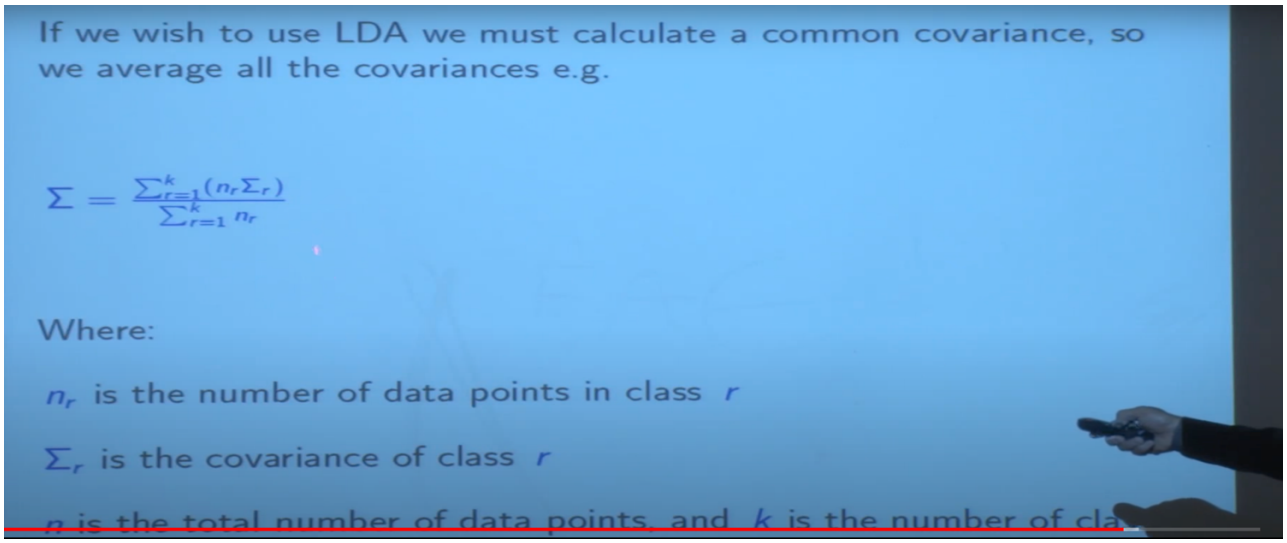
Taking the log of the last equation in Figure 2 would result in the decision boundary equation below:

$$\Sigma_0 = \Sigma_1 = \Sigma_2 = \cdots = \Sigma_{k-1}$$

(e.g. LDA), then

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \mu_k - \tfrac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + log(\pi_k)$$

Some notations to note:



In practice we don't know the parameters of Gaussian and will need to estimate them using our training data.

$$\hat{\pi}_k = \hat{Pr}(y = k) = \frac{n_k}{n}$$

where n_k is the number of class k observations.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\Sigma}_k = \frac{1}{n_k - k} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$$

Quadratic Discriminant Analysis

This method is simply an extension of LDA except one of the assumptions is being changed. The assumption is that the covariances of all classes are different. Its resulting graph is quadratic hence, its name QDA.

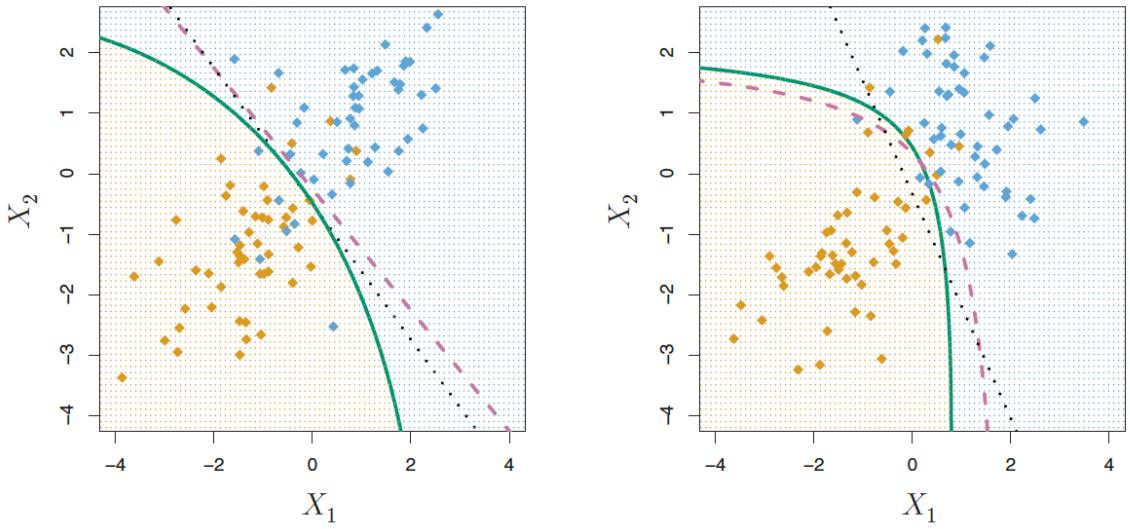


Figure 4

Figures 5 and 6 below simply expand on the different cases of what the covariance matrices could be.

For QDA we need to calculate:

$$\delta_k(\mathbf{x}) = -\tfrac{1}{2}log(|\Sigma_k|) - \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + log(\pi_k)$$

Equation 3

Let's expand on the different cases of the covariant matrices

Case 1:

$\Sigma_k = I$

$\delta_k(x) = -\frac{1}{2}\log(|I|) - \frac{1}{2}(x-\mu_k)^T(x-\mu_k) + \log(\Pi_k)$

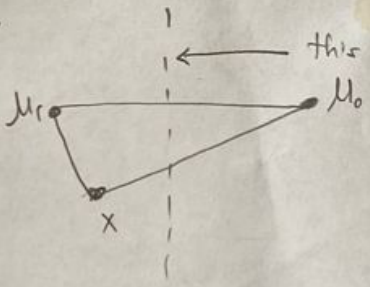
$\log(|I|) = 0$

this is basically a squared Euclidean distance

another assumption, let's assume we have equal class distribution hence, this would be negligible.

- The conclusion here is basically is coming to understand that QDA can be looked at in the sense of squared euclidean distance from the mean of our different classes to a particular data point.

Hence, given $\delta_0(x)$ $d_1(x)$,



Case 2:

$\Sigma_k \neq I$

recall in SVD (singular-value decomposition)

given: $A = U\Sigma V^T$

U is eigenvectors of AA^T

V " " " A^TA

Σ is a diagonal matrix such that diagonal values are eigenvalues of AA^T or A^TA

Figure 5

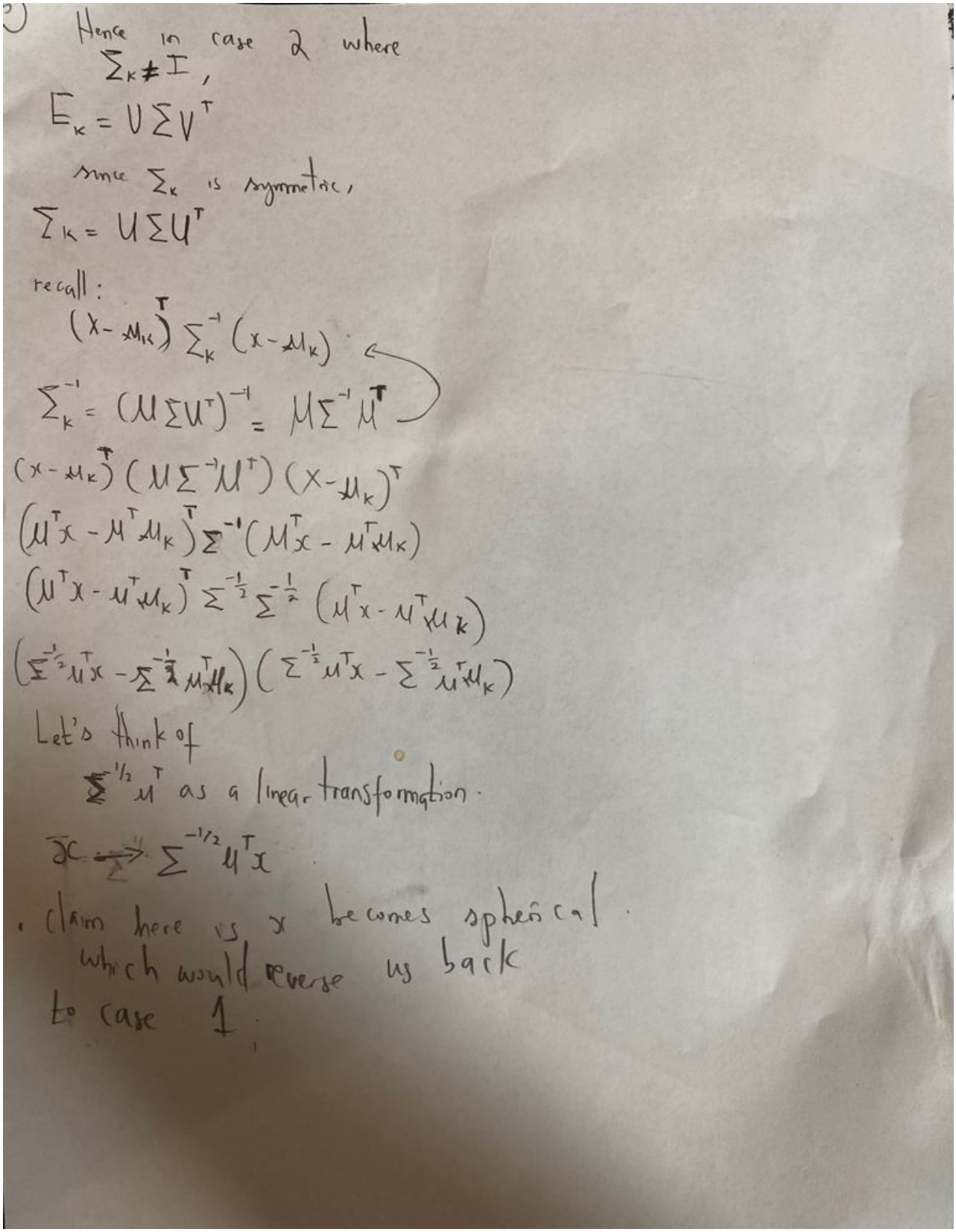

Hence in case 2 where
$\Sigma_k \neq I$,
$E_k = U \Sigma V^T$
since Σ_k is symmetric,
$\Sigma_k = U \Sigma U^T$
recall:
$(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$
$\Sigma_k^{-1} = (U \Sigma U^T)^{-1} = U \Sigma^{-1} U^T$
$(x - \mu_k)^T (U \Sigma^{-1} U^T)(x - \mu_k)^T$
$(U^T x - U^T \mu_k)^T \Sigma^{-1} (U^T x - U^T \mu_k)$
$(U^T x - U^T \mu_k)^T \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} (U^T x - U^T \mu_k)$
$(\Sigma^{-\frac{1}{2}} U^T x - \Sigma^{-\frac{1}{2}} U^T \mu_k)(\Sigma^{-\frac{1}{2}} U^T x - \Sigma^{-\frac{1}{2}} U^T \mu_k)$
Let's think of
$\Sigma^{-1/2} U^T$ as a linear transformation.
$x \Rightarrow \Sigma^{-1/2} U^T x$
• claim here is x becomes spherical.
which would reverse us back
to case 1.

Figure 6

Experiments and Result

To show the implementation of these classification methods, I used a Kaggle data set which contained data about passengers on board the titanic. Some exploratory analysis of the data set showed the distribution of the data including missing data sets and unclear ones.

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Figure 7

My experiment process revolved around implementing both LDA and QDA from scratch using the methodology above. The goal of this was to show the improvement in accuracy QDA had over LDA in this data set. The reason why this occurred was because this data set had a relatively larger training set and higher variance. The results are below:

```
Precision = 0.4257
Recall = 0.5676
F_measure = 0.4865
Accuracy for LDA Mine: 0.503731343283582
```

Figure 8: LDA

```
Precision = 0.7358
Recall = 0.7027
F_measure = 0.7189
Accuracy for QDA Mine: 0.7723880597014925
```

Figure 9: QDA

Conclusion

From the results above, we can see that QDA performs considerably better than LDA simply because our initial assumptions worked better for this data set. A few things to note specifically on this data set was that I noticed that being a female or a child increases the chances of survival. A higher-class ticket also improved survival compared to a third- or fourth-class ticket.

Future Work

Another algorithm I am looking forward to exploring is Multiple Discriminant Analysis (MDA). This algorithm is a multivariate dimensionality reduction technique (Similar to LDA). However, it can also be used to support classification by yielding a compressed signal which can then be used for classification. The method also reduces the curse of dimensionality by compressing the signal (Features) down to a lower dimensional space (Similar to LDA). What makes this method unique is its focus on when three or more target variables are involved, it can compute more than one function as opposed to LDA which only computes one.

Bibliography

Julenn. (2021, May 15). *Titanic*. Kaggle. https://www.kaggle.com/julenn/titanic.

YouTube. (2015, September 27). *Ali Ghodsi, Lec 2: Machine learning. classification, Linear and quadrtic discriminant analysis*. YouTube. https://www.youtube.com/watch?v=_m7TMkzZzus.

Petrik, M. (2017, February 16). *LDA, QDA, Naive Bayes*. Generative Classification Models. https://marek.petrik.us/teaching/intro_ml_17/intro_ml_17_files/class5.pdf.