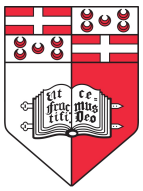

University of Malta

Faculty of Science

Gaussian Process Classification of Sportsbook Customers

Michael Filletti

supervised by

Prof. Lino Sant

May 2016

A dissertation presented to the Faculty of Science in part fulfillment of the requirements for the degree of Bachelor of Science (Hons.) at the University of Malta

Contents

List of Figures

List of Tables

Declaration of Authorship

I, Michael Ayrton Filletti (348794M), declare that this dissertation entitled:

"Gaussian Process Classification of Sportsbook Customers",

and the work presented in it is my own.

I confirm that:

1. This work is carried out under the auspices of the Department of Statistics and Operations Research as part fulfillment of the requirements of the Bachelor of Science (Hons.) course.

2. Where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.

3. Where I have used or consulted the published work of others, this is always clearly attributed.

4. Where I have quoted from the works of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.

5. I have acknowledged all sources used for the purpose of this work.

Signature: _____

Date: _____

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Lino Sant, for his full support and guidance throughout my study and research. Without his patience and knowledge this thesis would not have been possible.

I would also like to thank my parents, my sister, my girlfriend, and my grandparents for their unconditional love and support, as without them I would not have been able to complete this thesis.

Abstract

Segmentation is an underrated tool in management science that is many times implemented for different purposes, marketing being the more common. Classification of customers could be of great use to the online betting industry. In this dissertation, we shall use segmentation techniques on a sportsbook dataset using a number of customer characteristics as well as playing habits and performances. Gaussian processes have been much studied and harnessed to aid with diverse problems in statistics; regression and classification being major beneficiaries. In the work developed here techniques using Gaussian processes are considered at length studied and applied to the data. Classical clustering techniques offered benchmarks, background and context for comparative and evaluative purposes.

Notation

$\overset{c}{=}$: Equal up to an additive constant

$\overset{\Delta}{=}$: Equality which acts as a definition

$\|\mathbf{z}\|$: Size of a vector $\mathbf{z}$

$\mathbf{0}$: Vector of zeros

$\mathbf{1}$: Vector of ones

$\alpha(\cdot)$: Decision function

C: Number of clusters

$Cov\left(\mathbf{Z}\right)$: Covariance of vector $\mathbf{Z}$

δ_{ij}: Kronecker delta

$\mathbb{D}$: Matrix of distances containing elements d_{ij}

D_{ij}: Distance between the i^{th} and j^{th} entry

$diag(\mathbf{z})$: Diagonal matrix containing elements of the vector $\mathbf{z}$

$\mathbb{E}\left[Z\right]$: Expected value of some random variable Z

$\mathbb{E}_q\left[z(\mathbf{x})\right]$: Expectation of $z(\mathbf{x})$ when $\mathbf{x} \sim q$

$\boldsymbol{\varepsilon}$: Vector of error terms

$f(\mathbf{y}|\mathbf{X} = \mathbf{x})$: Density function of $\mathbf{Y} \sim \mathbf{y}$ conditional on a vector with value $\mathbf{X} = \mathbf{x}$

$\mathcal{F}$: Sigma algebra

γ: Decision rule outcome

$\mathbf{H}$: Random vector of latent values from dataset $\mathbb{X}$

H_*: Random latent value from test dataset $\mathbb{X}_*$

$\mathbf{h}$: Known vector of latent values

h_i: i^{th} entry of the latent vector $\mathbf{h}$

h_i^c: i^{th} entry of the latent vector $\mathbf{h}$ for c^{th} class

I: Identity matrix

$K(\cdot, \cdot)$: Covariance function

$l\left(\alpha\left(\cdot\right), y\right)$: Loss function with inputs selected decision $\alpha\left(\cdot\right)$ and outcome y

$\log(z)$: Natural logarithm

L^2: Space of square integrable functions

$L^2[a, b]$: Space of square integrable functions on the interval $[a, b]$

$L^2(P)$: Space of square integrable functions on the set P

N: Number of observations in $\mathbb{X}$

N_*: Number of observations in $\mathbb{X}_*$

$\mathbb{N}$: Set of natural numbers

$\mathcal{N}(\mu, \sigma)$: Normal distribution with mean μ and variance σ

∇: Partial derivative

$\nabla\nabla$: Second partial derivative, used to obtain Hessian

$\mathbf{\Omega}$: Parameter space

p: Number of parameters

ϕ_{ij}: Correlation coefficient

$\mathbf{\Phi}$: Feature map with input $\mathbb{X}$

$R(\cdot)$: Risk function with input decision $\alpha(\cdot)$

$\mathbb{R}$: Set of real numbers

$\mathbb{R}^d$: Set of d-dimensional real vectors

$\mathcal{R}_i$: Decision region of the class i

r_k: Represents the range of observations of the k^{th} variable

$\mathbf{\Sigma}_p$: Covariance matrix of order p

$S(z)$: Sigmoid function

S_{ij}: Similarity coefficient

$\text{supp}(\kappa)$: The support of a measure κ

T: Generic index set

θ_i: i^{th} cluster

$\mathbf{W}$: Random vector of parameter values

$\mathbf{w}$: Known vector of parameter values

w_j: j^{th} parameter value

$(X_t)_{t \in T}$: Stochastic process indexed by $t \in T$

$\mathbb{X}$: The input dataset matrix of dimension $p \times N$

$\mathbf{X}$: Random dataset matrix of dimension $p \times N$

$\mathbf{x}$: A vector of observed values belonging to the matrix $\mathbb{X}$ of dimension $p \times 1$

$\mathbb{X}_*$: The unclassified data matrix of dimension $p \times N_*$

$\mathbf{X}_*$: Unclassified random vector of dimension $p \times 1$ belonging to $\mathbb{X}_*$

$\mathbf{x}_*$: A vector of observed values belonging to the matrix $\mathbb{X}_*$ of dimension $p \times 1$

X_{ij}: Entry in $\mathbf{X}$ which is the value of the j^{th} parameter for the i^{th} observation

$\mathbf{X}_i$: i^{th} random vector

$\mathbf{x}_i$: i^{th} observed vector

$\mathbb{X}^T$: Transpose of matrix $\mathbb{X}$

$\mathbb{X}^{-1}$: Inverse of matrix $\mathbb{X}$

$\mathbf{Y}$: The output vector from the dataset $\mathbb{X}$ of dimension $p \times 1$

$\mathbf{y}$: The observed output vector from the dataset $\mathbb{X}$ of dimension $p \times 1$

$\mathbf{Y}_*$: The output vector from the dataset $\mathbb{X}_*$ of dimension $p \times 1$

$\mathbf{y}_*$: The observed output vector from the dataset $\mathbb{X}$ of dimension $p \times 1$

y: Scalar output

y_i: i^{th} output from the vector $\mathbf{Y}$

$\hat{z}$: Approximation of a value z

$\bar{z_{i\bullet}}$: Mean of values z_{ij} taken over all possible values of j

$\tilde{z}_i$: Normalization term

$\mathbb{Z}$: Set of integers

Chapter 1

Introduction

Classification is a vastly underrated tool in industry and various companies have tended to become more focused on their target market and catering to them, rather than looking at their customers as a whole and treating them accordingly. Classification allows an organization to look at their customers at a macro level, by assigning a customer into a group that best represents them, and following this, their needs can be catered to. In an ideal world, each customer is catered to individually. Using classification we can try to find a balance such that each customer can be catered to while keeping things relatively uncomplicated.

To carry this out, a local betting company provided a data sample of 1000 sports betting customers to be classified into a number of classes. These customers belong to a population of sports bettors and are of different ages, genders and nationalities. There are two types of bets a customer can make, a single bet and a combi bet. A single bet is made when a customer bets on just one event happening, for example, Juventus to win in the Juventus vs Inter match. On the other hand, a combi bet is made when a customer bets on multiple events occurring, for example, Barcelona, Juventus and Arsenal to all win their games, and should one of the teams fail to win their match, the customer loses the bet.

The customers are to be classified into classes according to the type of customer they are. Customers are looked at from the point of view of the company, so there

are some customers who are viewed as threats as they have the potential to win a lot of money, or have already won a lot of money. Naturally, there are various ways that the customers can be classified, depending on the requirements. In this problem, we will attempt to obtain classes based mainly on the amount bet and profit made by the customer. In addition, we are interested in the customers who bet very often and those who don't, the types of bets a customer makes, and so on.

We are trying to identify the customers that are threats and clustering them together. However, the problem is not that simplistic, as there are degrees to which a customer can be said to be a threat. Furthermore, some customers can be taken advantage of, and we also want these customers to be identified and grouped together. It is important to note when classifying the customers, that those who have won money over the past year are not necessarily smart players. This means that customers shouldn't be classified solely based on whether they are winners or losers. It is for this reason, that we choose to identify players based on how much they play, win, and the type of bets placed. However, the number of classes we have is not fixed.

This problem is first approached using a classical technique, cluster analysis. This method is ideal, firstly as it provides a benchmark for comparative purposes, and secondly as the clustered dataset can act as the training dataset for Gaussian classification to be carried out. Following this, the Gaussian classification is implemented, and the remaining dataset is used to test this classification technique. These results are evaluated and compared to the results of cluster analysis. The comparison is made as the results of cluster analysis give us a classification of the dataset.

This dissertation will continue by explaining the theory behind the classical approach, cluster analysis, in Chapter 3, which is required for the initial classification of the customers. Linear regression is also covered in this chapter as this technique is also used for prediction purposes, and is crucial to understanding Gaussian regression. In Chapter 4, Decision Theory and the Bayesian Approach are discussed, which are

useful tools to understanding the classification problem. The Bayesian Approach in particular, is fundamental to the Gaussian approach to regression and classification. Using Gaussian processes in regression requires a mathematical background, explained in Chapter 5. This is a discussion that explains several results related to Gaussian Processes and Reproducing Kernel Hilbert Spaces, which are necessary to be able to use Gaussian processes in such problems. Gaussian process regression is also explained in Chapter 5, and this topic is crucial to understanding Gaussian classification. Chapter 2 involves the literature review, whilst concluding remarks are made in Chapter 6, which summarize the findings and discuss future research.

1.1 Description of Data

1.1.1 Variables of the Data

The dataset under study is made of 1000 customers, all of different ages and nationalities. The variables characterizing each customer are both continuous and categorical, representing betting details of the customer over the timespan between September 2014 and August 2015. The variables involved are:

- MerchantCustomerID: Unique ID assigned to each customer
- TotalStake: Total amount of money staked on bets in Euro
- TotalPayout: Total amount of money won on bets in Euro
- NumberOfBets: Total number of bets placed
- NumberOfWins: Total number of bets won
- Profit: Total profit made by the company from the customer
- No.of.Single.Bets: Number of single bets made
- No.of.Combi.Bets: Number of combi bets made
- Average.Odds: Average odds of the single bets made
- Mobile.Single.Bets: Number of single bets made using a mobile platform
- Web.Single.Bets: Number of single bets made using a web platform
- Live.Single.Bets: Number of single bets made during the event
- Prematch.Single.Bets: Number of single bets made before the event began
- Margin: The percentage of profit made by the company relative to the total stake of the customer
- Gender: Returns a value of 1 if the customer is female, 0 if the customer is male
- Country: Represents the country that the customer resides in

- BetPreference: Returns a value of 1 if the customer bets more combi bets, otherwise returns a value of 0
- PlatformPreference: Returns a value of 1 if the customer bets more on a web platform, otherwise returns a value of 0
- Loser: Returns a value of 1 if there is a positive profit, otherwise returns a value of 0

When applying cluster analysis to this data, it is important to standardize the data to give all variables an equal effect so as to then be able to apply some form of weighting. The values of the meaningful continuous variables within the dataset to be used are weighted according to which variables are considered to be the most important in this classification. The weighting applied to the variables used for classification is displayed below:

Variable	TotalStake	TotalPayout	NumberOfBets
Weight	10	10	10
Variable	NumberOfWins	Profit	SingleBets
Weight	15	15	5
Variable	CombiBets	AverageOdds	MobileSingleBets
Weight	5	5	5
Variable	WebSingleBets	LiveSingleBets	PrematchSingleBets
Weight	5	2	2
Variable	Margin	Gender	Country
Weight	15	1	1
Variable	BetPreference	PlatformPreference	Loser
Weight	3	3	3

Table 1.1.1: Table of weights assigned to each variable

The weights were assigned by taking into consideration which variables were the most important in our clustering goal, and the values assigned were based off intuition and trial and error. The importance of the variables was defined by the company for whom the classification was being carried out. As can be seen, the variables considered to be the most important were the Margin, Profit and NumberOfWins variables. In

addition, the TotalStake, TotalPayout and NumberOfBets variables were given the second highest weights. It should be noted that there is not one correct way of assigning weights [11], but this can be adjusted according to the type of clustering the user wishes to achieve.

1.1.2 Preliminary Analysis of the Data

As a preliminary analysis of the data, the histograms of all the continuous variables were obtained to illustrate the structure of the data, as shown in Figure 1.1.1.

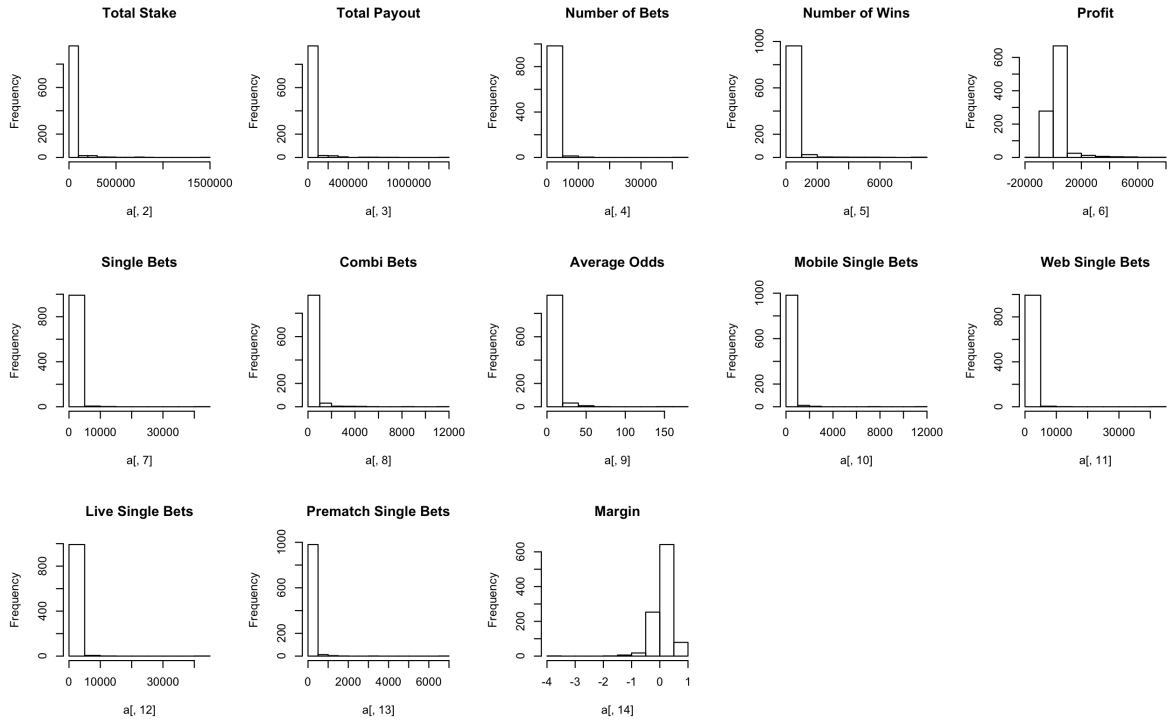

Figure 1.1.1: Histograms illustrating each of the continuous variables in the dataset

A common theme in all these histograms is the unimodal peak. In particular, many variables have one large peak initially, followed by an enormous drop off. In fact, the only variables that do not have this property are the Margin and Profit variables. These histograms indicate that there is a very high percentage of customers that have close values for each variable, whilst the rest of the customers have fairly extreme values and are outliers.

Considering only the customers with similar values and ignoring the outliers, for comparison purposes, gives the histograms a different look. As Figure 1.1.2 illustrates, the histograms are far more civilised than those when considering the entire dataset. In fact, when considering variables such as TotalStake, NumberOfBets and so on, there is a very gradual drop off in the number customers as the value increases, particularly when compared to the histograms in figure 1.1.1. Essentially, what is being shown here is a "zoom in" to the data and an illustration of how these normal customers behave.

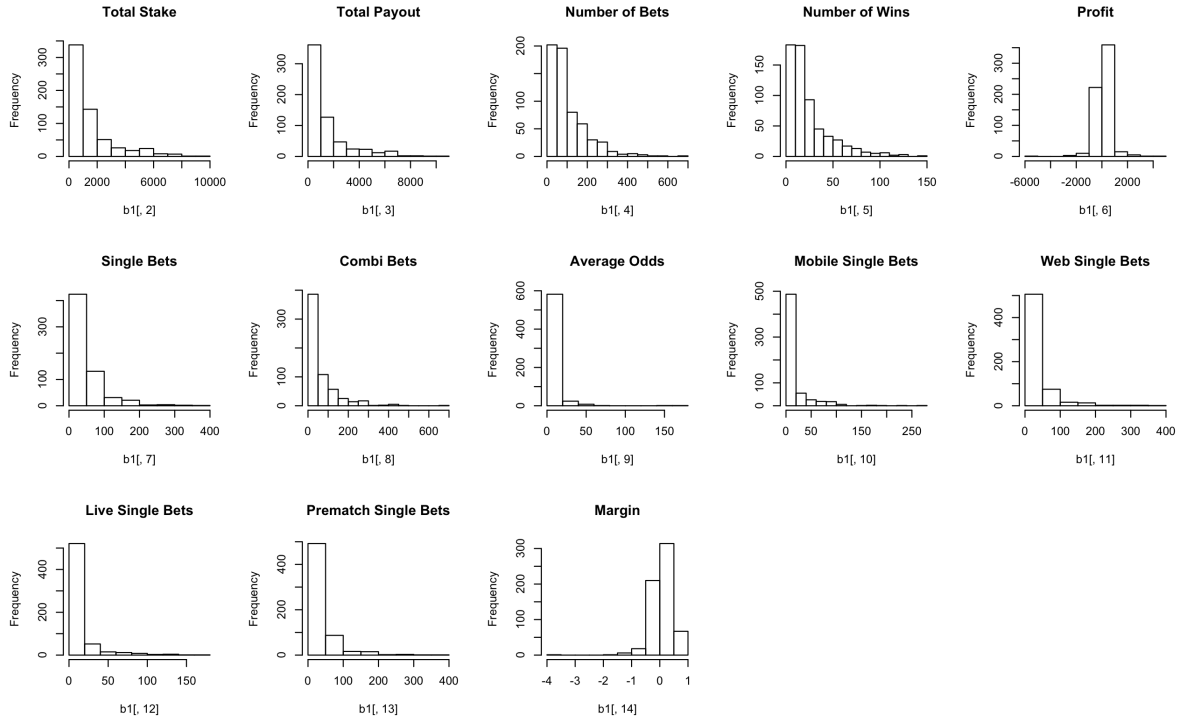

Figure 1.1.2: Histograms illustrating each of the continuous variables for the normal customers

Chapter 2

Literature Review

The data in consideration has no classification framework, so the first step in this problem was to apply some form of classification to it. This issue was approached by first looking at cluster analysis, which is a classical technique used to group a set of observations in a way that observations within the same group are more similar than to those in other groups. B.Everitt (2011) [11] discusses the main methods of finding clusters or groups of homogenous observations in multivariate data.

Once the classes have been identified using cluster analysis, classification techniques are used to construct rules for classifying individuals into the classes. Hence in the next step, Gaussian classification was looked at. Firstly, Decision Theory and the Bayesian Approach are crucial in understanding the Gaussian Process approach to classification that will be investigated. Various papers are available regarding Decision Theory, all of which have their own notation and own method of explaining this topic. The papers which were found to explain this topic best were the paper by Prof. Yuille [36] and the lecture notes by Jason Corso [6]. As the names suggest, both explain Decision Theory with a Bayesian approach, however, they vary as to how they explain it. Prof. Yuille's explanation is more statistically oriented and provides a clear explanation of how the Bayesian approach is implemented. The notation in Corso's explanation is less rigorous in a statistical sense and the explanation is not as sophisticated as that of Prof. Yuille, which is to be expected as it is in the form of lecture notes. Both pieces of literature explain Bayesian Decision Theory by explaining the Bayesian approach and Decision

Theory first and then bring them together.

Gaussian Processes are well known in the world of statistics and their use for making predictions is hardly a new thing [20]. However, using Gaussian Processes to tackle a regression or classification problem is not at all straightforward. For this reason, the mathematical background of this problem is required to be understood before delving into the theory behind Gaussian classification. The paper by Lalley [17] explains Gaussian Processes from a statistical point of view. This article relates Gaussian Processes to various mathematical definitions and theorems, before linking them to Reproducing Kernel Hilbert Spaces (RKHS). It provides results such as Mercer's Theorem, which are critical to the understanding of how Gaussian Processes can be used for regression, and explains how to move from a countable, finite index set to a more general uncountable index set. Reproducing Kernel Hilbert Spaces are required to be investigated since they associate positive definite kernels, which are critical in the subject of Gaussian Processes, with a Hilbert Space of functions. A piece of literature used to explain this theory is the book "Reproducing Kernel Hilbert Spaces in Probability and Statistics" by Berlinet and Thomas-Agnan [4]. This book presents the main points of theory behind RKHS and studies examples of its use in statistics. This book gives a more detailed and complex explanation of RKHS, and is vital to understanding this topic.

The final step before tackling the theory behind Gaussian classification is to look at Gaussian regression, which acts as a fundamental base to the theory behind Gaussian classification. A few papers related to Gaussian Process regression were used, namely that of Mark Ebden[10]. However, the main piece of literature used was Gaussian Processes for Machine Learning, by C.E. Rasmussen & C.K.I Williams [27]. This book explains how to use Gaussian Processes for machine learning tasks. It is explained from a computational angle, and has been widely criticized for the notation used. However, this book is widely used in the area of Gaussian regression and classification, as it explains the theory behind it without delving too deep into the subject of Gaussian Processes. In fact, many writings are derived from this book. The problem we are tackling makes use of Gaussian Process classification, so this literature is vital to being

able to understand the theoretical point of view. The first chapters were of use to us as they explain Gaussian Processes from a machine learning point of view, using them in regression and for classification. The authors give an introduction on the basic theory required to understand this topic, namely Bayesian Inference.

The book [27] begins by explaining regression from a linear point of view, and goes on to generalize that to Gaussian Processes. However, this explanation is contentious, as it is not mathematically sound. This is evident from other literature on Gaussian Processes and Reproducing Kernel Hilbert Spaces, which indicate to us the opposite of what Rasmussen and Williams [27] are claiming. Whereas the authors claim that Gaussian Processes are required when applying the Bayesian approach, the reality is vice-versa, that is, Gaussian Processes require a Bayesian framework for regression to be made possible. As this is explained from a computer science point of view, other literature resources were looked at, which explain how regression can be generalized from a more technical point of view.

The classification chapter of this book is of main interest for our problem, and deals with the theory behind finding the expected class of a new observation. Decision Theory is explained within the context of classification, as a prequel to the main theory discussed in this chapter. The authors explain the linear model for classification and this is extended to Gaussian Processes, using theory established in the regression chapter. Laplace's Approximation and the Expectation Propagation method are also discussed in this chapter. Once again, this chapter is weak from a mathematical point of view, and other papers were required to clarify certain details, usch as those by Ghahramani [13], Atiya [2] and Ebden [9].

Chapter 3

Classical Approaches

3.1 Cluster Analysis

Cluster Analysis is a technique used to cluster observations into groups such that observations within a group are alike, while observations in different groups are as different as possible. This technique does not seek to predict in which pre-existing class new observations will be put in, but instead, simply classifies given observations. The dataset considered can be expressed in terms of a $p \times N$ matrix denoted $\mathbb{X}$, where N is the number of observations and p is the number of attributes.

The entries of a dataset X_{ij} may be continuous, ordinal or nominal, depending on the type of variable, where i represents the parameter and j represents the observation. In practical situations, datasets have a mixture of different types of values. These values are key to clustering data accurately. There are various methods of using these values to cluster the data. Some methods involve plotting observations as data points and finding clusters using graphical methods. This may be used as an initial analysis to get an idea of the structure of the data, and following this a more detailed cluster analysis is carried out. Such an analysis involves creating a new $N \times N$ dataset which contains the distances between each pair of observations. The distance may be calculated using different measures, depending on the type of data at hand. Naturally, the observations found to be the most close are clustered together while those least close from each other are placed in different groups. The number of clusters available is often determined

by the user, whose decision may depend on the data plotted graphically, or on the type of problem being solved. There are various different methods of how to cluster observations, and some of these will be discussed in detail further on.

3.1.1 Basic Methods of Clustering

There are multiple methods of identifying clusters, namely using a histogram or a scatter plot. A histogram is a good indicator of the number of clusters in a dataset, by displaying the mode of the considered variable. Should there be two modes, then it is clear that there are two distinct types of observations, and so on. Naturally, this applies to fairly basic datasets, and is useful in multivariate datasets when trying to find how the data is structured.

Another notable method of identifying clusters involves using scatter plots. Consider an example where the dataset has two variables. Illustrating data in a two dimensional view is a technique which has been used since the 18th century [34]. It is a widely favoured technique as its strength lies in that it depends on the human capability of discerning patterns and possible groups of data. A simple example of identifying clusters through a scatter plot is illustrated below:

Figure 3.1.1: A simple scatter plot of a dataset with 3 distinct clusters

As one can see from the figure, there are three very clearly distinct clusters in this dataset. The three clusters were identified because the data points are close to other data points within the cluster, whilst they are relatively far from data points in other clusters. Applying this logic to more complex datasets gives a method on how to identify clusters within a dataset. Following this, it is natural to consider cases where there are more than two variables. There isn't one specific solution for this situation, however, there are many different techniques which one can use to solve this problem.

Plotting scatter plots of all possible pairs of variables is one option, however, there is an obvious time limitation with this, as there would be too many comparisons to make. This is referred to as a scatter plot matrix, that is a $p \times p$ matrix, where each entry is a scatter plot of one variable against another. This method allows one to compare the plots for one variable against all the others by looking at a row of the matrix, which could prove to be useful. One could question whether there is the need to plot scatter plots of the same pair of variables twice, however, not doing so would mean that it would not be possible to compare plots of one variable against all the others. Possible clusters and outliers may be identified by using this technique, to give a general idea of the structure of the data. On the other hand, this technique is not particularly accurate in a mathematical sense. It would make more sense to illustrate the effects of all variables on one plot and find any patterns from the obtained scatter plot.

3.1.2 Distance Measures

Distance, or similarity, is a key aspect in clustering. It is central to how the observations are clustered together, and represents the relationship between each pair of observations (whether they are similar or dissimilar). Distances between observations are represented in a $N \times N$ matrix with each element representing the distance between one observation and another. Consider a matrix $\mathbb{D}$ with the ij^{th} entry denoted D_{ij} which represents the distance between the i^{th} and j^{th} observations. We call the matrix $\mathbb{D}$ a dissimilarity matrix and it is derived from the dataset matrix $\mathbb{X}$. The way $\mathbb{D}$ is derived from $\mathbb{X}$ depends on the distance measure used. Different measures are used according to the type of data being dealt with. Categorical datasets cannot have the same types of

measures used for continuous datasets, and vice versa. Certain proximity measures are also available to be able to deal with mixed datasets.

3.1.2.1 Mixed Data

Mixed datasets involve both discrete and continuous variables. The measures previously mentioned cannot be applied to a mixed dataset as they would yield incorrect results, since some of the variables would not be suitable for the measure. Hence, the variables must be rescaled or transformed. One possible method of doing this is to convert all variables into binary variables. Doing so would allow binary distance measures that were mentioned above to be applied to the data. However, this method is unnecessarily complex and time-consuming. Another possibility would be to rescale the variables according to their rank to convert the entire dataset into a continuous dataset. Consequently, measures such as the Euclidean measure would become suitable.

One of the most attractive propositions to find the similarity between observations within a mixed dataset is that of Gower (1971), named Gower's general similarity measure. Suppose we define B_{ijk} as the similarity of the i^{th} and j^{th} observation according to the k^{th} variable. W_{ijk} is set to zero if the value for the k^{th} variable of observation i or j is not valid, that is, available, otherwise it is set to one. S_{ijk} is measured differently for categorical and continuous data. For categorical variables:

$$B_{ijk} = \begin{cases} 1 & X_{ik} = X_{jk} \\ 0 & \text{otherwise} \end{cases}$$

On the other hand, for continuous variables, the following formula is utilized:

$$B_{ijk} = 1 - \frac{|X_{ik} - X_{jk}|}{R_k}$$

such that R_k represents the range of observations of the k^{th} variable. Gower's general similarity measure is then expressed as:

$$S_{ij} = \frac{\sum_{k=1}^{p} W_{ijk} B_{ijk}}{\sum_{k=1}^{p} W_{ijk}}$$

3.1.2.2 Weighting

Weighting assigns a value to a particular variable such that it represents its importance in the clustering. As previously mentioned, a value of 1 indicates default importance. The values assigned to each variable show the relative importance of the variable compared to the others. The natural question that follows up is about how the importance of a variable is determined. In truth, there is no fixed method to determining the importance of a variable. The analyst may use judgement and place some greater importance on a certain variable over others due to the type of problem being dealt with.

A popular method of weighting is to use variability weights, which are expressed as $I_k = \frac{1}{Var(X_k)}$. This means that as the variability of a parameter increases, its weight, and hence, importance, decreases. When multiplying the weight by the parameter value, the dataset is being standardized such that the variance of the final model is equal throughout. A problem that arises from this method is that the difference between groups of variables is being eliminated. This works against what is trying to be achieved by clustering, as it is lessening the difference that observations may have for variables with high variance.

3.1.3 Cluster Methods

3.1.3.1 Introduction

The most well known cluster methods are the hierarchical cluster method, the two step algorithm and the k-means clustering. In general, cluster methods can be split into many different types, for example, the hierarchical cluster method is a type of connec-

tivity model, while the k-means clustering is a type of centroid model. Furthermore, the type of clustering may be hard clustering or soft clustering. Hard clustering can be compared to a binary variable, an observation is either in a cluster or it is not, whilst soft clustering assigns a certain value to an observation representing the degree to which it belongs to a certain cluster. In this section we will focus on hierarchical clustering, which is the method used to tackle our problem.

3.1.3.2 Hierarchical Clustering

There are two different types of hierarchical clustering, the agglomerative method and the divisive method. Consider the agglomerative method first, for which each observation is given its own cluster, such that if there are N observations, N clusters are constructed. Using the distance measure selected, the pair of closest clusters are combined into one cluster. This is repeated $N-1$ times, until the analyst ends up with just one cluster, containing all the observations. On the other hand, the divisive method involves all observations placed into one cluster, and this is split until all observations are within their own cluster. The observation with the highest dissimilarity from other observations is removed from the cluster and placed in its own cluster. The method of selecting which distance measure to use between single points has been discussed in section 3.1.2, however, the measure of distance between clusters must also be considered. One such method is called the complete linkage method, which measures the distance D_{AB} between clusters A and B by measuring the furthest distance between an observation in A and an observation in B:

$$\max_{i \in A,\, j \in B} D_{ij}$$

where $i \in$ A and $j \in$ B. The single linkage method is the opposite of the complete linkage method and takes the minimum distance:

$$\min_{i \in A,\, j \in B} D_{ij}$$

where $i \in$ A and $j \in$ B. The average linkage method finds the averages of all the distances between elements in different clusters:

$$D_{AB} = \frac{1}{|\mathrm{A}| \cdot |\mathrm{B}|} \sum_{i \in A} \sum_{j \in B} D_{ij}$$

where |A| is the cardinality of A and similarly for B.

In each step the distance is calculated for all pairs of clusters, and the two clusters A and B which satisfy $D_{AB} = \arg\min_{A,B} \{D_{AB} \; \forall \, \mathrm{A}, \mathrm{B}\}$ are joined together. The number of clusters at each step changes from N to 1 in the agglomerative method, while for the divisive method the opposite occurs. The analyst then chooses the number of clusters which give the optimal result. A dendrogram is useful to select the optimal number of clusters. It is a 2-dimensional graphical representation of the combinations or divisions that take place in hierarchical clustering, and allows the analyst to view the size of the clusters at different steps. The dendrogram is made of nodes, representing the clusters, such that there are internal nodes and terminal nodes. Terminal nodes refer to the observations, which belong to their own cluster in the first step, while internal nodes are those clusters constructed during the process. Dendrograms may have no numerical information displayed, in which case they are termed unranked dendrograms. It should be noted that selecting the number of clusters is a fairly subjective decision, and depends on the type of problem faced.

3.2 Initial Classification of the Data

3.2.1 Methodology

Since my dataset under study is not readily classified, we need to come up with our own classification of customers. A classical approach to this is cluster analysis, which is a fairly straightforward technique used to classify the observations of a dataset. The observations can be clustered in various ways, depending on the weights assigned, and the type of clustering used. The method of clustering chosen is hierarchical clustering, as it allows for more freedom in selecting the number of clusters. R Software was used to carry this out. The hierarchical clustering function used, named *hclust*, requires the

input of a matrix of distances between observations and the type of method to be used. The clustering is executed using an agglomerative method, whilst the complete linkage method was selected to cluster the data. The complete linkage method was preferred due to the structure of the data, as many customers are very similar to each other in terms of the number of bets placed and the monetary value involved. Moreover, it yielded the best results in comparison to the average linkage and single linkage methods. The *cutree* function is needed to select the clusters, for which its input is the number of clusters the user wishes to have, which we shall denote as C. This function looks through the output of the *hclust* function and finds the level at which there are the C clusters. Since the variables are both categorical and continuous, Gower's dissimilarity matrix is constructed, which is a commonly used similarity measure for mixed datasets as discussed in section 3.1.2.1.

Due to the nature of the problem, and the dataset involved, it is not easy to illustrate the results of the clustering and the dataset involved. Principal Component Analysis is a tool that could be used to illustrate the data graphically, to incorporate the effect of each continuous variable. Naturally, this is a very rough technique, however, it is a simple tool that yields satisfactory results. It is a method of condensing the multivariate dataset into a new dataset with a smaller number of variables which are uncorrelated with each other. Consider a dataset with p variables such that 2 variables are strongly correlated with each other, PCA transforms the data into a $p-1$ dimensional dataset. Its goal is to reduce the dataset as much as possible whilst retaining as much information as possible. This is referred to as 'parsimonious summarisation' of the data. In more mathematical terms, PCA reduces the dimensionality of the dataset while retaining as much of the variance as possible. Whilst a full model is responsible for 100% of the variance, those variables causing the smallest proportion of variance are discarded in PCA. Those variables that are not discarded are referred to as principal components, and are linear combinations of the original p random vectors.

It should be noted that Factor Analysis could also have been used, however, it differs from PCA as it assumes data comes from a well defined model with underlying factors

satisfying a number of assumptions. It also allows for some variance to be unexplained, which is more realistic than assuming all variance comes from the model, as PCA does. However, due to the number of assumptions required to be made, PCA is preferred. Applying this technique to the data results in thirteen principal components, for which the first two are used, being responsible for just over 60% of the variance. The other components are all accountable for less than 10% of the variance, and hence were not considered. A scree plot indicating the variance that each variable is responsible for is illustrated in figure 3.2.1.

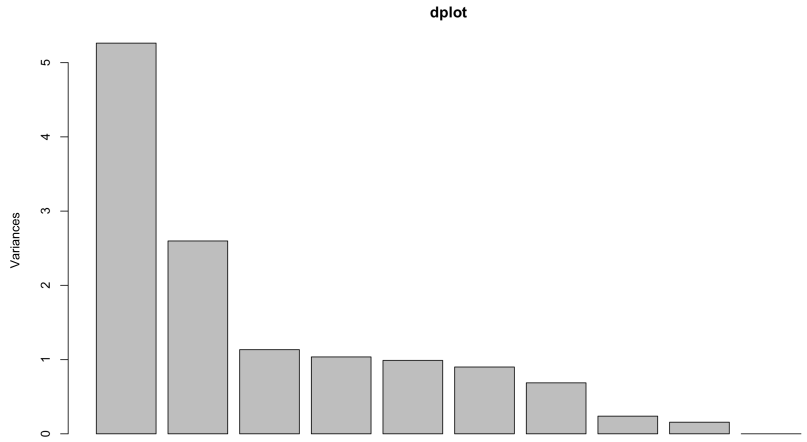

Figure 3.2.1: Scree plot of the principal components used when illustrating the data

The eigenvalue of the first principal component, which is chosen as PC1, is 5.26, while that of the second principal component, PC2, is 2.6. The third principal component had the eigenvalue 1.13, and the rest of the components all had smaller eigenvalues. The corresponding eigenvectors of the principal components contained the weights of each variable. The most influential variables in PC1 are NumberOfBets, NumberOfWins, NoOfSingleBets, LiveSingleBets and WebSingleBets. The weights assigned to these variables are 0.40, 0.39, 0.40, 0.39 and 0.34 respectively. What is evident here is that the most influential variables in this principal component are related to the amount the customer bets. On the other hand, the most influential variables in PC2 are TotalStake, TotalPayout and Profit. The weights these variables were assigned

are 0.47, 0.46 and 0.45 respectively. It should be noted that the variables that were assigned high weights in PC1 were assigned negative weights for this variable, which had an impact on the PC2 value. The weights assigned varied between -0.1 for the NumberOfWins variable, and -0.35 for the WebSingleBets variable. This indicates that the PC2 variable represents the amount a customer bet relative to the amount of bets made. That is, a high PC2 value implies that the customer bet high stakes, relative to the amount of bets made.

Below is the scatter plot obtained using the principal components, and as one can see, the data is shaped similar to a wedge. This dataset contains certain customers that stake a lot of money, but do not bet often, and vice versa. The data is illustrated in such a way that the customers that bet a large sum of money, and a high number of times, correspond to those customers towards the top right side of the plot, while the customers towards the bottom right side are those that are high volume bettors, but do not bet a significantly high enough sum of money. In addition, playing a high number of single bets on a web platform, lowers the PC2 value even more significantly. This is interesting to note as customers that bet single bets on a web platform are usually smarter players.

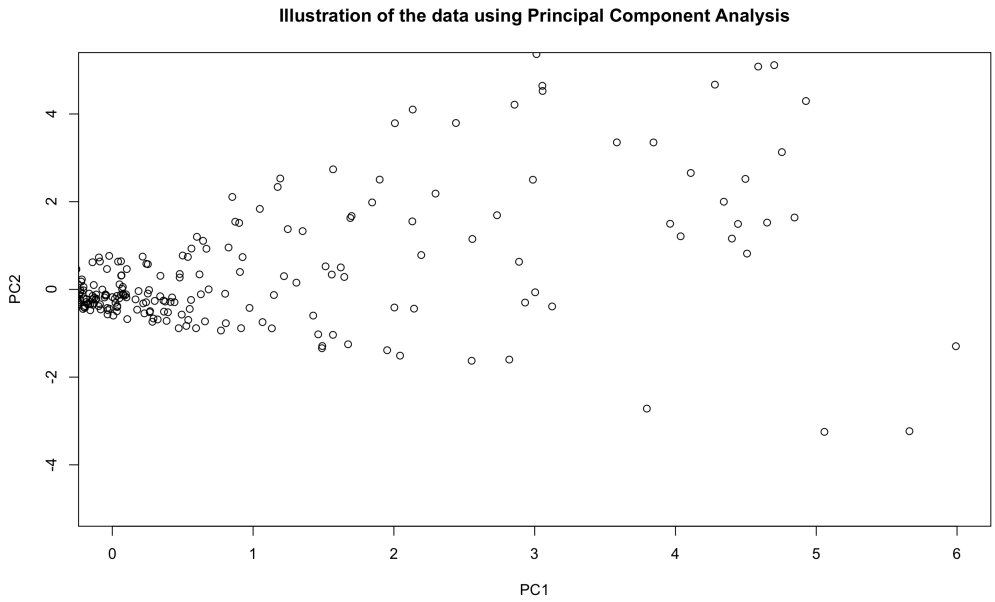

Figure 3.2.2: Scatter plot illustrating the data using Principal Component Analysis

Note that certain customers were not shown in this plot as illustrating them hindered the quality of the plot in figure 3.2.2. The illustration of the data allows us to see that there are many observations that are concentrated, which we refer to as "normal customers", and are very similar. These customers bet low stakes and bet a low amount of bets too. This is shown by the size of their PC values. There are other observations that are far more dispersed, which we shall refer to as "extreme customers". These are customers that have a particularly high value for a variable with a high weighting, for instance, number of bets, amount staked or the profit made. In fact, the dendrogram obtained when first carrying out hierarchical clustering illustrates this. Evidently, clustering cannot occur efficiently with such a high disparity in the types of customers, as the dissimilarity between an "extreme customer" and a "normal customer", is far greater than the dissimilarity between two "normal customers".

The method being carried out to eliminate the outliers is intuitive, whereby customers are clustered iteratively into two clusters. Should the number of observations in one cluster be significantly larger than the number in the second cluster, the observations are deemed to be "extreme customers" and are removed. The dendrogram is

also analysed, as this illustrates how the clustering is being carried out. This is iterated until the dendrogram illustrates that one cluster will not be overpopulated, and the clustering seems to be satisfactory. Once again, this is a completely subjective choice, and it requires a lot of trial and error to make the right choice of clusters and obtaining the desired results.

As previously mentioned, the dataset is not classified, and in this scenario we are trying to keep the number of clusters as low as possible. For this reason, and due to the way that we split the data, the smallest number of clusters we can have, without having an oversized cluster, is six clusters. Once this is done, the "extreme customers" are clustered. Clustering such an irregular dataset is not easy, and we use the trick which was used before, where customers are observed in two clusters. In this case however, customers in the smaller cluster are grouped together into their own cluster so that they will not be lost, and are addressed later on. This is repeated until the dendrogram and clustering indicate that there are no extreme observations left. The remaining observations are clustered into five clusters, which again, is a fairly intuitive number, however, was considered to give the optimal number of clusters.

Finally, these clusters are investigated, and the basic statistics of each cluster are found. The mean, maximum and minimum value for all continuous variables are found for each cluster, whilst the number of players that lost money, have a particular bet preference and have a specific platform preference is also found. Now, the number of clusters obtained after this whole process was far too great, with seventeen clusters being obtained through this whole process. We cannot proceed with so many clusters, as classification is the process of finding a balance between parsimony in the number of clusters and maintaining coherence within the clusters, and this number of clusters is definitely not parsimonious. For this reason, certain clusters were discarded or merged with others. Clusters with customers having similar characteristics were merged together, whilst a cluster containing a small number of customers, that were all quite different and could not be compared to any of the others, was removed. The clusters were investigated closely, comparing the types of customers in the two clusters and mak-

ing sure they were as similar as possible. This resulted in nine clusters, each containing different kinds of customers that are discussed further in the following section.

3.2.2 Results of Preliminary Classification

The first five clusters are made of the "normal customers", whilst the remaining clusters are made of the "extreme customers". The clusters are named as follows:

- Cluster 1: Low Value Winners
- Cluster 2: Low Value Losers
- Cluster 3: Normal Value Hunters
- Cluster 4: Normal Winners
- Cluster 5: Sharp Low Volume Players
- Cluster 6: Normal Losers
- Cluster 7: Sharp High Volume Bettors
- Cluster 8: High Roller Losers
- Cluster 9: Sharp Extreme Players

Table 3.2.1 displays the basic statistics for some key variables. This shows the average, maximum and minimum of the variables which showed the difference between clusters most clearly, and also shows the number of customers in each cluster. The average, maximum and minimum are shown to display the maximum and minimum variable values within a cluster whilst also showing what the customers were like on average. For example, in table 3.2.1, it can be seen that the lowest total stake in Cluster 8 is still a very large sum of money, indicating that the customers within this cluster bet high amounts of money. However, for the case below, there are certain customers that do not fit the cluster they were assigned to, and this affects what the table below shows. For instance, Cluster 7 has a customer that bet just 49 times, which does not

fit the definition of this cluster. This occurs because of the way cluster analysis groups the observations.

Cluster		TotalStake	TotalPayout	Bets	Wins	Profit	Singles	Combis	MobSinBets	WebSinBets	LiveSinBets	PMSinBets	Margin	CustNo
1	AVG	930.37	926.47	83.01	21.41	3.90	42.21	40.80	8.23	33.99	7.66	34.55	-0.04	289
	MAX	4,075.92	4,858.98	689.00	85.00	1,083.30	175.00	684.00	80.00	168.00	91.00	168.00	0.93	
	MIN	10.37	11.94	10.00	1.00	-1,427.94	0.00	0.00	0.00	0.00	0.00	0.00	-3.81	
2	AVG	971.87	701.11	117.30	18.30	270.76	26.52	90.78	19.17	7.35	9.18	17.34	0.35	217
	MAX	5,635.40	4,771.54	509.00	111.00	3,590.19	116.00	465.00	107.00	91.00	108.00	97.00	1.00	
	MIN	7.33	0.00	14.00	0.00	-352.87	0.00	0.00	0.00	0.00	0.00	0.00	-0.22	
3	AVG	4,207.93	3,764.34	232.33	62.97	443.59	144.72	87.61	57.72	87.00	50.06	94.67	0.12	36
	MAX	8,581.39	7,111.80	585.00	143.00	1,762.75	394.00	291.00	273.00	394.00	170.00	390.00	0.42	
	MIN	240.89	162.56	99.00	20.00	-1,860.52	1.00	0.00	0.00	0.00	0.00	1.00	-0.37	
4	AVG	4,206.46	4,780.36	161.60	61.73	-573.90	111.00	50.60	24.77	86.23	24.57	86.43	-0.13	30
	MAX	9,515.80	10,180.82	413.00	130.00	-2.63	298.00	386.00	187.00	298.00	126.00	282.00	-0.01	
	MIN	66.78	70.85	48.00	8.00	-5,418.88	0.00	0.00	0.00	0.00	0.00	0.00	-1.14	
5	AVG	3,395.43	3,496.39	45.38	19.44	-100.96	41.98	3.40	2.87	39.11	10.78	31.20	-0.14	45
	MAX	7,275.09	6,831.80	99.00	35.00	4,166.53	84.00	30.00	29.00	84.00	37.00	75.00	0.59	
	MIN	1,066.02	1,153.82	20.00	7.00	-2,464.75	19.00	0.00	0.00	14.00	0.00	12.00	-0.91	
6	AVG	2,289.19	1,941.90	481.55	91.29	347.30	130.36	351.19	17.22	113.15	56.36	74.00	0.19	176
	MAX	18,122.84	16,105.82	4,108.00	431.00	2,835.42	542.00	3,858.00	170.00	531.00	542.00	489.00	1.00	
	MIN	98.84	0.00	12.00	0.00	1.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
7	AVG	40,699.89	38,485.28	1,107.92	443.37	2,214.61	826.84	281.08	287.06	539.78	596.96	229.88	0.03	120
	MAX	345,938.85	327,396.32	5,150.00	1,602.00	35,618.90	2,846.00	3,638.00	2,651.00	2,261.00	2,816.00	1,488.00	0.41	
	MIN	94.54	81.07	49.00	24.00	-10,781.65	48.00	0.00	0.00	0.00	0.00	0.00	-0.38	
8	AVG	59,022.80	48,375.45	1,240.29	200.25	10,647.35	298.02	942.27	169.17	128.85	240.48	57.54	0.23	52
	MAX	247,254.88	221,994.10	9,367.00	793.00	30,350.29	1,381.00	8,804.00	711.00	1,281.00	1,320.00	385.00	0.53	
	MIN	7,285.35	4,935.97	123.00	8.00	726.96	1.00	1.00	0.00	0.00	0.00	0.00	0.06	
9	AVG	312,920.32	289,767.61	5,524.06	2,415.58	23,152.71	4,822.84	701.23	1,133.52	3,689.32	4,346.87	475.97	0.08	31
	MAX	1,412,481.55	1,355,634.80	40,451.00	8,830.00	75,055.52	40,451.00	5,699.00	11,821.00	40,451.00	40,451.00	6,961.00	0.18	
	MIN	722.18	687.15	719.00	288.00	35.03	48.00	0.00	0.00	0.00	0.00	0.00	0.00	

Table 3.2.1: The basic statistics of each cluster

Cluster	Number	Combi	Singles	Winner	Loser
1	289	83	206	162	127
2	217	190	27	9	208
3	36	10	26	2	34
4	30	6	24	30	0
5	45	0	45	33	12
6	176	131	45	0	176
7	120	17	103	43	77
8	52	40	12	0	52
9	31	6	25	0	31

Table 3.2.2: The type of customers in each cluster over the entire dataset

Table 3.2.2 shows the number of customers within each cluster which were classified as a certain type of player. One customer is either a combi or singles player, or a winner or loser. Combi players are named as such as they bet more combi bets than single bets, and vice versa for singles players. Meanwhile, winners have a negative profit, that is, they end up with more money than they staked, while the opposite holds for losers. For example, in clusters 6, 8 and 9 there are no winners, whilst in cluster 4 there are no losers.

Displaying the properties of each cluster through these tables, particularly table 3.2.1, does not indicate the true properties of the cluster when there are many customers that have not been clustered correctly. For this reason, one should look at the training set statistics in table 3.2.1 to show the properties of each cluster better. In the next section, each cluster will be explained in terms of what it represents and its properties. Each cluster was defined with the help of the company, where each cluster was looked at at a customer level. This was done using table 3.2.1 and studying each customer in the clusters. We will look at the clusters made of “normal customers” separately to those made of “extreme customers”.

3.2.2.1 Normal Customers

Cluster 1 is named "Low Value Winners" and is made up of customers that are not considered to be threats, as over the year they do not bet a large amount of money. This cluster is identified as a cluster of winners amongst the customers that bet fairly low stakes. The customers are not significant winners, and despite many of them making a profit, they do not bet enough money to be considered as threats. In the clustering results, there is a mix of winners and losers that tended to vary as to how much money they won or lost. Naturally, the losers did not fit the profile of this cluster. In the clustering there also were other customers that did not fit the profile of "Low Value Winners", for instance the customer that made a profit of €1,427, needs to be considered as a serious threat, and does not fit in. Similarly, a customer that bet over €4,000 does not fit within our description of this cluster, as that sum of money is far too high for him to fit the profile of this cluster.

Cluster 2, on the other hand, is made of customers that do not bet a significant amount of money, similarly to Cluster 1, but have lost money over the year. These customers are referred to as "Low Value Losers". Table 3.2.2 indicates that there were nine customers that won money over the year and these do not fit our description of this cluster. Most customers in this cluster are combi players, a trait which is associated with weaker customers. Comparing the average number of wins with the number of bets in table 5.3.2 indicates that these customers do not win many bets. Similarly, the margin of these customers is fairly high, and these players are not particularly big threats.

Cluster 3 is made of customers that are referred to as "Normal Value Hunters". This group of customers bet a moderately high amount of money and tend to vary as to how much money they win. There is a mix of players in terms of bet preference, however most customers prefer combi bets. Customers that make combi bets lose more often than those that bet single bets, however, when combi bets are won, a significant payout is won. In general, these customers tend to not be very dangerous, however, a big win could result in a massive profit for the player, so they should be monitored. There

are some customers within this cluster that do not satisfy this, as can be seen through the minimum stake in the cluster, which is only €240. Nevertheless, the average stake and maximum stake indicates to us that customers with such a low stake are in the minority; indeed there are just three customers that bet under €1000.

Cluster 4 is made of "Normal Winners" and as the name suggests, these customers bet a moderately high amount of amount of money. These players all made a profit over the time period, which is a defining characteristic of this cluster. These customers are not "Low Value Winners" as they bet far too much money to be considered as part of those customers. There are some customers with a low stake, and low winnings that do not belong in this cluster.

Cluster 5 is named "Sharp Low Volume Bettors" as it is made of players that do not bet often, however, bet high stakes. Most of these customers can be dangerous due to the amount staked per bet. These players are all singles bettors, and most players of this type are winners. The player that bet the highest number of bets bet just 99 times, whilst the highest number of wins was just 35 wins. On the other hand, the lowest stake in this cluster is €1000, which is suitable for the type of customers that we are describing. The main difference between Cluster 4 and Cluster 5 is the number of times the players bet. In addition, the margin of this cluster is quite low, indicating that these players are quite smart.

As previously explained, the structure of the data can be illustrated using Principal Component Analysis (PCA). The scree plot in figure 3.2.3 illustrates the variance each principal component is responsible for.

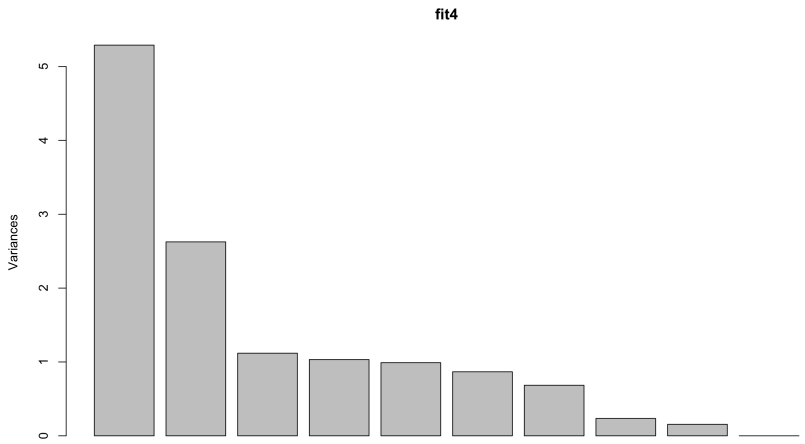

Figure 3.2.3: Scree plot of the principal components used when illustrating the data

As suspected, due to the fact that just four customers were removed when clusters were being merged and discarded, the eigenvalues returned were very similar. Indeed, the eigenvalues of the first two components were 5.29 and 2.63. Once again, the two principal components PC1 and PC2 are responsible for just over 40% and 20% of the variance respectively. The other components are each responsible for less than 9% of the variance, indicating why they were not considered. The weights assigned and proportion of variance are practically identical to those when using the entire dataset. The very slight difference in weights comes about since when clustering some customers were removed when clusters were being merged and discarded. Recall that PC1 represents the amount of bets a customer has made, whilst PC2 represents the amount a customer has spent, relative to the number of bets made. That is, a higher PC2 value indicates that the customer did bet high stakes, while make relatively few bets and vice versa. The clusters were investigated separately to get a better idea of the classification.

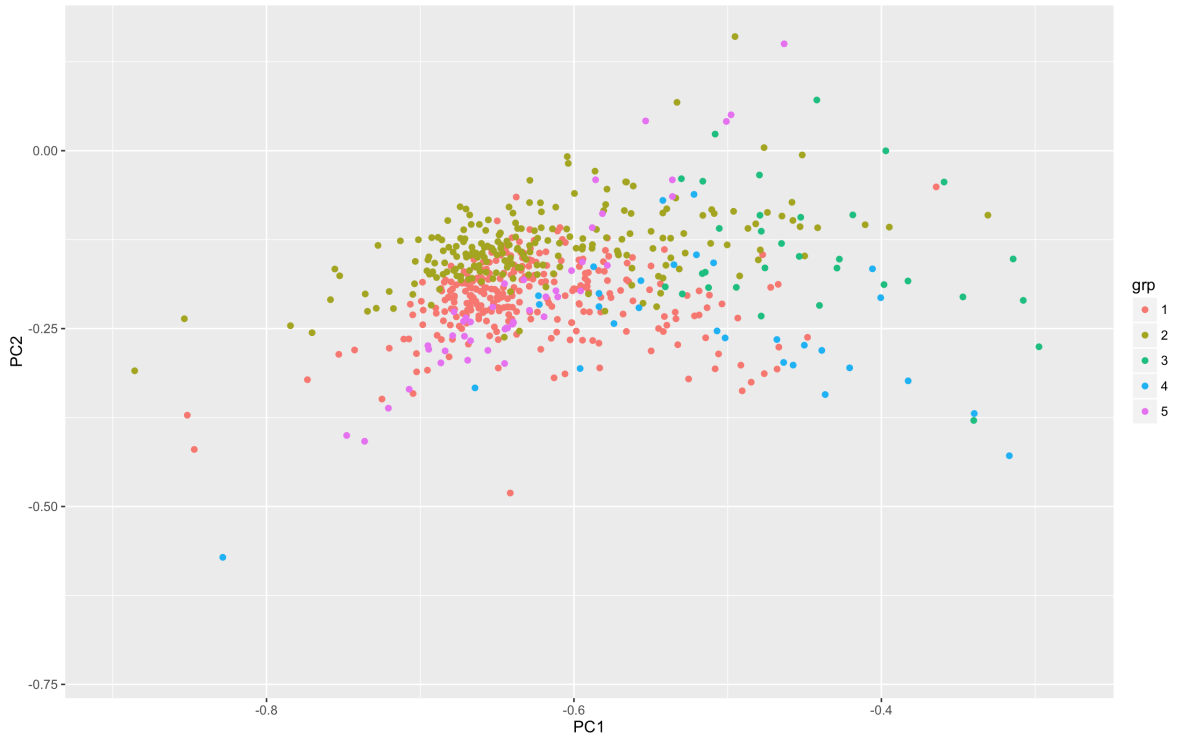

Figure 3.2.4: Illustration of Normal Customers using PCA

Figure 3.2.4 is an illustration of the datapoints which are considered to be "Normal Customers" using PCA. The datapoints are all extremely close, and the overlapping of the clusters is practically unavoidable. Cluster 1 (Low Value Winners) and Cluster 2 (Low Value Losers) have very similar plots, however the main discernible difference is that Cluster 2 has values which are higher than those of Cluster 1. This is mainly due to the fact that this cluster has customers that lose, whilst many customers in Cluster 1 are winners. The values for both clusters of PC1 are close because whilst most customers in Cluster 1 bet less than those in Cluster 2, they also win more, which due to the weights of these variables, has a balancing effect.

Both the Cluster 3 (Normal Value Hunters) and Cluster 4 (Normal Winners) datapoints are the most dispersed, which is to be expected, as they are the players which vary the most from player to player. Customers from Cluster 3 are towards the top right of the plot as they tend to bet the highest number of times and the highest stakes. They are so largely dispersed due to the varying profit in the cluster. It is no coinci-

dence that Cluster 4 has comparatively low PC2 values, this is because the profit of all the customers is negative. The observations in Cluster 5 (Sharp Low Volume Bettors) have relatively close values for PCA, mainly due to the small number of bets and wins in this cluster. On the other hand, the monetary variables in this cluster vary slightly more, particularly in terms of the profit variable. The observations in this cluster have similar PC values to those of Cluster 1, the major difference being the PC2 values. Indeed, since the customers in Cluster 5 make less bets than those in Cluster 1, and bet higher stakes, this results in a higher PC2 value.

Figure 3.2.5 illustrates the centroids of the clusters containing the "normal customers", and this plot shows the differences between the clusters further. The centroid of Cluster 2 has the highest PC2 value, and this is since the customers in this cluster are almost all losers. Only the centroid of Cluster 3 has a PC2 value close to that of Cluster 2, as these customers are also losers, but bet and win more often. Note how Cluster 4 has the lowest PC2 values, as was previously mentioned, this is due to the negative profit of the customers in this cluster. Cluster 1 and Cluster 5 have very similar centroids, despite being very different clusters. This comes about as they both bet similar amounts on average, but the difference lies in the stakes, whereby Cluster 1 has comparitively low stakes to those in Cluster 5. Note how the clusters containing many winners are towards the bottom half of the plot, whilst those consisting mostly of losers are towards the top half of the plot.

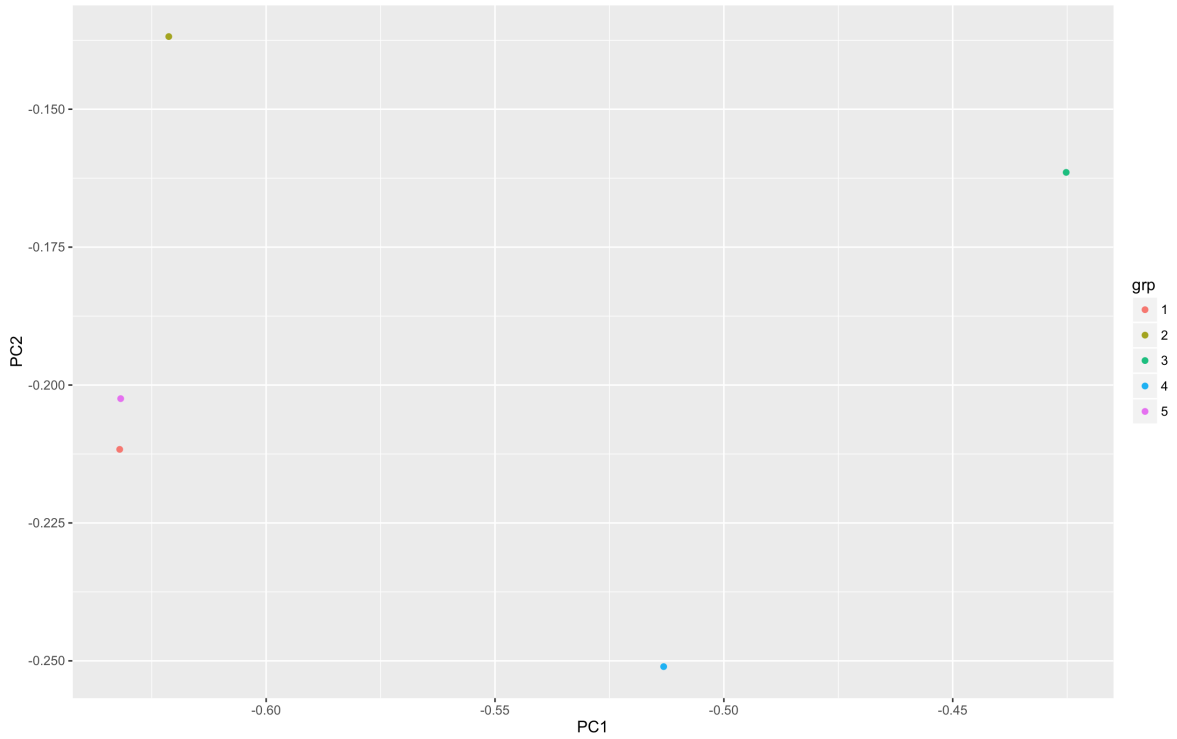

Figure 3.2.5: Illustration of centroids of clusters containing Normal Customers using PCA

3.2.2.2 Extreme Customers

Cluster 6 is made of players we refer to as "Normal Losers". These players are all losers, but vary in the amount that they bet, both in terms of the stake and the volume of bets. In this cluster there are some customers with a low stake that bet a huge amount of bets, such that they cannot possibly be classed as "normal customers". The margin of these customers is quite high. Any customer with less than 200 bets and a stake lower than €1,000 is considered to be misclassified. Additionally, these customers do not win many of their bets.

Cluster 7 consists of "Sharp High Volume Bettors". These are players that, as the name suggests, make a high number of bets. The margin of these customers is low, and they are considered to be fairly smart players. The winners within this cluster tend to win large amounts of money over the year, barring a few exceptions. Any customer with less than 500 bets was not considered to belong to this cluster.

Cluster 8 is the cluster named "High Roller Losers" as these customers do not win often and are all big losers. In fact, the lowest margin is 0.06, which is not a low margin at all. In summary, these customers bet significant sums of money, but do not bet smartly. The customer that bet the lowest sum of money bet €7,285, which is quite a high sum of money. However, this customer could also have been classed as a "Normal Loser". For this reason, we considered all customers that had a stake of at least €20,000, a significantly high sum of money, as being correctly clustered.

Cluster 9 is made of the most "extreme customers" and these customers are referred to as "Sharp Extreme Players". These customers either bet an extremely high amount of times, or bet an extremely high amount of money. The reason that these players do not form part of the previous two clusters is due to the sheer amount of investment that is made by them. These customers are all losers, however, the margin is very low, indicating that these customers may be smarter than the "High Roller Losers". Any customers that bet less than 5,000 bets or bet a lower stake than €10,000 are not considered to be part of this cluster. We illustrate the customers from these clusters using PCA in figure 3.2.6.

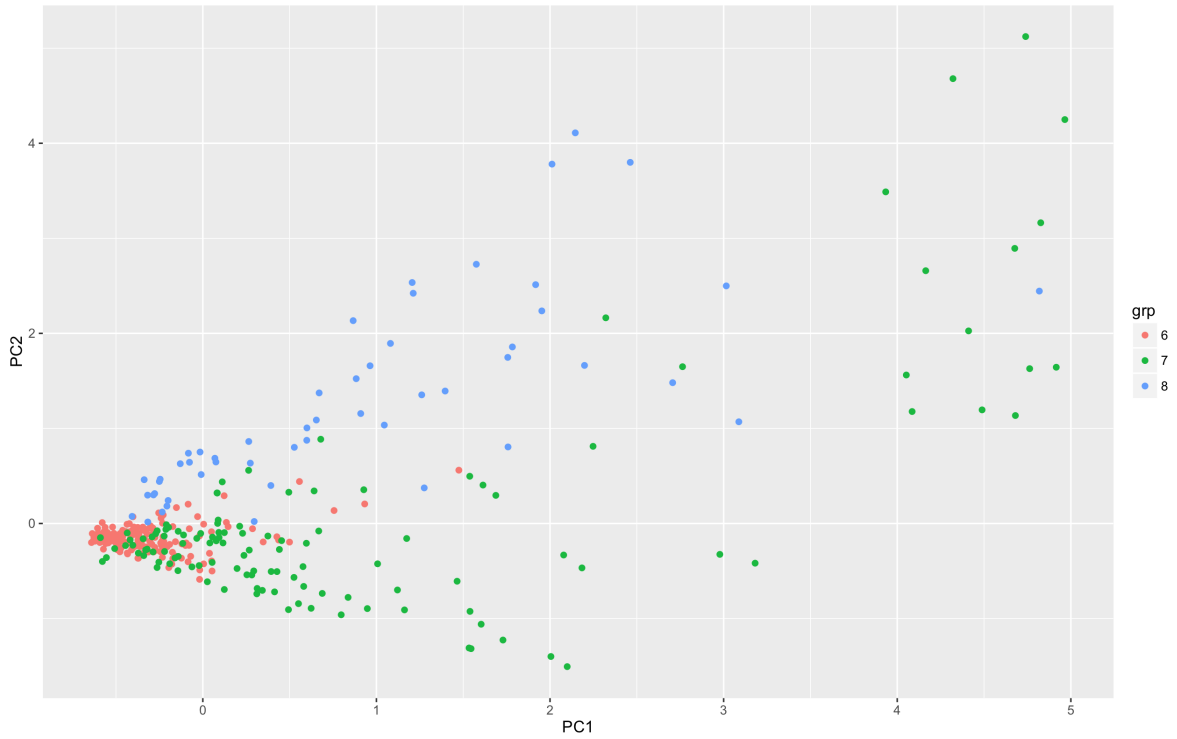

Figure 3.2.6: Illustration of Extreme Customers using PCA

The illustration of the extreme customers above indicates that the clusters 7 (Sharp High Volume Bettors) and 8 (High Roller Losers) are responsible for the more extreme customers whilst the customers in cluster 6 (Normal Losers) are less "extreme." A closer look to the clusters 6, 7 and 8 shows us the disparity in more detail. The observations of Cluster 6 are shown to be conservative compared to those from Cluster 7 and Cluster 8. Indeed, the influential variables of the principal components are all fairly low in this cluster. Note how all PC2 values of observations belonging to Cluster 8 are positive, whilst most observations of the other clusters have negative PC2 values.

Now, Cluster 7 contains many customers that bet a significant amount of times and are also winners, and this corresponds to those points that have negative PC2 values. However, there are certain customers in Cluster 7 that bet a significantly high number of bets but are also losers, and these correspond to the observations in the top right of the plot. Note that there are many customers in Cluster 7 that do not bet a high number of times, which explains why there are many customers with a negative PC1

value. It also shows that the data points of both clusters 7 and 8 vary far more than those belonging to Cluster 8.

The illustration of Cluster 9 (Sharp Extreme Players) indicates the different type of players contained within this cluster. The observations that have positive PC2 values represent customers with a high total stake and a high number of bets. On the other hand, the observations with negative PC2 values are those customers that did not have a very high stake relative to the amount of bets made. Note how far apart these data points are from those in clusters 6, 7 and 8. This is due to the sheer amount of bets and stake these customers make.

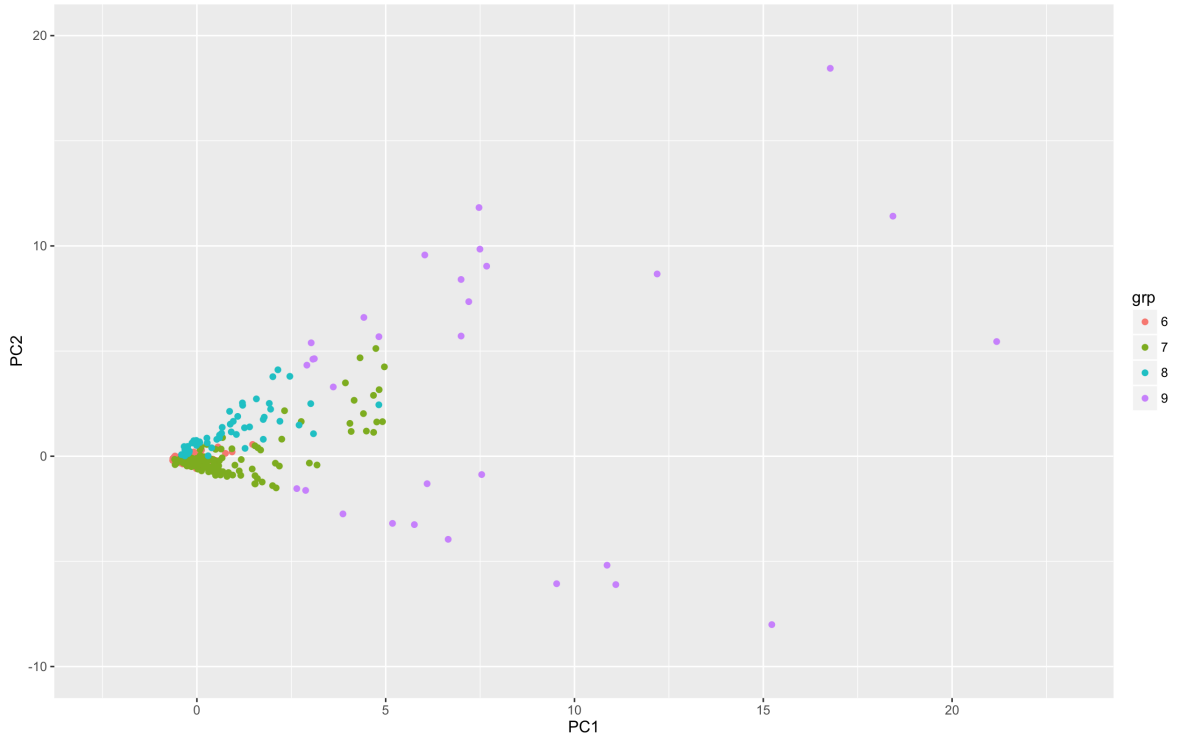

Figure 3.2.7: Illustration of observations in Cluster 9 compared to other clusters using PCA

Once again, we illustrate the centroids of the clusters in figure 3.2.8. This plot illustrates the difference in the clusters further, once again note how observations in Cluster 9 are far form those in the other clusters. Overall, customers in this cluster tend to have an extremely high stake, much higher than that of the other clusters, and a

significantly high number of bets also. Despite clusters 7 and 8 having similar a number of bets, they vary significantly in terms of stake, such that customers in Cluster 8 have much higher stakes than those in Cluster 7, as seen in the PC2 values. On the other hand, customers in Cluster 6 can be seen even more clearly to be more conservative than those in the other clusters.

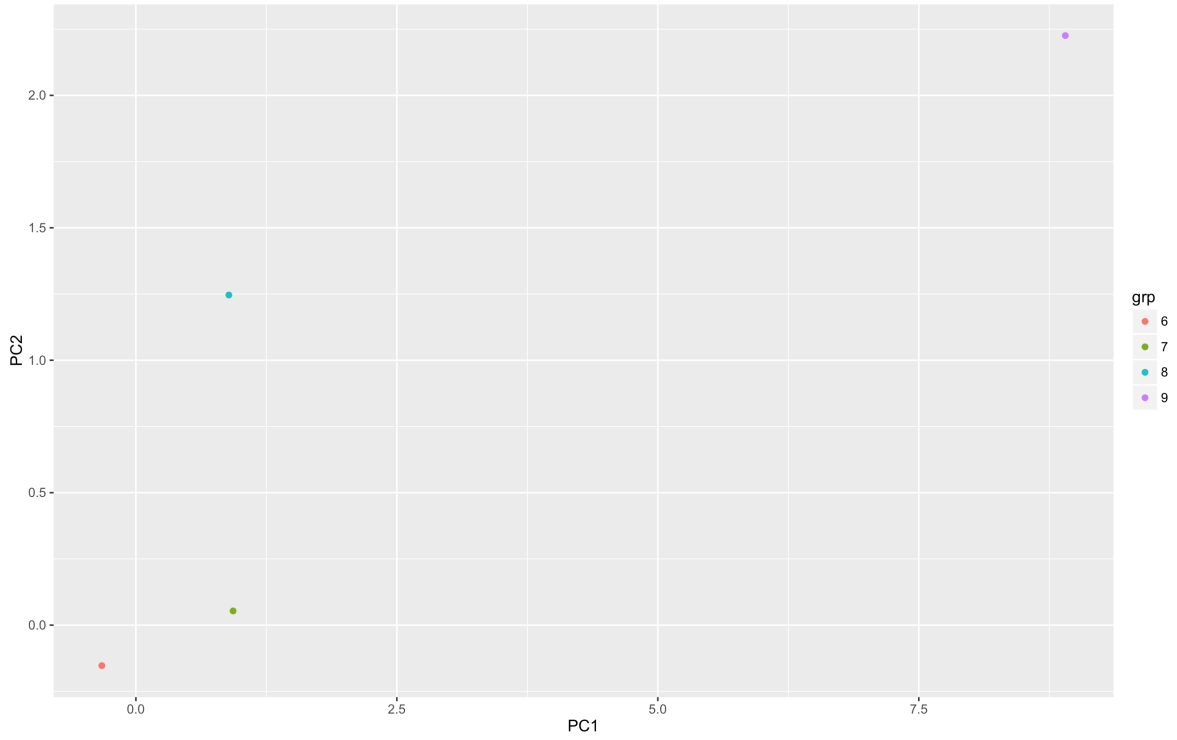

Figure 3.2.8: Illustration of centroids of clusters containing Extreme Customers using PCA

3.3 Linear Regression

Sir Francis Galton studied the relationship between the height of parents and children, and noted that the children tended to buck the trend of their parents, that is, they were smaller if their parents were large, and larger if their parents were small. The experiment indicated that relatively extreme values tend to be followed by values which regress to the mean. This is why the term regression was used to describe the statistical tool with which we find an estimate for the relationship among variables. This is done by finding a function which best represents the relationship among the variables. A general linear model is considered to be a flexible generalization of ordinary linear regression. It considers response variables denoted Y_i with error distributions other than the normal distribution. However, in this section, we will be discussing linear regression, for which the error term $\boldsymbol{\varepsilon}$ is normally distributed and homoscedasticity is satisfied:

$$\boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}_n) \tag{3.3.1}$$

which allows us to obtain more general results without unnecessary complications. The vector of parameters is taken to be$\boldsymbol{\beta}$, whilst the dataset matrix is denoted by $\mathbb{X}$, with entry X_{ij} representing the j^{th} parameter of the i^{th} observation.

In linear regression we find the linear model, that is, a linear function which provides the best fit given the data points. One entry of $\mathbf{Y}$ is expressed as follows:

$$Y_i = \mathbf{X}_i{}^T \boldsymbol{\beta} + \varepsilon$$

In mathematical terms, we are finding the conditional expectation of a random vector $\mathbf{Y}$, given the value of some input denoted $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_p)$ influenced by the vector of parameters $\mathbf{w}$. This is expressed as

$$\mathbb{E}[\mathbf{Y}|(\mathbf{X}_1, \ldots, \mathbf{X}_p) = (\mathbf{x}_1, \ldots, \mathbf{x}_p)] = \mathbb{X}^T \boldsymbol{\beta} \tag{3.3.2}$$

The results (3.3.1) and (3.3.2) are used to show that the conditional distribution is expressed as $\mathbf{Y}|\mathbb{X} \sim \mathcal{N}_n(\mathbb{X}^T \boldsymbol{\beta}, \sigma^2 \mathbf{I})$.

There are numerous method of estimating the parameter vector $\boldsymbol{\beta}$. Maximum Likelihood Estimation requires the distribution of the response variables Y_i, while the Least Squares method does not, however it does require additional assumptions on Y_i. There is no significant advantage to using the Least Squares Method. Note that the Maximum Likelihood Estimation can be used as long as the distribution of the error terms is known, and this need not be normal. However, as we are assuming normality, the estimated parameters are identical to the Least Square Estimators.

Linear regression is a technique which forms the base of classification in Gaussian Processes. This technique was developed so as to include the Bayesian approach and this was further generalized to be able to apply Gaussian regression, which we shall discuss in Chapter 5. The results of Gaussian classification follow from those of Gaussian regression, however, with certain adjustments that are necessary. Note that linear regression could be used for classification purposes, however it is very limited and far less powerful than a tool such as Gaussian classification. Due to the nature of linear regression, it is very unlikely that satisfactory results will be obtained using this method.

Chapter 4

Useful Tools in Classification

This chapter provides a brief overview to certain topics which are essential to Gaussian Process Classification. As will be discussed later on, Bayesian Inference forms the basis of the topic, so a quick revision of the area is useful. Similarly, Decision Theory is vital to understanding the way classification works. For this reason it will also be discussed, and also giving a summary of Bayesian Decision Theory, to remain along the lines of Bayesian theory.

4.1 Bayesian Inference

Statistical inference is defined as a theory supporting method from which one can make conclusions about a population by considering samples drawn from the population. It can be applied in two different philosophies, one method called the classical approach, also known as the frequentist approach, and the other known as the Bayesian approach. The first step in statistical analysis is always to construct a probabilistic model. The notion of probability in both approaches is significantly different. The classical approach has the more common form of probability, known as objective probability. This is loosely defined as the probability that an event will occur based on a measure for which a recorded observation is considered. On the other hand, the frequentist approach can include probability which measures a person's degree of belief of an event occurring, known as subjective probability. Whereas the frequency of an event matters in objective probability, this is not the case for subjective probability. Subjective probability takes

into account some prior knowledge of the event or lack of it.

The following is the general setup for statistical inference, and is the same for both approaches. In the classical approach the parameter vector $\mathbf{W}$ is made of fixed, unknown values, hence there is no probability involved. On the other hand, in the frequentist approach we take the parameter vector to be a random variable $\mathbf{W}$. The probability mechanism that generates a probability distribution in the parameter space will be supported by the probability space $(\mathbf{\Omega_W}, \mathcal{F}_\mathbf{W}, \mathbb{P}_\mathbf{W})$. Consider a sequence of independent, identically distributed random variables $Y_1, \ldots, Y_N$ which are mappings from the sample space $\mathbf{\Omega}_s$ to a value in $\mathbb{R}$. The process of selecting a sample $\mathbf{Y} = (Y_1, \ldots, Y_N)$ from the existing population generates a probability mechanism which generates a probability distribution in the sample space supported by the probability space defined as $(\mathbf{\Omega}_s, \mathcal{F}_s, \mathbb{P}_s)$. $\mathbf{\Omega}_m$ is the parameter space when sampling from a distribution with parameters $\mathbf{W}$ and represents the model with its own probability space, defined as $(\mathbf{\Omega}_m, \mathcal{F}_m, \mathbb{P}_m)$. Technically, the underlying probability space should be the product of the above three spaces.

The Bayesian approach incorporates some prior knowledge about the parameters, the prior distribution denoted as $f(\mathbf{w})$, a distribution over the parameter space $\mathbf{\Omega_W}$. This distribution is subjective, that is, there is no one right prior distribution. The likelihood distribution, denoted $f(\mathbf{y}|\mathbf{W} = \mathbf{w})$, represents the data and is used to improve the prior. It represents all the information given by the data about the parameter $\mathbf{w}$. This is derived using Bayes' Rule, which is crucial in this approach. Consider the definition of a conditional distribution for a discrete random variable $\mathbf{Y}$:

$$
\begin{aligned}
f\left(\mathbf{y}, \mathbf{w}|\mathbf{X} = \mathbb{X}\right) &= f(\mathbf{w}|\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbb{X}) f(\mathbf{y}) \\
f\left(\mathbf{w}|\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbb{X}\right) &= \frac{f(\mathbf{y}, \mathbf{w}|\mathbf{X} = \mathbb{X})}{f(\mathbf{y})} \qquad (4.1.1) \\
f\left(\mathbf{w}|\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbb{X}\right) &= \frac{f(\mathbf{y}, \mathbf{w}|\mathbf{X} = \mathbb{X})}{\sum_\mathbf{w} f(\mathbb{X}|\mathbf{W} = \mathbf{w}) f(\mathbf{w})} \qquad (4.1.2)
\end{aligned}
$$

The denominator of equation (4.1.2) is said to be the marginal likelihood. For a

continuous random variable, (4.1.2) is expressed as:

$$f\left(\mathbf{w}|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbb{X}\right)=\frac{f(\mathbf{y},\mathbf{w}|\mathbf{X}=\mathbb{X})}{\int f(\mathbb{X}|\mathbf{W}=\mathbf{w})f(\mathbf{w})d\mathbf{w}}$$

This integrates the parameter $\mathbf{w}$ out from the likelihood equation, making use of the fact that the probability $f(\mathbf{w})$ is known. The marginal likelihood is denoted by:

$$f(\mathbf{y}|\mathbf{X}=\mathbb{X})=\int f(\mathbf{y}|\mathbf{X}=\mathbb{X},\mathbf{W}=\mathbf{w})f(\mathbf{w})d\mathbf{w}$$

In general, consider the random vector $\mathbf{Y}=\mathbf{y}$, with free variables $\mathbf{u}$ and $\mathbf{u}'$:

$$\begin{aligned} f(\mathbf{u}|\mathbf{Y}=\mathbf{y}) &= \frac{f(\mathbf{u},\mathbf{y})}{\int f(\mathbf{u}',\mathbf{y})d\mathbf{u}'} \\ f(\mathbf{u},y) &= f(\mathbf{u}|Y=y)\cdot\int f(\mathbf{u}',y)d\mathbf{u}' \end{aligned}$$

The above are important rules in probability and are used in conjunction with Bayes Rule to give the posterior expressed as:

$$f(\mathbf{w}|\mathbf{X}=\mathbb{X},\mathbf{Y}=\mathbf{y})=\frac{f(\mathbf{y}|\mathbf{X}=\mathbb{X},\mathbf{W}=\mathbf{w})f(\mathbf{w})}{f(\mathbf{y}|\mathbf{X}=\mathbb{X})}$$

4.2 Decision Theory

Decision Theory is a critical tool in classification, the basic idea of which is to find an optimal decision which minimizes the expected loss. Note that we are denoting the observed data by the known vector $\mathbf{X} = \mathbf{x}$, which represents one observation. Define the decision function $\alpha : \mathbb{R}^d \mapsto \mathcal{A}$ such that it maps the input $\mathbf{x}$ to an output $\gamma_i \in \mathcal{A}$, where $\mathcal{A}$ is said to be the class of decision rules such that $\mathcal{A} = \{\gamma_1, \ldots, \gamma_a\}$, known as the hypothesis class.

A decision rule divides the feature space, which is the space where the variables exist, into regions denoted $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_c$ such that when a vector falls into a specific region it is classified to that class. Naturally, there are cases when a vector falls onto two or more classes. In this case, a reject option is used, which states that if the probability of $\mathbf{x}$ being in the class it's most likely to be classified in is greater than the threshold probability τ then the vector can be classified into it. This applies to binary classification, however this method can also be applied to multiclass classification by considering the difference in probabilities of the most likely class and the second most likely class. Should this exceed τ then the vector can be classified to the most probable class. If these thresholds are not satisfied, then classification is conducted by a more complex tool.

Define $l(\alpha(\mathbf{x}), y)$ to be the loss when the decision $\alpha(\mathbf{x})$ is made but the correct decision is $Y = y$, where Y is categorical. There are various formulas for the loss function, and each are tailored for specific situations. The most popular loss function is the zero-one loss function, as it is the most basic:

$$l(\alpha(\mathbf{x}), y) = \begin{cases} 1 & if\ \alpha(\mathbf{x}) \neq y \\ 0 & if\ \alpha(\mathbf{x}) = y \end{cases} \tag{4.2.1}$$

Whilst this function is very straightforward, it is not always correct. Consider the situation that a car has a fault or is suspected to have a fault. The loss incurred when the car is suspected to have a fault but does not should not be the same as the loss

when the car is not suspected to have a fault but does.

The expected loss of selecting a class $\alpha(\mathbf{x})$ is the product of the loss which occurs in the case of a misclassification and the probability of a misclassification occurring. This is said to be the risk function which we define for discrete values of $\mathbf{x}$ in equation (4.2.2) and for continuous values of $\mathbf{x}$ in equation (4.2.3):

$$R\left(\alpha(\mathbf{x})\right) = \sum_{y,\mathbf{x}} l\left(\alpha(\mathbf{x}), y\right) \mathbb{P}(y, \mathbf{x}) \tag{4.2.2}$$

$$R\left(\alpha(\mathbf{x})\right) = \sum_{y} \int l\left(\alpha(\mathbf{x}), y\right) f(y, \mathbf{x}) d\mathbf{x} \tag{4.2.3}$$

Consider the decision rule γ_i which satisfies $R(\gamma_i) \leq R(\alpha(\mathbf{x}))\ \forall \alpha(\mathbf{x}) \in \mathcal{A}$. This would mean that the decision γ_i is always the correct decision to make, however, this is rarely the case. Consider a rule γ_j such that:

$$\begin{aligned} R(\alpha(\mathbf{x})) &\leq R(\gamma_j)\ \forall \alpha(\mathbf{x}) \in \mathcal{A} \\ R(\gamma_t) &< R(\gamma_j) \end{aligned}$$

This means that the rule γ_j is always at least as bad as the decision $\alpha(\mathbf{x})$ for all parameters, and is strictly worse for some other decision γ_t. The decision rule $\alpha(\mathbf{x})$ is said to dominate the rule γ_j, hence γ_j is said to be inadmissible. Considering only admissible rules lessens the number of feasible rules significantly, however, this is still not enough. To address this problem, we will consider the Bayesian approach.

4.2.1 Bayesian Decision Theory

The Bayesian approach, discussed in the previous section, can be applied to decision theory. It addresses the decision problem in a probabilistic sense, such that certain outcomes are more likely than others. This tells us that the decision making relies on the prior $f(\mathbf{w})$ and likelihood $f(\mathbb{X}|\mathbf{W} = \mathbf{w})$, which are combined through the Bayesian

approach discussed in the previous section. The posterior expected loss in this case is:

$$R\left(\gamma_i|\mathbf{x}\right) = \sum_{c=1}^{n} l\left(\gamma_i, \mathbf{w}_c\right) f\left(\mathbf{w}_c|\mathbb{X} = \mathbf{x}\right)$$

The goal is to minimize the expected risk under the posterior, and this is done using the expectation of the risk, defined as:

$$\mathbb{E}[R(\gamma_i|\mathbf{x})] = \int f(\mathbf{w}) R(\gamma_i|\mathbf{x}) d\mathbf{w}$$

where $f(\mathbf{w})$ is the prior distribution. Bayes' decision theory tells us to select the decision $\hat{\gamma}$ which minimizes the risk, which, as we can see, is dependent on the posterior probability $f(\mathbf{w}|\mathbb{X} = \mathbf{x})$:

$$\begin{aligned} \hat{\gamma} &= \arg\min_{\gamma_i \in \mathcal{A}} \left[R\left(\gamma_i|\mathbf{x}\right)\right] \\ &= \arg\min_{\gamma_i \in \mathcal{A}} \int l\left(\gamma_i, \mathbf{w}\right) f\left(\mathbf{w}|\mathbb{X} = \mathbf{x}\right) d\mathbf{w} \end{aligned} \tag{4.2.4}$$

This is applicable if the equation(4.2.4) can be computed and if the data $\mathbb{X}$ with outputs $\mathbf{Y}$ are independently, identically distributed as $f(\mathbb{X}, \mathbf{y}|\mathbf{W} = \mathbf{w})$. When $Y \in \{-1, 1\}$ or $Y \in \{1, \ldots, n\}$ we have a situation of classification, whilst when $Y \in (-\infty, \infty)$ we have regression.

Chapter 5

Gaussian Processes

Once the clustering has been carried out, we will look at Gaussian classification as a more complex mathematical object, which has infinitely more structure and elements to capture correlations and ways of separating large sets. Gaussian classification requires an already classified dataset and obtains results using this dataset. As the dataset was not already classified, the results of cluster analysis, covered in Chapter 3, were used as somewhat of a stepping stone to obtaining results using Gaussian classification. The results from cluster analysis were crucial to carrying out Gaussian classification. This chapter covers the Gaussian Processes part of the dissertation. The topics in Chapter 4, which are the Bayesian approach and Decision theory, are both necessary as a basis to this chapter, particularly the Bayesian approach. This chapter involves a brief introduction to Gaussian Processes, as well as to Reproducing Kernel Hilbert Spaces. These are essential topics to Gaussian Process classification, which is covered in the final part of this chapter. In addition, Gaussian Process regression is also covered as its results are essential to Gaussian Process classification.

5.1 Introduction to Gaussian Processes

Formally, a Gaussian Process G_t is a collection of random variables, indexed by $t \in T$, where T is any general set, such that for any finite subset $F \subseteq T$ the random variable $\{G_t\}_{t \in F}$ has a joint multivariate Gaussian distribution. This is a powerful property as it allows us to work with samples of an infinite dimensional process. Con-

ditioning and marginalization of Gaussians give Gaussians again, a very useful feature to have when dealing with high dimensions. These properties allow the application of simple linear algebra to higher dimensions. The covariance matrix of a multivariate Gaussian random vector is a symmetric, non-negative definite matrix. When the covariance function is strictly positive definite, we say that the Gaussian Process is non-degenerate. The mean vector and covariance function determine a Gaussian distribution uniquely, and consequently the process is uniquely determined by the mean function denoted $\mathbb{E}[G_t] = m(t)$ and the covariance function denoted $Cov\,(G_s, G_t) = \mathbb{E}\left[(G_s - \mathbb{E}[G_s])\,(G_t - \mathbb{E}[G_t])\right] = K(s,t)$ where $s, t \in T$.

Gaussian Processes can be used to carry out supervised learning, a technique belonging to machine learning that involves deducing values of some function at certain points given others. The result produced is continuous or discrete, such that a continuous output is obtained when applying regression, while classification results in a categorical output. Usually such processes take an index of the set $\mathbb{R}_+$, however, by using Mercer's Theorem we can expand this to replace the index set by T, which generalizes this idea to various applications.

It should be noted that not all Gaussian processes have continuous paths. Marcus and Shepp [21][22] show that for some conditions, the paths of Gaussian Processes are not continuous. Indeed, a lot of research was put into finding necessary and sufficient conditions on the covariance function that would ensure continuity. Lalley explains [17] how it can be shown that some Gaussian Processes have continuous sample functions.

This is done through two steps. Firstly, recall that Gaussian Processes with countable index sets can be constructed from independent, identically distributed unit normals. Let $K(s,t)$ be a positive definite function indexed by a countable set T. Without loss of generality, assume that $T = \mathbb{N}$. Let F_m be a probability distribution on $\mathbb{R}^m$ where $m = 1, 2, \ldots$and assume for each Borel subset B of $\mathbb{R}^m$ that $F_m(B) = F_{m+1}(B \times \mathbb{R})$. It is to be shown that on some probability space the random variables G_i are defined such that $\forall m\ (G_1, \ldots, G_m) \sim F_{m+1}$. Using the Inverse Probability Integral Transform

(Section 7.2), there is a function $\varphi_1(Y_1) = G_1$ with distribution F_1. Suppose G_i are constructed for $i \leq m$ using Y_i such that the joint distribution of the random variables $(G_i)_{i \leq m}$ is F_m.

Now, let G_{m+1} be the conditional distribution on $\mathbb{R}$ of the $(m+1)^{st}$ parameter of an F_{m+1} distributed random variable given $(G_i)_{i \leq m}$. Y_{m+1}, which is the corresponding $(m+1)^{st}$ continuous variable, is then used to construct this random variable G_{m+1}. The consistency property allows us to deduce that $(G_i)_{i \leq m+1} \sim F_{m+1}$. Then the sequence of distributions $F_m = \mathcal{N}(0, \Sigma_m)$ is mutually consistent by Kolmogorov's Consistency Theorem, and hence, by the above, there exists a sequence of random variables G_i such that for each finite subset $F \subset T$ we have $G_F \sim \mathcal{N}(0, \Sigma_F)$. There are a number of examples of both discrete and continuous covariance functions that provide no centred Gaussian Processes from which we can obtain continuous sample paths. However, under certain constraints, the covariance function can provide a Gaussian Process that can be extended continuously from a countable dense index set to a continuum, which is the set of real numbers. This is the final step of showing that some Gaussian Processes have continuous sample functions.

5.1.1 Reproducing Kernel Hilbert Spaces

Consider a centred, non-degenerate Gaussian Process $\{G_t\}_{t \in T}$ with covariance function $K(s,t)$, and let L^2_G be the closure, relative to the L^2 norm of the vector space made of all finite linear combinations of G_t. Since the L^2 limits of centred Gaussian random variables are also random variables, this means that every random variable in the closure L^2_G is Gaussian. The Reproducing Kernel Hilbert space associated with $\{G_t\}_{t \in T}$ is a Hilbert Space made of real valued functions on the set T which is naturally isomorphic to L^2_G.

Definition: A function $K : E \times E \mapsto \mathbb{C}$ is said to be a reproducing kernel of the Hilbert Space $\mathcal{H}$, where E is a nonempty abstract set, if and only if $\forall t \in E$ and

$\forall \varphi \in \mathcal{H}$

$$\begin{aligned} K(\cdot, t) &\in \mathcal{H} \\ < \varphi, K(\cdot, t) > &= \varphi(t) \end{aligned}$$

A Hilbert Space of complex valued functions which possess a reproducing kernel is called a Reproducing Kernel Hilbert space.

The second condition is said to be the reproducing property, as it tells us that the value of φ at t can be reproduced by the inner product of φ and $K(\cdot, t)$. To explain the construction of the RKHS better, consider another isometry, the Wiener Integral. Wiener considered that if a centred Gaussian process $\{W_t\}_{t\in[0,1]}$ with the covariance function $\mathbb{E}[W_s W_t] = \min(s, t)$ then for $s < t$ and $u < v$:

$$\mathbb{E}\left[(W_t - W_s)\,(W_v - W_u)\right] = \int_0^1 \mathbf{1}_{(s,t]}(x)\mathbf{1}_{(u,v]}(x)dx$$

Since both the expectation and Lebesgue integral are linear, the above equation extends to all finite linear combinations. Now, the linear combinations of indicator functions such as $\mathbf{1}_{(a,b]}$ are step functions, which are dense relative to the L^2-distance, so the linear mapping

$$T\left(\sum_{i=1}^{k} a_i \mathbf{1}_{(s_i,t_i]}\right) = \sum_{i=1}^{k} a_i (W(t_i) - W(s_i))$$

extends to a linear mapping from $L^2[0, 1]$ to $L^2(P)$. This comes about by the following proposition:

Proposition: Let H_o be a dense, linear subspace of a Hilbert space H and let $J : H_o \mapsto H'$ be a linear isometry mapping H_o into another Hilbert space H'. Then J extends uniquely to a linear isometry $J : H \mapsto H'$.

Proof: See [17]

In our case we are taking $L^2[0,1]$ as H_o and $L^2(P)$ as H since the linear combinations of step functions are dense. Consider the countable orthonormal basis ψ_n and the Wiener isometry T. Wiener showed that if $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space that supports the sequence of independent, identically distributed normal random variable $\{\xi_n\}_{n\geq 1}$, then the mapping $T\psi_n = \xi_n$ extends to a linear isometry of $L^2[0,1]$ into $L^2(P)$.

Let us now look at the Reproducing Kernel Hilbert space. A symmetric, real valued function $K : T \times T \mapsto \mathbb{R}$ is said to be a positive definite kernel if for every finite subset $F \subset T$ then $(K(s,t))_{s,t\in F}$ is positive definite. The covariance function of all non-degenerate Gaussian Process satisfies this property. Furthermore, every positive definite kernel R on T induces a metric on T defined as $d(s,t) = \sqrt{R(t,t) + R(s,s) - 2R(s,t)}$. The Moore-Aronszajn Theorem below tells us that every symmetric, positive definite kernel defines a unique RKHS:

Definition: A positive type function is a function $K : E \times E \mapsto \mathbb{C}$ such that

$$\forall n \geq 1,\ \forall\,(a_1, \ldots, a_n) \in \mathbb{C}^n,\ \forall\,(x_1, \ldots, x_n) \in E^n,$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i \bar{a}_j K\left(x_i, x_j\right) \in \mathbb{R}^+$$

Theorem: Let K be a positive type function on $E \times E$. There exists only one Hilbert Space $\mathcal{H}$ of functions on E with reproducing kernel K. The subspace $\mathcal{H}_o$ of $\mathcal{H}$ spanned by the functions $K(\cdot, x)_{x\in E}$ is dense in $\mathcal{H}$ and $\mathcal{H}$ is the set of functions on E which are pointwise limits of Cauchy sequences in $\mathcal{H}_o$ with the inner product

$$\left\langle \sum_{i=1}^{n} \alpha_i K(\cdot, x_i), \sum_{j=1}^{m} \beta_j K(\cdot, y_j) \right\rangle_{\mathcal{H}_o} = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \bar{\beta}_j K(y_j, x_i)$$

Proof: See [4]

Definition: The Reproducing Kernel Hilbert space H associated with the covariance kernel K is the closure of H_o with respect to the norm induced by the inner product

in the above theorem.

Lemma: If T is separable relative to the canonical metric, then so is the RKHS H, and if T is compact then $\{K(s,t)\}_{s\in T}$ is also compact, and hence bounded in H.

Proof: See [17]

The statement below is known as Mercer's Theorem, which tells us that if T is compact, this ensures that the RKHS is separable, resulting in every orthonormal basis being countable. Now, if T is finite, then the expansion in the theorem follows from the spectral theorem for finite symmetric matrices, so we restrict our attention to a compact, but infinite T.

Theorem: Let K be a continuous kernel on the compact metric space T, and let v be a finite Borel measure on T with $\text{supp}(v) = T$. Then $\forall x, y \in T$

$$K(s,t) = \sum_{j\geq 1} \lambda_j e_j(x) e_j(y)$$

and the convergence of the sum is uniform on $T \times T$ and absolute $\forall (x,y) \in T \times T$.

Proposition: If T is compact relative to the canonical metric then on any probability space that supports a sequence $\{\xi_n\}_{n\geq 1}$ of independent $\mathcal{N}(0,1)$ random variables there exists a centred Gaussian Process $\{G_t\}_{t\in T}$ with covariance function K. For any orthonormal basis $\{e_j\}_{j\geq 1}$ and any $t \in T$ the random variable X_t is the almost sure limit of

$$G_t = \sum_{n=1}^{\infty} \xi_n e_n(t)$$

Proof: See [17]

5.2 Regression

The linear approach to regression described in Chapter 3 considers only linear functions, and while this is easy to work with it reduces the number of possible functions significantly, which is not necessarily a good thing. Linear regression finds the line which provides the best fit, given the points $\mathbb{X}$, which is a far less sophisticated method than that carried out by Gaussian Processes, which offer a far more flexible approach, as they contain all possible continuous functions. Consider the number of functions which are only required to pass through the data points and be continuous. There is an infinite number of functions that satisfy this. A Bayesian approach is used to narrow down the number of possible functions to represent the model.

In the first section of this subchapter we discuss solving a regression problem using the Bayesian approach, and following this we consider mapping the input to a higher dimension and find similar results with the input set to a higher dimension. The next section explains Bayesian regression using the previous two sections.

5.2.1 Bayesian Regression

Classical approaches such as Maximum Likelihood Estimation and Least Squares Estimation are commonly used to estimate the parameters of a regression model. Another method of finding the parameters is the Bayesian approach. The Bayesian approach, as explained in Chapter 4, combines observed data with prior knowledge, which is in the form of a density function. Consider Bayesian analysis of the standard linear model with Gaussian noise:

$$Y = g(\mathbf{x}) + \varepsilon$$

$$\mathbb{E}[Y|\mathbb{X} = \mathbf{x}] = g(\mathbf{x})$$

where $g(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$ is the function vector when $\mathbb{X} = \mathbf{x}$, and $\mathbb{X}$ is the regressor, also known as the input matrix. These assumptions, along with those made above allow us

to express the likelihood as follows:

$$\begin{aligned} f(\mathbf{y}|\mathbb{X}=\mathbf{x},\mathbf{W}=\mathbf{w}) &= \prod_{i=1}^{N} f(y_i|\mathbf{X}_i=\mathbf{x}_i,\mathbf{W}=\mathbf{w}) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y_i-\mathbf{x}_i^T\mathbf{w})^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\sum_{i=1}^{N}(y_i-\mathbf{x}_i^T\mathbf{w})^2}{2\sigma^2}\right) \quad (5.2.1) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left\|\mathbf{y}-\mathbb{X}^T\mathbf{w}\right\|}{2\sigma^2}\right) = \mathcal{N}\left(\mathbb{X}^T\mathbf{w},\sigma^2 I\right) \quad (5.2.2) \end{aligned}$$

where $\|\mathbf{a}\|$ is the Euclidean length of the vector $\mathbf{a}$.

Since we are carrying out a Bayesian approach, a prior is specified over the unknown parameter vector $\mathbf{W}$:

$$\mathbf{W} \sim \mathcal{N}_p(\mathbf{0}_p, \mathbf{\Sigma}_p) \quad (5.2.3)$$

which is multivariate normal with mean assumed to be $\mathbf{0}_p$ and variance $\mathbf{\Sigma}_p$. Assuming that the mean function is $\mathbf{0}$ means that positive values are not preferred to negative values, and vice versa. Inference in the Bayesian approach is based on the posterior distribution $f(\mathbf{w}|\mathbf{Y}=\mathbf{y},\mathbb{X})$, obtained through the likelihood, prior and marginal distribution, as explained in Chapter 1. The likelihood distribution is obtained from the data and expressed as $f(\mathbf{y}|\mathbb{X}=\mathbf{x},\mathbf{W}=\mathbf{w})$. The marginal likelihood is denoted by

$$f(\mathbf{y}|\mathbb{X}) = \int f(\mathbf{y}|\mathbb{X}=\mathbf{x},\mathbf{W}=\mathbf{w})f(\mathbf{w})d\mathbf{w}$$

The above is used to give the posterior expressed as:

$$\begin{aligned} f(\mathbf{w}|\mathbb{X},\mathbf{Y}=\mathbf{y}) &= \frac{f(\mathbf{y}|\mathbb{X}=\mathbf{x},\mathbf{W}=\mathbf{w})f(\mathbf{w})}{f(\mathbf{y}|\mathbb{X})} \quad (5.2.4) \\ &= \frac{f(\mathbf{y}|\mathbb{X}=\mathbf{x},\mathbf{W}=\mathbf{w})f(\mathbf{w})}{\int f(\mathbf{y}|\mathbb{X}=\mathbf{x},\mathbf{W}=\mathbf{w})f(\mathbf{w})d\mathbf{w}} \end{aligned}$$

Considering only the terms dependent on $\mathbf{w}$ in (5.2.4), the posterior is simplified:

$$f(\mathbf{w}|\mathbb{X}, \mathbf{Y} = \mathbf{y}) \quad \alpha \quad \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbb{X}^T\mathbf{w})^T(\mathbf{y} - \mathbb{X}^T\mathbf{w})\right] \cdot \exp\left[-\frac{1}{2}\mathbf{w}^T\boldsymbol{\Sigma}_p^{-1}\mathbf{w}\right]$$

$$f(\mathbf{w}|\mathbb{X}, \mathbf{Y} = \mathbf{y}) \quad \alpha \quad \exp\left[-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T\left(\frac{1}{\sigma^2}\mathbb{X}\mathbb{X}^T + \boldsymbol{\Sigma}_p^{-1}\right)(\mathbf{w} - \bar{\mathbf{w}})\right] \tag{5.2.5}$$

$$f(\mathbf{w}|\mathbb{X}, \mathbf{Y} = \mathbf{y}) \quad \alpha \quad \exp\left[-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T\mathbb{A}(\mathbf{w} - \bar{\mathbf{w}})\right] \tag{5.2.6}$$

where $\bar{\mathbf{w}} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\mathbb{X}\mathbb{X}^T + \boldsymbol{\Sigma}_p^{-1}\right)^{-1}\mathbb{X}\mathbf{y}$ and $\mathbb{A} = \frac{1}{\sigma^2}\mathbb{X}\mathbb{X}^T + \boldsymbol{\Sigma}_p^{-1}$. Equation (5.2.6) is obtained from (5.2.2) and (5.2.3), and it is of the form of the Gaussian distribution with mean $\bar{\mathbf{w}}$ and covariance matrix $\mathbb{A}^{-1}$, hence:

$$\mathbf{w}|\mathbb{X}, \mathbf{Y} = \mathbf{y} \sim \mathcal{N}_n\left(\frac{1}{\sigma^2}\mathbb{A}^{-1}\mathbb{X}\mathbf{y}\ ,\ \mathbb{A}^{-1}\right) \tag{5.2.7}$$

The distribution of the output value Y_*, given a test input $\mathbf{X}_* = \mathbf{x}_*$, training input $\mathbb{X} = \mathbf{x}$ and output $\mathbf{Y} = \mathbf{y}$, is obtained by averaging over all the outputs of the possible linear models with respect to the Gaussian posterior obtained in (5.2.6):

$$f(Y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}) = \int f(y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbf{W} = \mathbf{w})f(\mathbf{w}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})d\mathbf{w}$$

Recall that above we found that (5.2.7) while the distribution of $f(y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbf{Y} = \mathbf{y})$ is equivalent to that given in (5.2.2)

$$y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbb{X}, \mathbf{Y} = \mathbf{y} \sim \mathcal{N}\left(\frac{1}{\sigma^2}\mathbf{x}_*^T A^{-1}\mathbb{X}\mathbf{y}\ ,\ \mathbf{x}_*^T A^{-1}\mathbf{x}_*\right) \tag{5.2.8}$$

The predictive distribution is once again Gaussian with mean equal to $\mathbf{x}_*$ multiplied by the mean of $f(\mathbf{w}|\mathbb{X}, \mathbf{Y} = \mathbf{y})$. Similarly, the variance of the prediction distribution involves a quadratic form of the test input multiplied by the covariance matrix of the posterior, implying that as the test input increases in magnitude, so does the variance.

5.2.2 Gaussian Process Regression

In this section, inference is applied directly in the function space, which should give us the same results as above. Recall that a Gaussian Process is said to be completely specified by its mean function $m(\mathbf{x})$ and its covariance function $K(\mathbf{x}, \mathbf{x}')$. Denote the Gaussian Process as $G_{\mathbf{x}}$ such that:

$$m(\mathbf{x}) = \mathbb{E}[G_{\mathbf{x}}] \tag{5.2.9}$$

$$K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(G_{\mathbf{x}} - m(\mathbf{x}))(G_{\mathbf{x}'} - m(\mathbf{x}'))] \tag{5.2.10}$$

Stochastic Processes are defined over the time index set, that is $t \in \mathbb{R}_+$, however, in this scenario this is not the index set we are interested in. Since the index set is the d-dimensional input $\mathbf{x} \in \mathbb{R}^d$, this process is said to be a random field. Mercer's Theorem, mentioned previously, allows us to generalize this index set to be represented by any set T, in this chapter.

The definition of a Gaussian process implies consistency, also known as the marginalization property. This property implies that given that:

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{5.2.11}$$

then the marginal distribution of $\mathbf{y}_1$ is $\mathbf{y}_1 \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{1}, \boldsymbol{\Sigma}_{11})$. This is shown through the Useful Theorems found in Section 7.1.

5.2.2.1 Derivation of the parameters

We can create a Gaussian process from the linear model $\phi(\mathbf{x})^T\mathbf{W}$, where $G_{\mathbf{x}} = \phi(\mathbf{x})^T\mathbf{W}$ such that the vector $\mathbf{W}$ is normally distributed with mean $\mathbf{0}$ and covariance Σ_p. Note that the theory explaining embedding inputs to higher dimensions is discussed in the Appendix in Section 7.3. Using equations (5.2.9) and (5.2.10) we can find the mean

$m(\mathbf{x})$ of the Gaussian Process:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[G_{\mathbf{x}}] \\ &= \mathbb{E}[\phi(\mathbf{x})^T \mathbf{W}] \\ &= \phi(\mathbf{x})^T \mathbb{E}[\mathbf{W}] \\ &= 0 \end{aligned}$$

and the covariance function $K(\mathbf{x}, \mathbf{x}')$:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(G_{\mathbf{x}} - m(\mathbf{x}))(G_{\mathbf{x}'} - m(\mathbf{x}'))] \\ &= \mathbb{E}[(G_{\mathbf{x}} - 0)(G_{\mathbf{x}'} - 0)] \\ &= \mathbb{E}[(G_{\mathbf{x}'})(G_{\mathbf{x}'})] \\ &= \mathbb{E}[(\phi(\mathbf{x})^T \mathbf{w})(\phi(\mathbf{x}')^T \mathbf{w})^T] \\ &= \phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}\mathbf{w}^T] \phi(\mathbf{x}') \\ &= \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}') \end{aligned}$$

The derivations above indicate that $G_{\mathbf{x}}$ and $G_{\mathbf{x}'}$ are jointly Gaussian. A commonly used covariance function is the squared exponential covariance:

$$Cov(G_{\mathbf{x}}, G_{\mathbf{x}'}) = K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\frac{||\mathbf{x} - \mathbf{x}'||^2}{l^2}\right) \tag{5.2.12}$$

where l represents the length scale of the function, which affects the smoothness of the function. The covariance function controls the form of the Gaussian Process, and any changes to it have a significant effect on the process.

5.2.2.2 Noiseless Outputs of Gaussian Regression

We need to withdraw random functions from the prior to consider information about the function that fits our data best. The joint distribution of the training outputs $\mathbf{H}$ and test outputs $\mathbf{H}_*$, according to the prior, is

$$\begin{bmatrix} \mathbf{H} \\ \mathbf{H}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{x},\mathbf{x}) & K(\mathbf{x},\mathbf{x}_*) \\ K(\mathbf{x}_*,\mathbf{x}) & K(\mathbf{x}_*,\mathbf{x}_*) \end{bmatrix}\right) \tag{5.2.13}$$

To obtain the posterior over the functions we need to restrict the joint prior distribution to contain only functions which agree with the observed data points. In particular, consider the value $\mathbf{H} = \mathbf{h}$. We condition the joint Gaussian prior distribution on the observations such that the conditional distribution of the new output is:

$$\mathbf{h}_*|\mathbf{x}_*,\mathbf{x},\mathbf{h} \sim \mathcal{N}\left(K(\mathbf{x}_*,\mathbf{x})K(\mathbf{x},\mathbf{x})^{-1}\mathbf{h}\,,\; K(\mathbf{x}_*,\mathbf{x}_*) - K(\mathbf{x}_*,\mathbf{x})K(\mathbf{x},\mathbf{x})^{-1}K\left(\mathbf{x},\mathbf{x}_*\right)\right) \tag{5.2.14}$$

This is derived from the following theorem:

Theorem: Consider the general case with a dataset D made of two subvectors $\mathbf{a}$ and $\mathbf{b}$ where given that $D \sim \mathcal{N}(\mathbf{0}, V)$ we have

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} A & C^T \\ C & B \end{bmatrix}\right)$$

such that A, B and C make up the matrix V. The distribution of $\mathbf{b}$ given $\mathbf{a}$ is also Gaussian distributed, by the properties of the distribution. If D is a block diagonal matrix such that $C = \mathbf{0}$ then $\mathbf{b}|\mathbf{a} \sim \mathcal{N}(\mathbf{0}, B)$, however in the case that this is not so, then

$$\mathbf{b}|\mathbf{a} \sim \mathcal{N}(CA^{-1}\mathbf{a}, B - CA^{-1}C^T)$$

5.2.2.3 Noisy Outputs of Gaussian Regression

Above it was assumed that there is no noise in the observations. In this section, we consider a model which includes noisy observations, and hence is a more general model. A noisy linear model is of the form:

$$Y = \mathbf{X}^T\mathbf{w} + \varepsilon$$

with the assumption that the error terms are independent and identically distributed as $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, and letting $\mathbf{W} = \mathbf{w}$. The covariance function is adjusted to include the noise:

$$Cov(Y_p, Y_q) = K(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2\delta_{pq}$$

such that

$$\delta_{pq} \begin{cases} 1 & p = q \\ 0 & otherwise \end{cases}$$

We include the noise term in (5.2.13) and denote the new noise training outputs $\mathbf{Y}$ and noisy test outputs $\mathbf{Y}_*$ to obtain:

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{Y}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbb{X},\mathbb{X}) + \sigma_n^2 I & K(\mathbb{X}, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathbb{X}) & K(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right)$$

Note the difference in the distribution of the noisy observations $\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, K(\mathbb{X},\mathbb{X}) + \sigma_n^2 I)$ and that of the noiseless observations $\mathbf{H} \sim \mathcal{N}(\mathbf{0}, K(\mathbb{X},\mathbb{X}))$. Deriving the conditional distribution in a similar fashion to equation (5.2.14) and taking $\mathbf{Y} = \mathbf{y}$ we obtain the following predictive distribution:

$$\mathbf{y}_*|\mathbb{X}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_* \sim \mathcal{N}\left(\bar{\mathbf{y}}_*\,,\, Cov(\mathbf{y}_*)\right) \tag{5.2.15}$$

where

$$\bar{\mathbf{y}}_* = K(\mathbf{x}_*, \mathbb{X})\left[K(\mathbb{X},\mathbb{X}) + \sigma_n^2 I\right]^{-1}\mathbf{y} \tag{5.2.16}$$

$$Cov(\mathbf{y}_*) = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbb{X}) \left[K(\mathbb{X}, \mathbb{X}) + \sigma_n^2 I\right]^{-1} K(\mathbb{X}, \mathbf{x}_*) \tag{5.2.17}$$

Note how $K(\mathbb{X}, \mathbb{X})$ has been replaced with $K(\mathbb{X}, \mathbb{X}) + \sigma_n^2 I$. Notice also how the variance of the distribution depends only on the inputs, not the observed term. The variance is the difference between the prior covariance and the information about the function given by the data and is a useful property of Gaussian processes.

5.3 Gaussian Process Classification

5.3.1 Introduction

Having discussed regression, we now turn our attention to classification. The obvious key difference between these two techniques is the output, which is continuous for regression, but discrete for classification. Classification problems are harder to deal with, as for regression it was assumed that the likelihood function is Gaussian, due to a Gaussian prior and likelihood, but this is not the case for classification. Due to the outputs being discrete, a Gaussian likelihood is not appropriate.

Classification problems are approached in two different ways, named the generative approach and the discriminative approach. The generative approach involves finding the joint distribution $f(\mathbf{x}, y)$. It is the more complete method, as it models the class of conditional distributions $f(\mathbf{x}|Y = y)$, such that Y represents the classes $(\theta_1, \theta_2, \ldots, \theta_C)$, and the prior probabilities for each class, then computes the probability for the class θ_c using Bayes' Rule:

$$f(\theta_c|\mathbf{X} = \mathbf{x}) = \frac{f(\theta_c)f(\mathbf{x}|Y = \theta_c)}{\sum_{j=1}^{C} f(\theta_j)f(\mathbf{x}|Y = \theta_j)} \tag{5.3.1}$$

The class with the highest probability is selected, which gives a full probabilistic model of all the involved variables. The discriminative approach, on the other hand, models $f(y|\mathbf{X} = \mathbf{x})$ directly, and is far simpler to work with and less computationally demanding. The approach is chosen according to the type of problem being faced. Vapnik [35] claims: "One should solve the problem directly and never solve a more general problem as an intermediate step." In the generative approach a more general problem is being solved as an intermediate step to obtain $f(y|\mathbf{X} = \mathbf{x})$. In this section the discriminative approach shall be considered, for which we can convert the output of a regression model to a class probability by making use of the response function. This is carried out using the inverse of the link function, and compresses the function to have values which range from 0 to 1.

The most common example of a response function is the logistic function:

$$f(y|\mathbf{X}=\mathbf{x}) = S(\mathbf{x}^T\mathbf{w}) \tag{5.3.2}$$

where $S(z) = \frac{1}{1+e^{-z}}$

Another type of response function is the probit function, which utilizes the cumulative standard normal distribution function.

For now, we shall consider binary classification, taking into account Gaussian Processes, to provide a clear description of what shall be done. A Gaussian Process prior is used to obtain the latent function, denoted $H(\mathbf{X})$, and this has its output converted using the sigmoid function to obtain a prior on $f(Y=+1|\mathbf{X}=\mathbf{x}) = S(H(\mathbf{x}))$. A latent function is defined as a function that is not directly observed, but is derived from other functions. It is used to act as a link between observable and symbolic data, and is often useful to reduce the dimensionality of the data. In Gaussian Process Classification, one must first find the distribution of a test latent variable, then using this distribution find the conditional expected value. The distribution of the latent variable in the test case is expressed as:

$$f\left(h_*|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y},\mathbf{X}_*=\mathbf{x}_*\right) = \int f(h_*|\mathbb{X}=\mathbf{x},\mathbf{X}_*=\mathbf{x}_*,\mathbf{H}=\mathbf{h}) f(\mathbf{h}|\mathbb{X},\mathbf{Y}=\mathbf{y}) d\mathbf{h} \tag{5.3.3}$$

where $f\left(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y}\right)$ is the posterior over latent variables. The predictive distribution is obtained using the above:

$$f\left(Y_*=+1|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y},\mathbf{X}_*=\mathbf{x}_*\right) = \int S(H_*) f\left(H_*|\mathbb{X}=\mathbf{x},\mathbf{y},\mathbf{x}_*\right) f\left(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{y}\right) dH_* \tag{5.3.4}$$

In the regression scenario this technique was fairly straightforward, as the integrals considered were Gaussian and could be evaluated with no complications. However, in the case of classification, recall that Y is discrete, hence we do not have the same luxury.

Thus, the probability $f\left(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y}\right)$ cannot be evaluated. Therefore, the integral in (5.3.4) also cannot be evaluated for certain sigmoid functions. An approximation is required to compute the integrals, which approximate the non-Gaussian joint posterior with a Gaussian joint posterior. In this dissertation we will explain the Laplace Approximation of the non-Gaussian posterior.

5.3.2 Laplace Approximation

The Laplace Approximation is a technique used to approximate integrals of the form $\int_a^b \exp\left[Mg(x)\right]dx$ where M is any large number. This method is used to approximate the posterior, which is analytically intractable. The approximation of $f\left(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y}\right)$ in (5.3.3) is denoted as as $q(\mathbf{h}|\mathbf{x},\mathbf{y})$, which is a Gaussian probability. This can be expressed using a second order Taylor expansion of $\log\left[f\left(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y}\right)\right]$ about the maximum of the posterior $f\left(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y}\right)$ which is denoted as $\hat{\mathbf{h}}$. Consider the Taylor expansion of a function $g: A \mapsto B$ about $\mathbf{x}_0$:

$$\ln\left[g(\mathbf{x})\right] = \ln\left[g(\mathbf{x}_0)\right] + \nabla\ln\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\nabla\nabla\ln\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \ldots$$

Assuming that higher order terms are negligible:

$$\ln\left[g(\mathbf{x})\right] = \ln\left[g(\mathbf{x}_0)\right] + g(\mathbf{x})^{-1}\nabla\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\nabla\nabla\ln\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0)$$

Suppose $\mathbf{x}^* = \arg\max_x\left\{g(\mathbf{x}) : \mathbf{x}\in A\right\}$ then taking $\mathbf{x}_0 = \mathbf{x}^*$ gives $\nabla\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}_0} = 0$:

$$\begin{aligned}
\ln\left[g(\mathbf{x})\right] &\approx \ln\left[g(\mathbf{x}^*)\right] + \frac{1}{2}(\mathbf{x}-\mathbf{x}*)^T\nabla\nabla\ln\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}*}(\mathbf{x}-\mathbf{x}*) \\
\exp\left[\ln\left[g(\mathbf{x})\right]\right] &\approx \exp\left[\ln\left[g(\mathbf{x}^*)\right] + \frac{1}{2}(\mathbf{x}-\mathbf{x}*)^T\nabla\nabla\ln\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}*}(\mathbf{x}-\mathbf{x}*)\right] \\
\int\exp\left[\ln\left[g(\mathbf{x})\right]\right]d\mathbf{x} &\approx \exp\left[\ln\left[g(\mathbf{x}^*)\right]\right]\int\exp\left[\frac{1}{2}(\mathbf{x}-\mathbf{x}*)^T\nabla\nabla\ln\left[g(\mathbf{x})\right]\big|_{\mathbf{x}=\mathbf{x}*}(\mathbf{x}-\mathbf{x}*)\right]d\mathbf{x}
\end{aligned}$$

Let $L(\mathbf{x}) = \ln[g(\mathbf{x})]$:

$$\begin{aligned}\int \exp[L(\mathbf{x})]\, d\mathbf{x} &\approx \exp[L(\mathbf{x}^*)] \int \exp\left[\frac{1}{2}(\mathbf{x}-\mathbf{x}*)^T \nabla\nabla L(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}*}(\mathbf{x}-\mathbf{x}*)\right] d\mathbf{x} \\ \int e^{L(\mathbf{x})} d\mathbf{x} &\approx e^{L(\mathbf{x}^*)} \int \exp\left[\frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^T L''(\mathbf{x}^*)^{-1}(\mathbf{x}-\mathbf{x}^*)\right] d\mathbf{x}\end{aligned}$$

Suppose $\sigma^2 = -\frac{1}{L''(\mathbf{x}^*)}$:

$$\int e^{L(\mathbf{x})} d\mathbf{x} \approx e^{L(\mathbf{x}^*)} \int \exp\left[-\frac{(\mathbf{x}-\mathbf{x}^*)^2}{2\sigma^2}\right] d\mathbf{x}$$

Notice the similarities between the final integral and the Gaussian density functions. This means that the posterior can be approximated using the distribution:

$$q(\mathbf{x}) = \mathcal{N}(\mathbf{x}^*, \sigma^2)$$

Applying the above approximation to our situation, for which we replace $g(\mathbf{x})$ by $f(\mathbf{h}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y})$ such that the approximation now becomes:

$$q(\mathbf{h}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y}) = \mathcal{N}(\hat{\mathbf{h}}, \mathbb{A}^{-1})$$

which is of the form $\exp\left[-\frac{1}{2}\left(\mathbf{h}-\hat{\mathbf{h}}\right)^T \mathbb{A}\left(\mathbf{h}-\hat{\mathbf{h}}\right)\right]$ such that $\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmax}}\left[f(\mathbf{h}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y})\right]$ and $\mathbb{A} = -\nabla\nabla \log\left[f(\mathbf{H}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y})\right]\big|_{\mathbf{H}=\hat{\mathbf{h}}}$ which is the Hessian of the negative log posterior at the maximum $\hat{\mathbf{f}}$.

5.3.2.1 Approximation of Parameters

In this section the method of obtaining $\hat{\mathbf{h}}$ and $\mathbb{A}$ is described, which are necessary to find the approximate posterior $q(\mathbf{h}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y})$. Consider $f(\mathbf{h}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y})$ expressed using Bayes' Rule:

$$f(\mathbf{h}|\mathbb{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y}) = \frac{f(\mathbf{y}|\mathbf{H}=\mathbf{h}) f(\mathbf{h}|\mathbb{X}=\mathbf{x})}{f(\mathbf{y}|\mathbb{X}=\mathbf{x})} \tag{5.3.5}$$

Now, the interest is in finding the maximum value of $\mathbf{H}$, hence we shall maximize (5.3.5). Notice how the denominator $f(\mathbf{y}|\mathbb{X})$ is independent of $\mathbf{H}$, which means that when maximizing it can be ignored as it has no effect on $\mathbf{H}$. Consider the logarithm of the above equation and define $\Psi(\mathbf{h}) = \log\left[f\left(\mathbf{h}|\mathbb{X}, \mathbf{Y} = \mathbf{y}\right)\right]$:

$$\begin{aligned}
\Psi(\mathbf{h}) &\stackrel{c}{=} \log[f(\mathbf{y}|\mathbf{H} = \mathbf{h})] + \log[f(\mathbf{h}|\mathbb{X})] \\
\Psi(\mathbf{h}) &\stackrel{c}{=} \log[f(\mathbf{y}|\mathbf{H} = \mathbf{h})] - \frac{1}{2}\mathbf{h}^T K(\mathbb{X}, \mathbb{X})^{-1}\mathbf{h} \qquad (5.3.6) \\
&\quad -\frac{1}{2}\log|K(\mathbb{X}, \mathbb{X})| - \frac{n}{2}\log(2\pi) \qquad (5.3.7)
\end{aligned}$$

using a result derived from regression of Gaussian Processes which states:

$$\log\left[f(\mathbf{h}|\mathbb{X} = \mathbf{x})\right] = -\frac{1}{2}\mathbf{h}^T K(\mathbf{x}, \mathbf{x})^{-1}\mathbf{h} - \frac{1}{2}\log|K(\mathbf{x}, \mathbf{x})| - \frac{n}{2}\log(2\pi)$$

Now to obtain the maximum $\hat{\mathbf{h}}$, equation (5.3.6) must be differentiated with respect to $\mathbf{h}$:

$$\begin{aligned}
\nabla\Psi(\mathbf{h}) &= \nabla\log[f(\mathbf{y}|\mathbf{H} = \mathbf{h})] - K(\mathbf{x}, \mathbf{x})^{-1}\mathbf{h} \\
\nabla\nabla\Psi(\mathbf{h}) &= \nabla\nabla\log[f(\mathbf{y}|\mathbf{H} = \mathbf{h})] - K\left(\mathbf{x}, \mathbf{x}\right)^{-1} = -W - K(\mathbf{x}, \mathbf{x})^{-1}
\end{aligned}$$

such that $W = -\nabla\nabla\log\left[f(\mathbf{y}|\mathbf{H} = \mathbf{h})\right]$, which is diagonal. The negative of the inverse of $\nabla\nabla\Psi(\mathbf{h})$ gives the variance of the approximate posterior $q\left(\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}\right)$.

If $\log\left[f(\mathbf{y}|\mathbf{H} = \mathbf{h})\right]$ is concave, then this implies that the elements of W are non-negative, since by concavity $\nabla\nabla\log[f(\mathbf{y}|\mathbf{H} = \mathbf{h})]$ is negative definite, and that $\Psi(\mathbf{h})$ is log concave and hence has a unique maximum. Two commonly used likelihoods are the logistic and probit likelihood, which were mentioned previously. Consider the logistic likelihood:

$$
\begin{aligned}
f\left(y_i|H_i=h_i\right) &= \frac{1}{1+e^{-y_i h_i}} \\
\log\left[f\left(y_i|H_i=h_i\right)\right] &= -\log\left[1+e^{-y_i h_i}\right] \\
\frac{\partial}{\partial h_i}\log\left[f\left(y_i|H_i=h_i\right)\right] &= \frac{y_i}{\left(1+e^{-y_i h_i}\right)e^{y_i h_i}} \\
\frac{\partial^2}{\partial h_i^2}\log\left[f\left(y_i|H_i=h_i\right)\right] &= -\frac{y_i^2 e^{y_i h_i}}{\left(1+e^{y_i h_i}\right)^2}
\end{aligned}
$$

To find the maximum value of $\Psi(\mathbf{h})$, let $\nabla\Psi(\mathbf{h})=0$, which occurs at $\mathbf{h}=\hat{\mathbf{h}}$:

$$
\begin{aligned}
\nabla\log[f(\mathbf{y}|\mathbf{H}=\hat{\mathbf{h}})] &= K(\mathbf{x},\mathbf{x})^{-1}\hat{\mathbf{h}} \\
K(\mathbf{x},\mathbf{x})\cdot(\nabla\log[f(\mathbf{y}|\mathbf{H}=\hat{\mathbf{h}})]) &= \hat{\mathbf{h}} \qquad (5.3.8)
\end{aligned}
$$

Since $\nabla\log\left[f(\mathbf{y}|\mathbf{H}=\mathbf{h})\right]$ is a non-linear function, Newton's method is required to approximate the maximum of Ψ:

$$
\begin{aligned}
\mathbf{h}^{'} &= \mathbf{h}-\left(\nabla\nabla\Psi\left(\mathbf{h}\right)\right)^{-1}\left(\nabla\Psi\left(\mathbf{h}\right)\right) \\
&= \mathbf{h}+(K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1}(\nabla\log[f(\mathbf{y}|\mathbf{H}=\mathbf{h})]-K(\mathbf{x},\mathbf{x})^{-1}\mathbf{h}) \\
&= \mathbf{h}+(K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1}\nabla\log[f(\mathbf{y}|\mathbf{H}=\mathbf{h})]-(K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1}K(\mathbf{x},\mathbf{x})^{-1}\mathbf{h} \\
&= (K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1}\nabla\log[f(\mathbf{y}|\mathbf{H}=\mathbf{h})]+(I-(K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1}K(\mathbf{x},\mathbf{x})^{-1})\mathbf{h}) \\
&= (K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1}[\nabla\log[f(\mathbf{y}|\mathbf{H}=\mathbf{h})]+W\mathbf{h}]
\end{aligned}
$$

The above is iterated until the approximate maximum $\hat{\mathbf{h}}$ is found to be satisfactory. The Laplace approximation to the posterior is then denoted as:

$$
q(\mathbf{h}|\mathbb{X}=\mathbf{x},\mathbf{Y}=\mathbf{y})=\mathcal{N}(\hat{\mathbf{h}},(K(\mathbf{x},\mathbf{x})^{-1}+W)^{-1})
$$

5.3.2.2 Conditional Expectation

The conditional expected value of a new output H_*, derived from the posterior $q(\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$, under the Laplace approximation is derived using (5.2.16) and (5.3.8):

$$\begin{aligned} \mathbb{E}_q[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*] &= K(\mathbf{x}_*, \mathbf{x})^T K(\mathbf{x}, \mathbf{x})^{-1} \hat{\mathbf{h}} \\ &= K(\mathbf{x}_*, \mathbf{x})^T \nabla_{\hat{\mathbf{h}}} \log \left[f\left(\mathbf{y}|\hat{\mathbf{h}}\right) \right] \end{aligned} \tag{5.3.9}$$

The exact mean, on the other hand, as given by Opper and Winther (2000) is:

$$\begin{aligned} \mathbb{E}_p[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*] &= \int \mathbb{E}\left[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right] f\left[\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}\right] d\mathbf{h} \\ &= \int K(\mathbf{x}_*, \mathbf{x})^T K(\mathbf{x}, \mathbf{x})^{-1} \mathbf{h} f\left[\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}\right] d\mathbf{h} \\ &= K(\mathbf{x}_*, \mathbf{x})^T K\left(\mathbf{x}, \mathbf{x}\right)^{-1} \mathbb{E}\left[\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}\right] \end{aligned} \tag{5.3.10}$$

which is obtained via $\mathbb{E}[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*] = K(\mathbf{x}_*, \mathbf{x})^T K(\mathbf{x}, \mathbf{x})^{-1}\mathbf{h}$. Notice that equations (5.3.9) and (5.3.10) are similar, however, the approximate mean replaces $\mathbb{E}\left[\mathbf{H}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}\right]$ with $\hat{\mathbf{h}}$.

The conditional variance of H_* is also obtained using the approximate posterior and applying the Matrix Inversion Lemma. Note that $K(\mathbf{x}, \mathbf{x})$ has been denoted as $\mathbb{K}$, while $\mathbb{K}_*$ represents $K(\mathbf{x}_*, \mathbf{x})$ and $\mathbb{K}_{**}$ represents $K(\mathbf{x}_*, \mathbf{x}_*)$:

$$\begin{aligned} Var_q\left[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right] &= \mathbb{E}[(H_* - \mathbb{E}[H_*|\mathbb{X} = \mathbf{x}, \mathbf{x}_*, \mathbf{h}])^2] \\ &\quad + \mathbb{E}[\mathbb{E}[H_*|\mathbb{X} = \mathbf{x}, \mathbf{x}_*, \mathbf{h}] - \mathbb{E}[H_*|\mathbb{X} = \mathbf{x}, \mathbf{y}, \mathbf{x}_*])^2] \\ &= \mathbb{K}_{**} - {\mathbb{K}_*}^T \left(\mathbb{K}^{-1} - \mathbb{K}^{-1} \left(\mathbb{K}^{-1} + \mathbb{W} \right)^{-1} \mathbb{K}^{-1} \right) K_* \\ &= \mathbb{K}_{**} \\ &\quad - {\mathbb{K}_*}^T \left(\mathbb{K}^{-1} - \mathbb{K}^{-1} \left(\mathbb{K} - \mathbb{K} \left(\mathbb{W}^{-1} + \mathbb{K} \right)^{-1} \mathbb{K} \right) \mathbb{K}^{-1} \right) \mathbb{K}_* \\ &= \mathbb{K}_{**} - {\mathbb{K}_*}^T \left(\mathbb{K}^{-1} - \mathbb{K}^{-1} + \left(\mathbb{W}^{-1} + \mathbb{K} \right)^{-1} \right) \mathbb{K}_* \\ &= \mathbb{K}_{**} - {\mathbb{K}_*}^T \left(\mathbb{W}^{-1} + \mathbb{K} \right) \mathbb{K}_* \end{aligned} \tag{5.3.11}$$

$\mathbb{E}\left[(H_* - \mathbb{E}[H_*|\mathbb{X} = \mathbf{x}, \mathbf{X}_* = \mathbf{x}_*, \mathbf{H} = \mathbf{h}])^2\right]$ represents the variance of H_* if we condition on the value $\mathbf{H} = \mathbf{h}$ while $\mathbb{E}\left[\left(\mathbb{E}\left[H_*|\mathbb{X} = \mathbf{x}, \mathbf{X}_* = \mathbf{x}_*, \mathbf{H} = \mathbf{h}\right] - \mathbb{E}\left[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right]\right)^2\right.$ comes about since $\mathbb{E}\left[H_*|\mathbb{X} = \mathbf{x}, \mathbf{X}_* = \mathbf{x}_*, \mathbf{H} = \mathbf{h}\right]$ depends on $\mathbf{H}$. Predictions are made by computing:

$$f\left[Y_* = +1|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right] = \int S(H_*)q(H_*|\mathbb{X} = \mathbf{x}, \mathbf{y}, \mathbf{x}_*)dH_* \tag{5.3.12}$$

where $q(H_*|\mathbb{X} = \mathbf{x}, \mathbf{y}, \mathbf{x}_*) = \mathcal{N}\left(\mathbb{K}_*^T\mathbb{K}^{-1}\hat{\mathbf{h}}\ ,\ \mathbb{K}_{**} - \mathbb{K}_*^T\left(\mathbb{W}^{-1} + \mathbb{K}\right)^{-1}\mathbb{K}_*\right)$ as obtained from (5.3.9) and (5.3.11).

A simpler method of finding the predicted class is known as the MAP prediction which is equivalent to:

$$S\left(\mathbb{E}_q\left[H_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right]\right) \tag{5.3.13}$$

The predicted class is the one that returns the highest value of (5.3.13). The predicted class chosen by selecting the class corresponding to the higher probability obtained through the MAP prediction and (5.3.12) is always the same for the binary case. Should we require some form of confidence in the expected class, then the equation (5.3.12) is used. It should be noted that the integral must be approximated when taking the logistic function as the sigmoid function.

5.3.2.3 Categorical Laplace Approximation

In the previous subsection, we looked at the binary case for Laplace Approximation, in this section we will consider the categorical case. The latent function random vector is now expressed as:

$$\mathbf{H} = \left(H_1^1, \ldots, H_N^1, H_1^2, \ldots, H_N^2, \ldots, H_1^C, \ldots, H_N^C\right)^T$$

such that $\mathbf{H} \sim \mathcal{N}\left(\mathbf{0}, \mathbb{K}\right)$ where $\mathbb{K}$ is the covariance matrix which is block diagonal

as the C latent functions are not independent. Each block of the matrix are denoted as $K_1, \ldots, K_C$ respectively, which represent the correlation of the functions within each class. $\mathbf{Y}$ is a vector of size $1 \times NC$, just like $\mathbf{H}$, however, made of 0s and 1s, such that 1 at the c^{th} position corresponds to the input X belonging to the class c and the rest of the vector is filled with 0s.

Recall that in the binary case the logistic function was used to map real values of a function to the interval $[0, 1]$. The softmax function is the equivalent when considering the categorical case, such that a k-dimensional vector has it's values mapped to the interval $[0, 1]$. This is expressed as:

$$f(y_i^c|\mathbf{H}_i = \mathbf{h}_i) = \frac{\exp\left[h_i^c\right]}{\sum_{c'} \exp\left[h_i^{c'}\right]}$$

Recall that $\Psi(\mathbf{h}) = \log\left[f\left(\mathbf{y}|\mathbf{H} = \mathbf{h}\right)\right] + \log\left[f\left(\mathbf{h}|\mathbb{X}\right)\right]$. The categorical equivalent to (5.3.6) is:

$$\Psi(\mathbf{h}) = \mathbf{y}^T\mathbf{h} - \sum_{i=1}^{N} \log\left[\sum_{c'=1}^{C} \exp\left(h_i^{c'}\right)\right] - \frac{1}{2}\mathbf{h}^T\mathbb{K}^{-1}\mathbf{h} - \frac{1}{2}\log\|\mathbb{K}\| - \frac{CN}{2}\log\left[2\pi\right]$$

By differentiating the above with respect to $\mathbf{h}$:

$$\nabla\Psi = -\frac{1}{2}K^{-1}\mathbf{h} + \mathbf{y} - \mathbf{p}$$

where $\mathbf{p}$ is a vector of size CN with the i^{th} entry equivalent to $f(y_i^c|\mathbf{H}_i = \mathbf{h}_i)$. The maximum value of Ψ occurs when $\nabla\Psi = 0$:

$$\hat{\mathbf{h}} = K(\mathbf{Y} - \mathbf{P})$$

Differentiating again gives:

$$\nabla\nabla\Psi = -K^{-1} - W$$

where $\mathbb{W} = diag(\mathbf{p}) - \mathbf{Q}\mathbf{Q}^T$ such that $\mathbf{Q}$ is a $CN \times N$ matrix obtained by stacking the matrices $diag(\mathbf{p}^c)$ vertically, where $\mathbf{p}^c$ is the subvector of $\mathbf{p}$ considering class c only.

Since $-\nabla\nabla\Psi$ is positive definite, this implies that the function $\Psi(\mathbf{h})$ is concave and has a unique maximum. Now, Newton's method is used to find the peak value of Ψ as follows:

$$\mathbf{h}' = (\mathbb{K}^{-1} + \mathbb{W})^{-1}(\mathbb{W}\mathbf{h} + \mathbf{y} - p)$$

Predictions are made by computing the posterior distribution:

$$q(\mathbf{h}_*|\mathbb{X}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*) = \int f(\mathbf{h}_*|\mathbb{X} = \mathbf{x}, \mathbf{X}_* = \mathbf{x}_*, \mathbf{H} = \mathbf{h}) q(\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{y}) d\mathbf{h}$$

Now, since both $f(\mathbf{h}_*|\mathbb{X}, \mathbf{x}_*, \mathbf{H} = \mathbf{h})$ and $q(\mathbf{h}|\mathbb{X}, \mathbf{Y} = \mathbf{y})$ are Gaussian, this means that the above integral will also be Gaussian. The mean and variance of $q(\mathbf{h}_*|\mathbb{X}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*)$ is computed below:

$$\mathbb{E}_q \left[H^c(\mathbf{x}_*)|\mathbb{X}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right] = K_{c*} K_c^{-1} \hat{\mathbf{h}}_c$$

where K_{c*} is the vector of covariances between the new point $\mathbf{x}_*$ and each of the original points $\mathbf{X}$, while $\hat{\mathbf{h}}_c$ is a subvector of $\hat{\mathbf{h}}$ related to the class c. Define Q_* to be the $CN \times C$ matrix such that it is block-diagonal with K_{c*} on the diagonals. Similar to the derivation of (5.3.11) we derive:

$$\begin{aligned} Cov_q(\mathbf{H}_*|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*) &= \Sigma + Q_*^T \mathbb{K}^{-1}(\mathbb{K}^{-1} + \mathbb{W})^{-1}\mathbb{K}^{-1} Q_* \\ &= diag(\mathbb{K}_{**}) - Q_*^T(\mathbb{K} + \mathbb{W}^{-1})^{-1} Q_* \end{aligned}$$

such that Σ is a diagonal $C \times C$ matrix and K_{**} is a vector of covariances. Applying the softmax function to the Gaussian distribution $q(\mathbf{h}_*|\mathbf{Y} = \mathbf{y})$ gives the predictive distribution. A method of obtaining the conditional expected value is to obtain samples from the distribution $q(\mathbf{h}_*|\mathbf{Y} = \mathbf{y})$, apply the softmax function and then find the average.

The Laplace Approximation of the marginal likelihood for the categorical case is

found in a similar fashion to the binary case:

$$\log\left[q(\mathbf{y}|\mathbb{X})\right] = -\frac{1}{2}\hat{\mathbf{h}}^T K^{-1}\hat{\mathbf{h}} - \sum_{i=1}^{N}\log\left[\sum_{c=1}^{C}\exp\left(h_i^c\right)\right] - \frac{1}{2}\log\left[I_{CN} + W^{\frac{1}{2}}KW^{\frac{1}{2}}\right]$$

5.3.2.4 Quality of the Laplace Approximation

Laplace's technique uses a normal distribution curve to approximate the posterior distribution, but does not say anything in regards to the quality of this approximation. This approximation is improved when it has certain assumptions satisfied, such that the prior is smooth and the sample size n is large enough. The Bernstein-von Mises Theorem [1] is the result equivalent to the asymptotic normality results for Maximum Likelihood Estimation. This theorem tells us that when the sample provides enough information, the posterior distribution for unknown quantities becomes independent of the prior distribution. Now, for large n, the conjugate posterior should have a similar shape to the normal distribution [1]. However, according to Chao Gao and Harrison H. Zhou [12], little is known about this theorem in the high-dimensional case.

Kuss and Rasmussen wrote a paper [16] comparing the approximations of the Laplace Approximation and the Expectation Propagation. To do this, they compare the results from both approximation methods to that of the Markov Chain Monte Carlo (MCMC) method, which is a complex but accurate method of approximation. The aspects of the approximations tested are the accuracy of the approximations to the marginal likelihood, and the quality of the probabilistic predictions. To do this, various experiments are carried out, and the results are evaluated using what the authors refer to as "the average information in bits of the predictions about test targets in excess of that of random guessing". The experiments revealed a significant difference between the two approximations. In particular, the Laplace Approximation of the posterior tends to underestimate the mean. For this reason, over conservative predictive probabilities are obtained with diminished information about the test labels. This combines to give inaccurate approximations, and in fact, the authors prefer using the Expectation Prop-

agation method as it is considered to be far more accurate. There are various papers attempting to improve the Laplace Approximation, such as that by Ruli, Sartori and Ventura [28].

5.3.3 Using Gaussian Process Classification to classify the dataset

The program using Gaussian Process classification is GPCR15v2.R and it makes use of the *gausspr* function in R, and its corresponding *predict* function. These functions are used to find the expected class of inputs given a dataset of already classified observations. The *gausspr* function is used to find the parameters of the approximated posterior distribution $q(\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}) \sim \mathcal{N}\left(\hat{\mathbf{h}}, (\mathbb{K}^{-1} + \mathbb{W})^{-1}\right)$ where $\hat{\mathbf{h}}$ is the maximum value of the latent function, $\mathbb{K}$ is the covariance matrix and $\mathbb{W} = -\nabla\nabla \log\left(f\left(\mathbf{y}|\mathbf{H} = \mathbf{h}\right)\right)$. This function uses the logistic likelihood $f\left(y_i|H_i = h_i\right) = -\log\left(1 + \exp\left(-y_i h_i\right)\right)$ and utilizes the Laplace Approximation to find the posterior. The parameter which is given as an output by the *gausspr* function is named alpha, which represents the vector $\mathbb{K}^{-1}\hat{\mathbf{h}}$, and this is used to obtain the parameters of the approximated posterior distribution.

Since the function is comparing pairs of clusters iteratively, and we have nine clusters, thirty six vectors are returned. Each vector corresponds to the equivalent $\mathbb{K}^{-1}\hat{\mathbf{h}}$ of every cluster comparison. The program which obtains the parameters of the approximating distribution $q(\mathbf{h}|\mathbb{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$, $\hat{\mathbf{h}}$ and $(\mathbb{K}^{-1} + \mathbb{W})^{-1}$, is the program GPParameters.R. The size of each vector and covariance matrix, however, is too large to display, as the covariance matrix is of size $(N_1 + N_2) \times (N_1 + N_2)$, where N_1 is the size of the first cluster being compared and N_2 is the size of the second cluster being compared, whilst the mean vector $\hat{\mathbf{h}}$ is of size $(N_1 + N_2) \times 1$. Naturally, it is impossible to show the distributions, due to their high dimensional nature.

The program GPCR15v2.R only considers the customers not used in the training set as the test set. Since the dataset was not originally classified, the clusters that were created before via cluster analysis are used as the given classes in the training set $\mathbb{X}_T$. So the dataset is partitioned into two parts, the training set $\mathbb{X}_T$ containing 493 customers which are assigned to the correct cluster, and the test set $\mathbb{X}_V$, containing the 500 customers which are to be classified. Note that 7 customers were excluded from the training set as they did not fit any of the clusters suitably. It is important to emphasize

the importance of the cluster analysis, as in normal scenarios, the observations belong to a certain class, and this can be said with a certainty. Gaussian classification is used for predictive purposes, and in most cases it is used to find the predicted class of an observation for which the class is already known. For example, consider a dataset consisting of dimensions of a flower's petals and sepals. We can say for certain what kind of flower this is, and using this data a Gaussian Process is used to classify new flowers, since the original flowers are by default a certain type of flower. However, this is not the case in the problem being faced. As we do not know the class of the observations, we cannot be certain of the predictions made by Gaussian classification. It is being used as a classification tool, rather than a predictive tool.

The *predict* function uses the parameters $\mathbb{K}^{-1}\hat{\mathbf{h}}$ obtained from the dataset $\mathbb{X}_T$ to find the conditional expectation $\mathbb{E}\left[H_*|\mathbb{X}_T = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right] = \mathbb{K}_*^T\mathbb{K}^{-1}\hat{\mathbf{h}}$. The probabilities corresponding to each class are evaluated using the MAP prediction:

$$S\left(\mathbb{E}\left[H_*|\mathbb{X}_T = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{X}_* = \mathbf{x}_*\right]\right)$$

The probability is calculated by comparing two classes once again and combines them using the *couple* function in the *kernlab* package. This returns a matrix of probabilities for which each row represents an observation, with the corresponding probability for each cluster in each column. The cluster corresponding to the highest probability, is the cluster to which the observation is assigned. Following this, basic statistics are found for each cluster so as to be able to analyse the difference in clusters.

5.3.4 Results of Application of Gaussian Process Classification

5.3.4.1 The training dataset

As previously mentioned, a training dataset is required for Gaussian classification, and the selection of customers within it is crucial. There is no fixed size for the training set, however, it was intuitively decided that half of each cluster is to be used as a training dataset $\mathbb{X}_T$. The rest of the dataset is unclassified and will be used as what is referred to as the test dataset $\mathbb{X}_V$. As the clustering itself is not ideal, and there are various customers in the wrong cluster, customers to be used in the training dataset must be selected carefully. This means that the customers selected must satisfy the profile of the cluster they belong to. Below are tables containing the basic statistics of the training dataset for each cluster. Table 5.3.2 shows that the clusters are far more coherent in this training set, and the mean values reflect the clusters accurately.

Cluster	Number	Combi	Singles	Winner	Loser
1	144	67	77	144	0
2	108	95	13	0	108
3	18	9	9	2	16
4	12	5	7	12	0
5	22	0	22	14	8
6	88	67	21	0	88
7	60	8	52	18	42
8	26	21	5	0	26
9	15	0	15	0	15

Table 5.3.1: The type of customers in the training dataset

Cluster		TotalStake	TotalPayout	Bets	Wins	Profit	Singles	Combis	MobSinBets	WebSinBets	LiveSinBets	PMSinBets	Margin	CustNo
1	AVG	831.79	962.79	88.60	21.18	-131.00	34.41	54.19	5.94	28.47	6.69	27.72	-0.24	144
	MAX	2,996.58	3,326.65	689.00	85.00	-0.13	167.00	684.00	59.00	154.00	91.00	166.00	0.00	
	MIN	10.37	11.94	15.00	1.00	-485.74	0.00	0.00	0.00	0.00	0.00	0.00	-3.81	
2	AVG	1,153.31	811.01	172.26	23.72	342.30	34.60	137.66	25.61	8.99	12.43	22.18	0.35	108
	MAX	3,641.95	2,595.38	509.00	111.00	2,145.29	116.00	465.00	107.00	91.00	108.00	97.00	1.00	
	MIN	35.23	0.00	81.00	0.00	16.36	0.00	1.00	0.00	0.00	0.00	0.00	0.05	
3	AVG	5,773.22	5,183.43	199.39	49.61	589.79	85.83	113.56	38.78	47.06	40.44	45.39	0.09	18
	MAX	8,581.39	7,111.80	355.00	96.00	1,762.75	163.00	291.00	114.00	160.00	121.00	124.00	0.22	
	MIN	3,725.88	3,134.06	100.00	20.00	-1,860.52	18.00	0.00	0.00	1.00	2.00	8.00	-0.37	
4	AVG	5,395.56	5,807.58	186.67	60.00	-412.01	97.33	89.33	19.00	78.33	16.75	80.58	-0.08	12
	MAX	9,515.80	9,657.84	413.00	103.00	-24.53	187.00	386.00	165.00	187.00	73.00	187.00	-0.01	
	MIN	3,054.00	3,092.34	106.00	11.00	-1,080.11	0.00	0.00	0.00	0.00	0.00	0.00	-0.20	
5	AVG	3,106.46	3,500.97	43.23	18.50	-394.51	40.59	2.64	0.77	39.82	8.82	31.77	-0.23	22
	MAX	5,907.76	5,572.02	67.00	28.00	1,022.23	66.00	12.00	9.00	65.00	29.00	51.00	0.17	
	MIN	1,066.02	1,674.87	22.00	7.00	-2,464.75	21.00	0.00	0.00	20.00	0.00	12.00	-0.91	
6	AVG	1,774.55	1,498.97	413.93	82.08	275.58	116.53	297.40	15.55	100.99	49.35	67.18	0.20	88
	MAX	8,072.52	7,885.24	1,738.00	221.00	1,436.76	436.00	1,738.00	153.00	436.00	429.00	359.00	1.00	
	MIN	102.51	0.00	39.00	0.00	3.79	0.00	0.00	0.00	0.00	0.00	0.00	0.01	
7	AVG	27,117.90	25,983.84	1,051.63	418.90	1,134.06	787.35	264.28	196.87	590.48	514.37	272.98	0.05	60
	MAX	345,938.85	327,396.32	5,150.00	1,479.00	18,542.53	2,431.00	3,131.00	1,037.00	2,261.00	2,372.00	1,441.00	0.41	
	MIN	95.05	81.07	504.00	68.00	-10,781.65	137.00	0.00	0.00	0.00	0.00	0.00	-0.38	
8	AVG	39,054.46	30,191.37	1,445.46	181.46	8,863.10	252.69	1,192.77	168.62	84.08	196.00	56.69	0.24	26
	MAX	75,594.10	65,449.69	9,367.00	793.00	17,663.72	996.00	8,804.00	711.00	508.00	954.00	385.00	0.52	
	MIN	20,704.60	12,746.85	123.00	8.00	2,608.45	7.00	1.00	0.00	0.00	0.00	1.00	0.06	
9	AVG	276,575.96	263,191.96	8,808.93	4,095.67	13,384.00	8,465.53	343.40	1,554.80	6,910.73	7,687.13	778.40	0.06	15
	MAX	1,412,481.55	1,355,634.80	40,451.00	8,830.00	56,846.75	40,451.00	1,991.00	11,821.00	40,451.00	40,451.00	6,961.00	0.17	
	MIN	722.18	687.15	1,210.00	1,172.00	35.03	1,196.00	0.00	0.00	124.00	0.00	0.00	0.00	

Table 5.3.2: The basic statistics of the each cluster within the training dataset

5.3.4.2 Gaussian classification of the customers belonging to the test dataset

The Gaussian classification resulted in three clusters being left unpopulated. The results from Cluster 6 were ignored as there is little to no correlation between the customers. What is most likely to have happened is that the customer was given an equal probability of being assigned to each cluster. The clusters that are still populated and had satisfactory results are:

- Cluster 1: Low Value Winners
- Cluster 2: Low Value Losers
- Cluster 7: Sharp High Volume Bettors
- Cluster 8: High Roller Losers
- Cluster 9: Sharp Extreme Players

Cluster	Number	Combi	Singles	Winner	Loser
1	226	70	156	81	145
2	188	98	90	0	188
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	5	4	1	0	5
7	24	11	13	2	22
8	26	18	8	0	26
9	31	7	24	3	28

Table 5.3.3: The type of customers in the Gaussian classification classes for the test dataset

Cluster		TotalStake	TotalPayout	Bets	Wins	Profit	Singles	Combis	MobSinBets	WebSinBets	LiveSinBets	PMSinBets	Margin	CustNo
1	AVG	2,790.48	2,913.58	150.76	55.28	-123.10	93.21	57.54	26.44	66.77	29.72	63.50	-0.01	226
	MAX	52,734.80	51,787.92	741.00	324.00	1,142.39	537.00	569.00	486.00	531.00	395.00	489.00	0.18	
	MIN	7.33	6.32	12.00	2.00	-5,418.88	0.00	0.00	0.00	0.00	0.00	0.00	-1.14	
2	AVG	1,187.99	785.35	141.46	22.34	402.64	55.63	85.82	19.15	36.48	20.61	35.02	0.40	188
	MAX	7,826.59	6,356.39	1,166.00	189.00	4,166.53	542.00	1,126.00	420.00	431.00	542.00	424.00	1.00	
	MIN	16.58	0.00	10.00	0.00	4.88	0.00	0.00	0.00	0.00	0.00	0.00	0.14	
3	AVG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
	MAX	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	MIN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	AVG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
	MAX	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	MIN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
5	AVG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
	MAX	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	MIN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
6	AVG	240,595.79	217,663.83	4,252.80	535.20	22,931.96	546.80	3,706.00	319.80	227.00	470.80	76.00	0.22	5
	MAX	742,185.29	715,506.36	9,367.00	1,155.00	75,055.52	1,452.00	8,804.00	1,322.00	508.00	1,388.00	189.00	0.32	
	MIN	837.78	620.63	2,151.00	153.00	217.15	222.00	699.00	0.00	0.00	58.00	6.00	0.04	
7	AVG	205,403.19	183,969.86	1,848.67	534.04	21,433.33	991.42	857.25	601.38	390.04	880.96	110.46	0.10	24
	MAX	625,694.45	596,414.04	5,747.00	1,172.00	53,487.05	3,334.00	5,699.00	2,537.00	1,741.00	3,121.00	405.00	0.22	
	MIN	201.85	200.31	448.00	154.00	-5,106.47	48.00	0.00	0.00	0.00	0.00	1.00	-0.07	
8	AVG	52,042.91	43,779.43	1,058.73	240.54	8,263.48	354.12	704.62	198.08	156.04	252.92	101.19	0.17	26
	MAX	209,743.70	179,487.20	2,688.00	363.00	30,350.29	743.00	2,453.00	711.00	515.00	734.00	404.00	0.35	
	MIN	749.94	649.13	400.00	90.00	100.81	49.00	0.00	0.00	2.00	12.00	1.00	0.08	
9	AVG	112,704.16	103,979.66	2,436.61	898.03	8,724.51	1,564.87	871.74	562.58	1,002.29	1,296.58	268.29	0.11	31
	MAX	718,169.30	655,512.99	5,095.00	1,954.00	62,656.30	3,799.00	4,242.00	2,651.00	3,784.00	3,795.00	1,686.00	0.37	
	MIN	379.54	237.62	467.00	105.00	-2,904.71	41.00	0.00	0.00	0.00	11.00	2.00	-0.08	

Table 5.3.4: The basic statistics of the Gaussian classification classes for the test dataset

Cluster 1 is more lenient as to how it selected customers, particularly in terms of stake, and also, profit. This can be seen from table 5.3.4 where the maximum stake, and even the average stake, in Cluster 1 is significantly higher than that in table 3.2.1. A significant difference between this cluster in the training set and in the test set, is the number of losers in the cluster as shown in table 5.3.3. The types of players in the cluster have changed slightly, and we can refer to players in this cluster as "Normal-Low Value Sharp Players" instead. However, it is important to note that the winners in this case are more dangerous and many won a significant sum of money.

Cluster 2 is made of customers that are only losers, as shown in table 5.3.3, and do not win many bets. Similarly to Cluster 1, Cluster 2 also has a greater range of customers, some of which have a higher stake and higher number of bets when compared to its selections in the training set. However, when looking at the means of the variables in table 5.3.4 and those in table 3.2.1, it can be seen that as a whole, the means of the training set and of the test set are quite similar. This indicates to us that the profile of this cluster was maintained, despite containing some customers with a slightly higher stake and number of bets. Since Gaussian classification is returning less populated clusters than before, it is natural that these two clusters included more customers.

Below is figure 5.3.2, which is the PCA plot of the classified customers for the first two clusters. Figure 5.3.1 illustrates the scree plot used to show the variances each component is responsible for. The scree plot indicates the eigenvalues for each component, the first of which has an eigenvalue of 6.12. The second component has a far smaller eigenvalue of 1.86. Their corresponding eigenvectors are responsible for the weights assigned to each variable. Once again, the first two components are responsible for around 60% of the variance, with the first component being responsible for 47% of the variance, and the second component being responsible for around 14% of the variance. The rest of the components were not considered as they were responsible for around 10% of the variance or less.

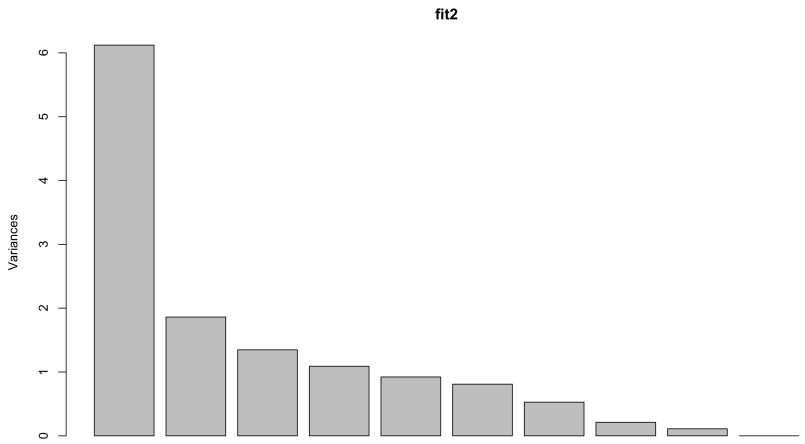

Figure 5.3.1: Scree plot of the principal components used to illustrate the Gaussian classification

The first principal component once again represents the amount that is bet, with NumberOfBets, NumberOfWins, No.Of.Single.Bets and Live.Single.Bets all being assigned weights between 0.33 and 0.37. However, in this case, slightly higher weights than previously were assigned to the TotalStake (0.31) and TotalPayout (0.31) variables. This component is essentially telling us how active a customer was, either in terms of bets made or the amount staked. On the other hand, in the second principal component, the TotalStake, TotalPayout and Profit are more influential. However, the variables that were important in the first principal component, were all assigned negative weights. Most notably, the Web.Single.Bets and Prematch.Single.Bets were assigned the weights -0.48 and -0.39 respectively. It should be noted, that these variables will only have high values if the customer bet a high amount of bets. Furthermore, customers that bet these types of bets tend to be smart players, which is a useful point to note.

The difference between clusters is more discernible in this classification than in cluster analysis. This plot illustrates that the customers in Cluster 2 have a higher profit than those in Cluster 1, and also illustrates the fact that the customers in Cluster 1 bet Single Bets on the web platform more often than those in Cluster 2. This variable

is useful to us as it gives an indication of the type of customers in the cluster that we are dealing with.

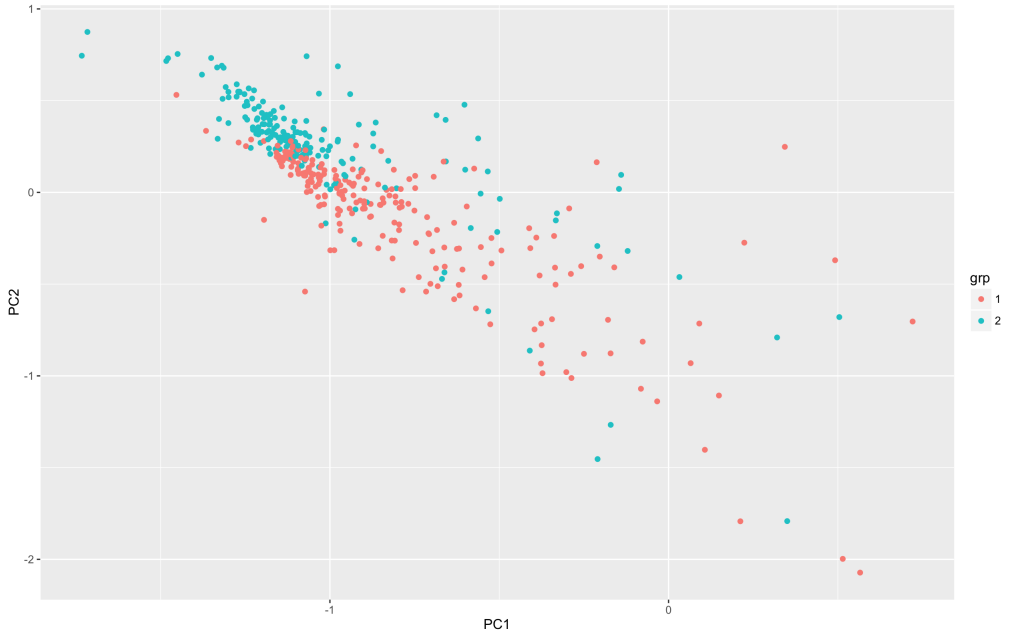

Figure 5.3.2: Illustration of the Normal Customers using PCA after Gaussian classification

Additionally, the customers in Cluster 1 also have lower profits than those in Cluster 2, indeed, on average, their profit is negative, as seen in figure 5.3.4. Note also that customers in Cluster 1 bet more often than those in Cluster 2, shown by their comparative PC1 values. The plot below illustrates the centroids of these clusters, which are fairly close. However, it is evident that the biggest difference between the two clusters is the number of bets made and the profit made. The PC2 value of Cluster 2 is higher since the profit made is higher, whilst the PC1 value of the centroid of Cluster 1 is higher as they bet a higher number of bets. Additionally, note that in Cluster 1 customers tend to bet more Web and Prematch Single Bets, which contributes to a lower PC2 value. These customers still retain their conservative property, that is, their PC values are both fairly close to 0, which is a big difference to the PC values of the other clusters.

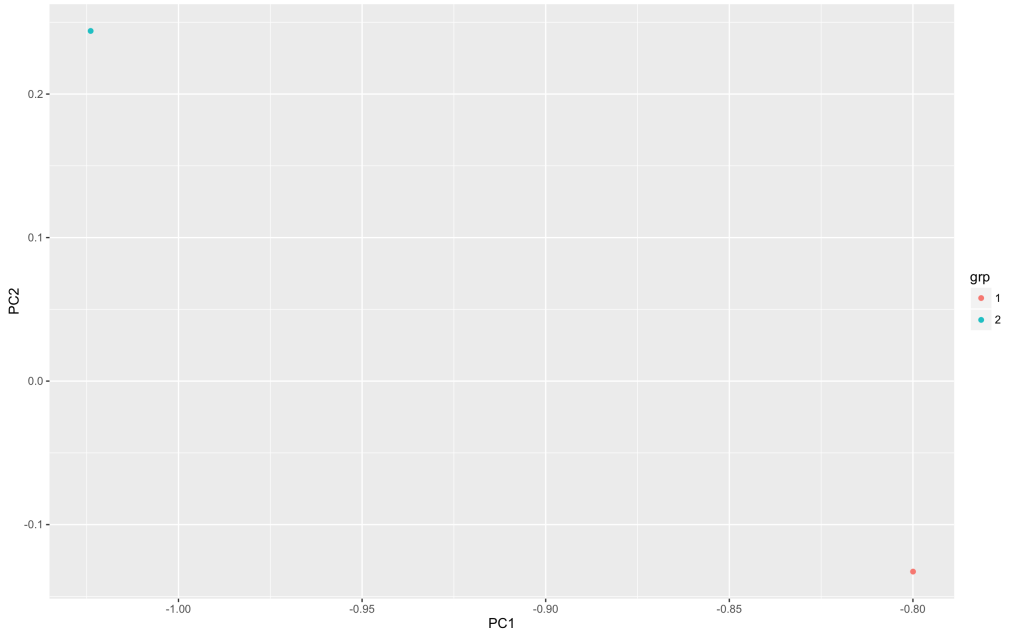

Figure 5.3.3: PCA illustration of the centroids of the clusters containing Normal Customers after Gaussian classification

Cluster 7 contains customers that made a high number of bets, as in the training set. The lowest number of bets that one customer made was 448 bets, indicating that the property of containing customers with a high number of bets was maintained. In addition, this cluster also contained some winners, and the customers in this class contained quite a low margin. This indicates that all players within this cluster are "Sharp High Volume Bettors", as in the cluster analysis.

Cluster 8 includes many customers with low stakes, which does not fit the definition we gave it in the cluster analysis step. In fact, eleven customers classified in this cluster had a significantly lower stake than €20,000. It should be noted how the range of stakes varies in the training set and test set. On average, the stakes in Cluster 8 are far smaller than those in clusters 7 and 9, as seen in figure 5.3.4. Similarly to the cluster in the training set, there are no winners in this cluster. The customers also have a lower margin than those in Cluster 7, as is the case in the training set.

Cluster 9 accepted extreme customers in this dataset, and retained the properties

shown in the training dataset. A plot of the Total Stake against the Number of Bets for this cluster is shown below. The results indicate that all customers in this cluster have a high number of bets, and those that do not bet a significantly high amount of money, as was the case in the training set $\mathbb{X}$. Figure 5.3.4 illustrates this point.

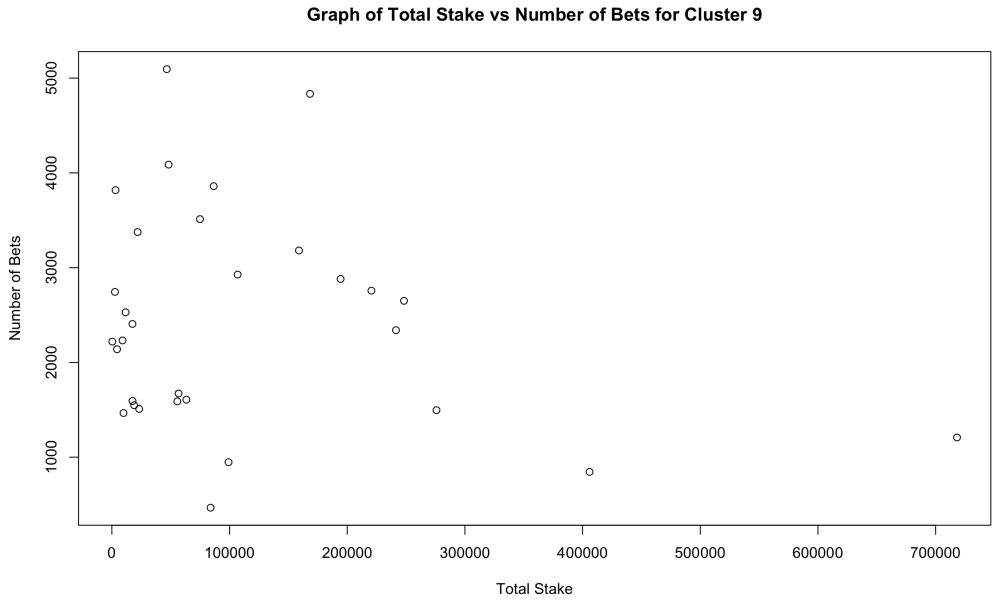

Figure 5.3.4: Scatter plot illustrating the total stake against the number of bets for Cluster 9

Once again, the bettors are illustrated using PCA with the same weights as those used when illustrating clusters 1 and 2. The plot in figure 5.3.5 illustrates further that the customers in clusters 7 and 9 differ in terms of the number of bets made relative to the amount staked. Cluster 9 has a significantly higher number of average bets made, than those in Cluster 7. In addition, customers in Cluster 7 tend to have a higher stake and profit than those in Cluster 9. These attributes indicate that customers in Cluster 9 are smarter than those in Cluster 7. Furthermore, customers in Cluster 7 had significantly less WebSingleBets made than those in Cluster 9, a huge factor when considering the PC2 value. Cluster 8 on the other hand, is far more conservative in terms of the number of bets made and the total stake. As the number of bets is comparatively small in this cluster, the PC1 value of each customer is much lower than

the values of those in clusters 8 and 9.

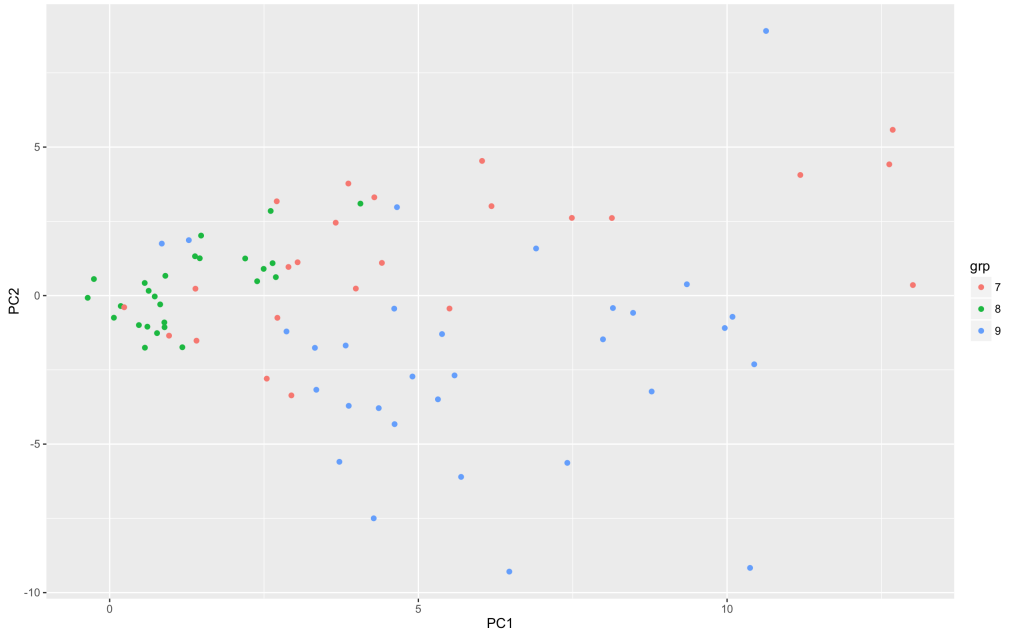

Figure 5.3.5: Illustration of the Extreme Customers using PCA after Gaussian classification

The plot of the centroids of the clusters further illustrates these points, and as can be seen the clusters are very different, either in terms of the amount they stake, or in the amount they bet. For instance, clusters 7 and 9 have similar PC1 values, however the PC2 values are totally different, mainly due to the high stakes and profit in Cluster 7. In addition, customers in Cluster 9 bet far more WebSingleBets and PreMatchSingleBets, indicating the customers in this cluster are also smarter than those in Cluster 7. On the other hand, customers in Cluster 8 are slightly more conservative than those in clusters 7 and 9. In particular, the PC2 value is close to 0, indicating that the number of bets made by these customers relative to the amount stake is fairly normal. Indeed on average, a customer stakes just over €52,000 and makes over 1,000 bets, which in comparison to the averages of clusters 7 and 9, is slightly more rational.

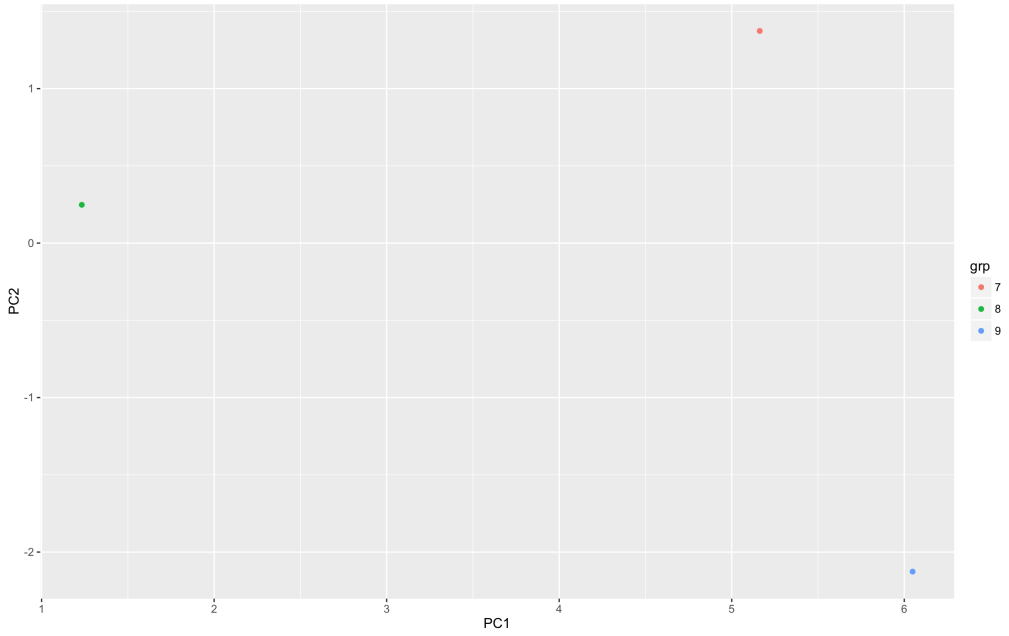

Figure 5.3.6: Illustration of the Extreme Customers using PCA after Gaussian classification

```
> table(dnew[,20],comp[,20])

   1  2  3  4  5  6  7  8  9
1 77 31 12 15 20 44 27  0  0
2 68 78  6  0  3 29  4  0  0
6  0  0  0  0  0  2  0  1  2
7  0  0  0  0  0  3  6  7  8
8  0  0  0  0  0 10  2 13  1
9  0  0  0  0  0  0 21  5  5
```

Figure 5.3.7: Confusion matrix showing how the customers are split in the cluster analysis and the Gaussian classification

The confusion matrix in figure 5.3.7 indicates how the customers are classified in Gaussian classification when compared to cluster analysis. Note how in Gaussian classification, Cluster 1 contains customers that are in the first seven clusters in cluster analysis, while Cluster 2 contains losers that were in the first seven clusters. Cluster 1, in particular, has some customers with a significantly high stake. In the cluster analysis, clusters 3, 4 and 5 all contained a few misclassified observations, as expected. However, as previously mentioned, these clusters were no longer populated in the Gaussian classification, and most of the customers in these clusters were included in Cluster 1. Notice how the customers in Cluster 6 were mostly included in clusters 1 and 2.

Additionally, it should be noted that many customers that were in Cluster 7 became a part of Cluster 9 and vice versa.

Chapter 6

Conclusion

The aim of this dissertation was to apply some form of classification to the dataset, and to compare this to the results of some higher level method of classification. Naturally, the results obtained were not crisp, and nor were they expected to be as such, due to the nature of classification. The results of cluster analysis were not especially accurate, however, gave a fairly clear indication as to how the clusters were divided. There were certain clusters that tended to overlap, and many customers that were far from the centroid, that is, they did not fit the description of the cluster. However, investigating each cluster using the basic statistics, and even at a customer level, allowed us to decipher profiles of each cluster. What is a strength of cluster analysis becomes a weakness in this scenario, and this is that cluster analysis allows the data to speak. Using cluster analysis on such a complex dataset implies that the clusters will almost certainly not be exactly as the user wishes, and there will always be some form of overlapping between clusters. Various different attempts were made at clustering the data effectively, however, in all cases there was no particular clustering that was ideal.

Half of the dataset was used to carry out the Gaussian classification and only the customers that best represented each cluster in the cluster analysis were used. Gaussian classification rendered four of the clusters redundant and compressed the classification into five clusters. It seems that this technique preferred to not have such a large number of different classes. This is not ideal, as it slightly alters the characteristics of some of the clusters. Most notably, Cluster 1 contained a much larger variety of customers and

did not retain the exclusivity it previously had. Nevertheless, there are still evident differences between the classes and the effects of the classification were still impressive. Upon comparing these results to those from the cluster analysis, there is an indication that these results are quite good. The fact that the definition of each cluster has been changed is certainly not what we wanted, however, this is most likely to have occurred due to the structure of the data being used. Identifying a customer with a particular cluster is not always obvious, even for people in the area of betting, so for this technique to be able to identify a certain pattern, is an impressive feat.

Unfortunately, there were some limitations in this methodology. The biggest limitation of the data, as has been previously mentioned, was the fact that it was not initially classified. This means that the Gaussian classification relied heavily on the cluster analysis and intuition. However, as there was no initial framework for how to cluster the data, this also allowed for more flexibility. As previously mentioned, the form of the data was not the easiest to work with. For instance, due to the structure of the data, the cluster analysis was quite complicated to carry out and involved removing what we referred to as "extreme customers" before actually carrying out the clustering effectively.

The area of classification has plenty of potential, and this dissertation is meant to illustrate this. There are many improvements that can be applied to the methodology, and various different scenarios that this problem can be applied to. For instance, it would be interesting to compare the results provided by the Expectation Propagation method, which is known to provide a better normal distribution approximation [16], to those of the Laplace Approximation. In conclusion, Gaussian classification continues to show its potential, and whilst the results were not ideal, there are plenty of improvements that could be made to yield a better classification and one that could be applied throughout various industries.

Chapter 7

Appendix

7.1 Useful Theorems

Theorem: Let $\mathbf{X}$ be a random vector satisfying $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ which can be divided into groups such that $\mathbf{X} = (\mathbf{X}_1\ ,\ \mathbf{X}_2)'$ where $\mathbf{X}_1 = (X_1, \ldots, X_p)$ and $\mathbf{X}_2 = (X_{p+1}, \ldots, X_D)$, $\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}$ and $Var[\mathbf{X}] = \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$. Then:

$$\mathbf{X}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$$

Proof: Let $\mathbf{I}$ be a $q \times q$ unit matrix, $\mathbf{O}$ be a $q \times (p-q)$ null matrix and $\mathbf{B} = (\mathbf{I}\ ,\ \mathbf{O})$. Then

$$\mathbf{BX} = (\mathbf{I}\ ,\ \mathbf{O}) \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} = \mathbf{X}_1$$

From this transformation we deduce:

$$
\begin{aligned}
\mathbb{E}[\mathbf{X}_1] &= \mathbb{E}[\mathbf{BX}] \\
&= \mathbf{B}\mathbb{E}[\mathbf{X}] \\
&= (\mathbf{I}\ ,\ \mathbf{O}) \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \\
&= \boldsymbol{\mu}_1
\end{aligned}
$$

$$
\begin{aligned}
Var[\mathbf{X}_1] &= Var[\mathbf{BX}] \\
&= \mathbf{B}Var[\mathbf{X}]\mathbf{B}^T \\
&= (\mathbf{I}\ ,\ \mathbf{O}) \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \mathbf{O} \end{pmatrix} \\
&= \boldsymbol{\Sigma}_{11}
\end{aligned}
$$

Since $\mathbf{X}_1$ is a linear transformation of $\mathbf{X}$ then normality is preserved:

$$\mathbf{X}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$$

□

Theorem: Let $\mathbf{X}$ be a random vector satisfying $\mathbf{X} = (\mathbf{X}_1\ ,\ \mathbf{X}_2)' \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that where $\mathbf{X}_1 \in \mathbb{R}^q$ and $\mathbf{X}_2 \in \mathbb{R}^{D-q}$. Let $\mathbf{X}_{21} = \mathbf{X}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\mathbf{X}_1$ form the partitioned covariance matrix $\begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$. Then for $\boldsymbol{\mu}_{21} = \boldsymbol{\mu}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_{221} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$:

$$\mathbf{X}_{21} \sim \mathcal{N}(\boldsymbol{\mu}_{21}, \boldsymbol{\Sigma}_{221})$$

Furthermore, $\mathbf{X}_1$ and $\mathbf{X}_{21}$ are statistically independent.

Theorem: Let $\mathbf{X}$ be a random vector satisfying $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that the rank of

$\mathbf{\Sigma}$ is p, the conditional distribution of $\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1$ is:

$$\mathcal{N}(\boldsymbol{\mu}_2 + \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1)\ ,\ \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12})$$

Proof: Let $f(\mathbf{x})$ be the density function of $\mathbf{X}$, $f_1(\mathbf{x})$ be the density function of $\mathbf{X}_1$ and $f_{21}(\mathbf{x}_2|\mathbf{x}_1)$ be the conditional density function of $\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1$**.** By definition:

$$f_{2|1}(\mathbf{x}_2|\mathbf{x}_1) = \frac{f(\mathbf{x})}{f_1(\mathbf{x}_1)} = \frac{f_{1,2}(\mathbf{x}_1, \mathbf{x}_2)}{f_1(\mathbf{x}_1)}$$

Let $\mathbf{Y}_1 = \mathbf{X}_1$ and $\mathbf{Y}_2 = \mathbf{X}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_1$, then:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{O} \\ -\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} = \mathbf{AX}$$

$$f_{2|1}(\mathbf{x}_2|\mathbf{x}_1) = \frac{f_{1,2}(\mathbf{x}_1, \mathbf{x}_2)}{f_1(\mathbf{x}_1)} = \frac{f_{\mathbf{Y}_1,\mathbf{Y}_2}\left[\mathbf{y}_1(\mathbf{x}_1, \mathbf{x}_2), \mathbf{y}_2(\mathbf{x}_1, \mathbf{x}_2)\right]}{f_1(\mathbf{x}_1)} \left|\frac{\delta \mathbf{y}}{\delta \mathbf{x}}\right|$$

From $\mathbf{Y} = \mathbf{AX}$ we obtain $\frac{\delta \mathbf{y}}{\delta \mathbf{x}} = \mathbf{A}$ which gives $\left|\frac{\delta \mathbf{y}}{\delta \mathbf{x}}\right| = 1$. Independence of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ given by the previous theorem gives:

$$\begin{aligned} f_{\mathbf{Y}_1,\mathbf{Y}_2}\left[\mathbf{y}_1(\mathbf{x}_1, \mathbf{x}_2), \mathbf{y}_2(\mathbf{x}_1, \mathbf{x}_2)\right] &= f_{\mathbf{Y}_1}\left[\mathbf{y}_1(\mathbf{x}_1, \mathbf{x}_2)\right] f_{\mathbf{Y}_2}\left[\mathbf{y}_2(\mathbf{x}_1, \mathbf{x}_2)\right] \\ &= f_{\mathbf{Y}_1}\left[\mathbf{x}_1\right] f_{\mathbf{Y}_2}\left[\mathbf{x}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{x}_1\right] \\ &= f_{\mathbf{X}_1}\left[\mathbf{x}_1\right] f_{\mathbf{Y}_2}\left[\mathbf{x}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{x}_1\right] \end{aligned}$$

Hence,

$$\begin{aligned} f_{2|1}(\mathbf{x}_2|\mathbf{x}_1) &= \frac{f_{\mathbf{X}_1}\left[\mathbf{x}_1\right] f_{\mathbf{Y}_2}\left[\mathbf{x}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{x}_1\right]}{f_1(\mathbf{x}_1)} \\ &= f_{\mathbf{Y}_2}\left[\mathbf{x}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{x}_1\right] \end{aligned}$$

The previous theorem also gives us the distribution of $\mathbf{Y}_2$ which is $\mathcal{N}(\boldsymbol{\mu}_2 - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\boldsymbol{\mu}_1, \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12})$. Thus:

$$f_{\mathbf{Y}_2}(\mathbf{y}_2) = \frac{1}{(2\pi)^{\frac{p-q}{2}} |Var(\mathbf{Y}_2)|^{\frac{1}{2}}} exp\left(-\frac{1}{2}[\mathbf{y}_2 - \mathbb{E}(\mathbf{Y}_2)]'[Var(\mathbf{Y}_2)][\mathbf{y}_2 - \mathbb{E}(\mathbf{Y}_2)]\right)$$

such that

$$\begin{aligned}\mathbf{Y}_2 - \mathbb{E}(\mathbf{Y}_2) &= \left(\mathbf{X}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\mathbf{X}_1\right) - \left(\boldsymbol{\mu}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\mu}_1\right) \\ &= \mathbf{X}_2 - \left(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{X}_1 - \boldsymbol{\mu}_1)\right)\end{aligned}$$

Since

$$f_{2|1}(\mathbf{x}_2|\mathbf{x}_1) = f_{\mathbf{Y}_2}\left[\mathbf{x}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\mathbf{x}_1\right] = f_{\mathbf{Y}_2}(\mathbf{y}_2)$$

then the conditional distribution of $\mathbf{X}_2|\mathbf{X}_1 = \mathbf{x}_1$ is $\mathcal{N}(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1)\ ,\ \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12})$

□

7.2 Inverse Probability Integral Transform

The Inverse Probability Integral Transform essentially converts a continuous random variable to a uniformly distributed random variable. It involves the following statement: Suppose that X is a continuous random variable with cumulative distribution function F_X. Then the random variable Y defined such that

$$Y = F_X(X)$$

is uniformly distributed.

7.3 Embedding Inputs to Higher Dimensions

In this section, we extend the model above to include higher dimensions. This model maps inputs into a high dimensional space and applies the linear model within this space instead of on the inputs. For example, consider a scalar x which we want to map into a high dimensional space, such as the space of powers of x. Using the function $\boldsymbol{\phi}$, we have $\boldsymbol{\phi}(x) = (x, x^2, x^3, \ldots)^T$ which we use for polynomial regression. As long as the function $\boldsymbol{\phi}$ is independent of the parameter vector $\mathbf{w}$ the model remains linear with the parameters.

Consider the function $\phi : \mathbb{R}^d \to \mathbb{R}^N$ with a d-dimensional input vector $\mathbf{X} = \mathbf{x}$ mapped to an N-dimensional feature space such that the model is now expressed as:

$$g(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w} \tag{7.3.1}$$

with vector of weights $\mathbf{w}$ of length N. Suppose $\Phi(\mathbb{X})$ is the combination of column vectors $\boldsymbol{\phi}(\mathbf{x})$ for all possibilities in the training set. Analysis is carried out in a similar fashion as to that for the linear model, see equation (5.2.8), however, since we are considering higher dimensions, $\mathbf{X}$ is replaced by $\Phi(\mathbf{x})$ and taking $\mathbf{Y} = \mathbf{y}$:

$$Y_*|\mathbf{x}_*, \mathbb{X}, \mathbf{y} \sim \mathcal{N}\left(\frac{1}{\sigma_n^2}\boldsymbol{\phi}(\mathbf{x}_*^T)A^{-1}\Phi(\mathbb{X})\mathbf{y}\ ,\ \boldsymbol{\phi}(\mathbf{x}_*^T)A^{-1}\boldsymbol{\phi}(\mathbf{x}_*)\right) \tag{7.3.2}$$

where $A = \frac{1}{\sigma_n^2}\Phi(\mathbb{X})\Phi(\mathbb{X}^T) + \mathbf{\Sigma}_p^{-1}$. To make predictions using this equation we need to invert A, which is a matrix of size $N \times N$. Finding the inverse of this matrix is too complicated when N is large. Hence, the equation is reformulated so as to eliminate this issue. Consider:

$$\begin{aligned}
\frac{1}{\sigma_n^2}\Phi(\mathbb{X})(R + \sigma_n^2 I) &= \frac{1}{\sigma_n^2}\Phi(\mathbb{X})(\Phi(\mathbb{X})^T\mathbf{\Sigma}_p\Phi(\mathbb{X}) + \sigma_n^2 I) \\
&= \frac{1}{\sigma_n^2}(\Phi(\mathbb{X})\Phi(\mathbb{X})^T\mathbf{\Sigma}_p\Phi(\mathbb{X}) + \sigma_n^2\Phi(\mathbb{X})I) \\
&= \frac{1}{\sigma_n^2}(\Phi(\mathbb{X})\Phi(\mathbb{X})^T\mathbf{\Sigma}_p\Phi(\mathbf{X}) + \sigma_n^2 I\Phi(\mathbb{X})) \\
&= \frac{1}{\sigma_n^2}(\Phi(\mathbb{X})\Phi(\mathbb{X})^T\mathbf{\Sigma}_p + \sigma_n^2 I)\Phi(\mathbb{X}) \\
&= \frac{1}{\sigma_n^2}(\Phi(\mathbb{X})\Phi(\mathbb{X})^T + \sigma_n^2 I\mathbf{\Sigma}_p^{-1})\mathbf{\Sigma}_p\Phi(\mathbb{X}) \\
&= A\mathbf{\Sigma}_p\Phi(\mathbb{X})
\end{aligned}$$

where $R = \Phi(\mathbb{X})^T\mathbf{\Sigma}_p\Phi(\mathbb{X})$. Now, we pre-multiply the above by A^{-1} and post-multiply by $(R + \sigma_n^2 I)^{-1}$ to obtain:

$$\begin{aligned}
\frac{1}{\sigma_n^2}\Phi(\mathbb{X})(R + \sigma_n^2 I) &= A\Sigma_p\Phi(\mathbb{X}) \\
\frac{1}{\sigma_n^2}A^{-1}\cdot\Phi(\mathbb{X})(R + \sigma_n^2 I)\cdot(R + \sigma_n^2 I)^{-1} &= A^{-1}\cdot A\Sigma_p\cdot\Phi(\mathbb{X})(R + \sigma_n^2 I)^{-1} \\
A^{-1}\Phi(\mathbb{X}) &= \sigma_n^2\Sigma_p\Phi(\mathbb{X})(R + \sigma_n^2 I)^{-1} \qquad (7.3.3)
\end{aligned}$$

Hence we have converted the mean in (7.3.2) into a simpler formula. However, there still remains the variance to adjust. Consider the Matrix Inversion Lemma which tells us:

$$(Z + UWV^T)^{-1} = Z^{-1} - Z^{-1}U\left(W^{-1} + V^TZ^{-1}U\right)^{-1}V^TZ^{-1} \qquad (7.3.4)$$

Consider the variance of $Y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbb{X}, \mathbf{Y} = \mathbf{y}$, given to be $\boldsymbol{\phi}(\mathbf{x}_*^T)A^{-1}\boldsymbol{\phi}(\mathbf{x}_*)$ in (7.3.2). Recall that $A = \frac{1}{\sigma_n^2}\Phi(\mathbb{X})\Phi(\mathbb{X}^T) + \Sigma_p^{-1}$ and note the similarity to the left hand side of equation (7.3.4). Let $Z^{-1} = \Sigma_p$, $W^{-1} = \sigma_n^2 I$ and $V = U = \Phi$ and make use of

(7.3.2) such that:

$$\begin{aligned} A^{-1} &= \left(\Sigma_p^{-1} + \frac{1}{\sigma_n^2}\Phi(\mathbb{X})\Phi(\mathbb{X})^T\right)^{-1} \\ A^{-1} &= \Sigma_p - \Sigma_p\Phi(\mathbb{X})(\sigma_n^2 I + \Phi(\mathbb{X})^T\Sigma_p\Phi(\mathbb{X}))^{-1}\Phi(\mathbb{X})^T\Sigma_p \end{aligned} \tag{7.3.5}$$

From equations (7.3.3) and (7.3.5) we obtain a rearrangement of the parameters of $Y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbb{X}, \mathbf{Y} = \mathbf{y}$:

$$Y_*|\mathbf{X}_* = \mathbf{x}_*, \mathbb{X}, \mathbf{Y} = \mathbf{y} \sim \mathcal{N}\left(\bar{y}_* \,,\, cov(y_*)\right) \tag{7.3.6}$$

where $\bar{y}_* = \boldsymbol{\phi}(\mathbf{x}_*^T)\Sigma_p\Phi(\mathbb{X})(R + \sigma_n^2 I)^{-1}\mathbf{y}$ and
$cov(y_*) = \boldsymbol{\phi}(\mathbf{x}_*^T)\left(\Sigma_p - \Sigma_p\Phi(\mathbb{X})\left(\sigma_n^2 I + \Phi(\mathbb{X})^T\Sigma_p\Phi(\mathbb{X})\right)^{-1}\Phi(\mathbb{X})^T\Sigma_p\right)\boldsymbol{\phi}(\mathbf{x}_*)$.

Bibliography

[1] Laplace approximation to the posterior. Notes 11.

[2] Amir Atiya. Gaussian processes for classification. Technical report, Cairo University, February 2011.

[3] Amir F. Atiya. A new monte carlo based algorithm for the gaussian process classification problem. Technical report, Cairo University, October 2013.

[4] Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer Science+Business Media, 2004.

[5] Dr. Volkan Cevher. Laplace approximation. Technical report, Rice University, September 2008.

[6] Jason Corso. Bayesian decision theory. Class Notes.

[7] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Springer, 2009.

[8] R. M. Dudley. The sizes of compact subset of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1:290–330, 1967.

[9] Mark Ebden. Gaussian processes for classification: A quick introduction. August 2008.

[10] Mark Ebden. Gaussian processes for regression: A quick introduction. August 2008.

[11] Brian S. Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. Wiley, fifth edition, 2011.

[12] Chao Gao and Harrison H. Zhou. Bernstein-von mises theorems for functionals of covariance matrix. November 2014.

[13] Zoubin Ghahramani. A tutorial on gaussian processes (or why i don't use svms). Technical report, University of Cambridge, 2011.

[14] Michael I. Jordan. The kernel trick. Advanced Topics in Learning and Decision Making.

[15] Alexandros Karatzoglou. kernlab – an s4 package for kernel methods in r. Technical report, Technische Universität Wien.

[16] M. Kuss and C. E. Rasmussen. Assessing approximations for gaussian process classification. Technical report, Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany.

[17] Steven P. Lalley. Introduction to gaussian processes.

[18] Henry Lin. Clustering. 15-831 Artificial Intelligence.

[19] David J. C. Mackay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1992.

[20] David J. C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[21] M. B. Marcus and L.A. Shepp. Sample behavior of gaussian processes.

[22] M. B. Marcus and L.A. Shepp. Continuity of gaussian processes. *Transactions of the American Mathematical Society*, 151(2):377–391, October 1970.

[23] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[24] Thomas P. Minka. Bayesian linear regression. 1998.

[25] Radford M. Neal. Bayesian training of backpropagation networks by the hybrid monte carlo method. Technical report, University of Toronto, 1992.

[26] A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society*, 40(1), 1978.

[27] Carl E. Rasmussen and Christopher K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[28] Erlis Ruli, Nicola Sartori, and Laura Ventura. Improved laplace approximation for marginal likelihood. February 2015.

[29] Matthias Seeger. Relationships between gaussian processes, support vector machines and smoothing splines. Technical report, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, December 2002.

[30] Dino Sejdinovic and Arthur Gretton. What is an rkhs? March 2012.

[31] Prof. Dr. Evgeny Spodarev. Random fields i. Lecture Notes, 2009.

[32] Erik Sudderth. Expectation propagation. Notes Week 11 Summary, November 2002.

[33] Peter Tryfos. *Cluster Analysis*, chapter 15. 1997.

[34] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition edition, 1983.

[35] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, October 1998.

[36] Prof. Alan Yuille. Bayes decision theory.