Detecting Fraud in Cellular Telephone Networks

Johan H van Heerden

Thesis presented for the degree

Master of Science

in the inter-departmental programme of Operational Analysis

University of Stellenbosch, South Africa

Supervisor: Prof JH van Vuuren

December 2005

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: ______________________ Date: ______________________

Abstract

Cellular network operators globally loose between 3% and 5% of their annual revenue to telecommunications fraud. Hence it is of great importance that fraud management systems are implemented to detect, alarm, and shut down fraud within minutes, minimising revenue loss. Modern proprietary fraud management systems employ (i) *classification methods*, most often artificial neural networks learning from classified call data records to classify new call data records as fraudulent or legitimate, (ii) *statistical methods* building subscriber behaviour profiles based on the subscriber's usage in the cellular network and detecting sudden changes in behaviour, and (iii) *rules and threshold values* defined by fraud analysts, utilising their knowledge of valid fraud cases and the false alarm rate as guidance. The purpose of this thesis is to establish a context for and evaluate the performance of well-known data mining techniques that may be incorporated in the fraud detection process.

Firstly, a theoretical background of various well-known data mining techniques is provided and a number of seminal articles on fraud detection, which influenced this thesis, are summarised. The cellular telecommunications industry is introduced, including a brief discussion of the types of fraud experienced by South African cellular network operators.

Secondly, the data collection process and the characteristics of the collected data are discussed. Different data mining techniques are applied to the collected data, demonstrating how user behaviour profiles may be built and how fraud may be predicted. An appraisal of the performances and appropriateness of the different data mining techniques is given in the context of the fraud detection process.

Finally, an indication of further work is provided in the conclusion to this thesis, in the form of a number of recommendations for possible adaptations of the fraud detection methods, and improvements thereof. A combination of data mining techniques that may be used to build a comprehensive fraud detection model is also suggested.

Opsomming

Sellulêre netwerk operateurs verloor wêreldwyd tussen 3% en 5% van hul jaarlikse inkomste as gevolg van telekommunikasie bedrog. Dit is dus van die uiterse belang dat bedrog bestuurstelsels geïmplimenteer word om bedrog op te spoor, alarms te genereer, en bedrog binne minute te staak om verlies aan inkomste tot 'n minimum te beperk. Moderne gepatenteerde bedrog bestuurstelsels maak gebruik van (i) *klassifikasie metodes*, mees dikwels kunsmatige neurale netwerke wat leer vanaf geklassifiseerde oproep rekords en gebruik word om nuwe oproep rekords as bedrog-draend of nie bedrog-draend te klassifiseer, (ii) *statistiese metodes* wat gedragsprofiele van 'n intekenaar bou, gebaseer op die intekenaar se gedrag in die sellulêre netwerk, en skielike verandering in gedrag opspoor, en (iii) *reëls en drempelwaardes* wat deur bedrog analiste daar gestel word, deur gebruik te maak van hulle ondervinding met geldige gevalle van bedrog en die koers waarteen vals alarms gegenereer word. Die doel van hierdie tesis is om 'n konteks te bepaal vir en die werksverrigting te evalueer van bekende data ontginningstegnieke wat in bedrog opsporingstelsels gebruik kan word.

Eerstens word 'n teoretiese agtergrond vir 'n aantal bekende data ontginningstegnieke voorsien en 'n aantal gedagteryke artikels wat oor bedrog opsporing handel en wat hierdie tesis beïnvloed het, opgesom. Die sellulêre telekommunikasie industrie word bekend gestel, insluitend 'n kort bespreking oor die tipes bedrog wat deur Suid-Afrikaanse sellulêre telekommunikasie netwerk operateurs ondervind word.

Tweedens word die data versamelingsproses en die eienskappe van die versamelde data bespreek. Verskillende data ontginningstegnieke word vervolgens toegepas op die versamelde data om te demonstreer hoe gedragsprofiele van gebruikers gebou kan word en hoe bedrog voorspel kan word. Die werksverrigting en gepastheid van die verskillende data ontginningstegnieke word bespreek in die konteks van die bedrog opsporingsproses.

Laastens word 'n aanduiding van verdere werk in die gevolgtrekking tot hierdie tesis verskaf, en wel in die vorm van 'n aantal aanbevelings oor moontlike aanpassings en verbeterings van die bedrog opsporingsmetodes wat beskou en toegepas is. 'n Omvattende bedrog opsporingsmodel wat gebruik maak van 'n kombinasie van data ontginningstegnieke word ook voorgestel.

Terms of Reference

This thesis was initiated as an investigation into the fraud detection and prevention arena after being awarded the opportunity to be involved with the implementation of a fraud management system at one of South Africa's cellular network operators. Early on in the project it became apparent that South African fraud management systems rely on the intuition of fraud analysts and their experience with fraudulent behaviour for defining fraud detection rules and threshold values rather than on objective scientific methods and techniques, resulting in large numbers of false alarms and undetected fraud. Modern fraud management systems make use of data mining techniques in the fraud detection process — not to detect fraud, but rather to confirm fraud detected by rule-based methods, or to assess the severity of detected fraud. Data mining techniques may aid fraud analysts to define fraud detection rules and threshold values, including classification methods able to classify call data records as fraudulent or legitimate and clustering methods grouping subscribers into behaviour profiles. The purpose of this thesis is to establish a context and evaluate the performance of these well-known data mining techniques in the fraud detection process.

Prof JH van Vuuren was the supervisor to the author when working on this thesis. The call data records used in this thesis, as well as insight into fraud detection and prevention processes, were provided by a South African cellular network operator, which has requested to remain anonymous. Work on this thesis commenced in February 2002 and was completed in May 2005.

Acknowledgements

The author hereby wishes to express his gratitude towards

- the anonymous South African cellular network operator, for the call data records they provided and for insight into their fraud detection processes;
- his employer, for their interest, understanding and financial support;
- prof JH van Vuuren, for his guidance, patience and dedication;
- friends and family, for their encouragement and support.

Glossary

Activation Function: A mathematical function within the *neuron* of a *neural network* that translates the summed score of the weighted input values into a single output value.

Adjusted Coefficient of Determination: A modified measure of the *coefficient of determination* that takes into account the number of *explanatory variables* included in a regression equation.

Agglomerative Hierarchical Method: A clustering procedure that begins with each *observation* in a separate cluster. In each subsequent step, two clusters that are most similar are combined to build a new cluster of observations.

Apriori Algorithm: A data mining algorithm for mining *frequent item sets* for boolean *association rules.*

Apriori Property: A property used to reduce the search space and improve the generation of *frequent item sets* in the process of *association rule mining.*

Artificial Neural Network: See *Neural Network.*

Association Measure: A measure of similarity used in *cluster analysis* representing similarity as the correspondence of patterns across variables measured in nonmetric terms.

Association Rule: A rule based on the correlation between sets of items is a data set.

Association Rule Mining: The process of mining for *association rules.*

Backpropagation: The most common learning process in *neural networks*, in which errors in estimating the output *nodes* are propagated back though the *neural network* and used to adjust the weights for each *node.*

Bayesian Classification: See *Bayesian Decision Making.*

Bayesian Decision Making: A fundamental statistical approach which aids in the design of an optimal classifier if the complete statistical model governing a set of observations is known.

Bayesian Network: A graphical model of causal relationships that allows class conditional dependencies to be defined between subsets of variables.

Base Station Controller: The part of a cellular telecommunications network's infrastructure that performs radio signal management functions for *base transceiver stations*, managing functions such as frequency assignment.

Base Station Subsystem: A subsystem in the cellular telecommunications network that refers to the combined functions of the *base transceiver station* and *base station controller.*

Base Transceiver Station: The name for the antenna and radio equipment necessary to provide cellular telecommunication service in an area.

Belief Network: See *Bayesian Network.*

Call Data Record: A record of a placed call. Call data records include the time when the call was placed and the duration of the call.

Call Selling: A method used by fraudsters as a means of setting up their own cut-price telephone service which they then proceed to sell — typically to fraudsters, to illegal immigrants or to refugees.

Cellular Telephone: See *Mobile Station.*

Class Assignment Rule: A rule assigning a class to every *terminal node* in a *classification tree.*

Classification: In classification-type problems one attempts to predict values of a categorical *response variable* from one or more *explanatory variables.*

Classification Tree: See *Decision Tree.*

Cloning: A technique used by fraudsters as a means of gaining free access to a cellular telecommunications network whereby a cellular telephone is reprogrammed to transmit the electronic serial number and telephone number belonging to another legitimate subscriber.

Cluster Analysis: A multivariate statistical technique which assesses the similarities between units or assemblages, based on the occurrence or non-occurrence of specific artifact types or other components within them.

Coefficient of Determination: A measure of the proportion of the variance of the *response variable* about its mean that is explained by the *explanatory variables.*

Conditional Mean: The mean value of the *response variable*, given the value of the *explanatory variables* in a regression equation.

Confidence: The proportion of times two or more *item sets* occur jointly during the process of *association rule mining*.

Correlation Measure: A measure of similarity used in *cluster analysis* representing similarity as the correspondence of patterns across the variables.

Data Mining: The exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules.

Decision Tree: A rule-based model consisting of nodes and branches that reaches multiple outcomes, based on passing through two or more nodes.

Descendent: If an arc is present from a node t to a node t_d in a *belief network*, then t_d is called a descendent of node t.

Deviation: A statistic used in *logistic regression* to determine how well a *logistic regression* model fits the data.

Deviation-based Outlier Detection: An outlier detection technique identifying outliers by examining the main characteristics of observations in a group. Observations that deviate from this description are considered outliers.

Discordancy Test: A test examining two hypotheses, a working hypothesis and an alternative hypothesis. The hypothesis is retained if there is no statistically significant evidence supporting its rejection.

Distance-based Outlier: An outlier detection technique identifying outliers by examining the distance between observations in a group. An observation is a distance-based outlier if a fraction of the observations in the group lie a distance larger than some threshold value from the observation.

Distance Measure: A measure of similarity used in *cluster analysis* representing similarity as the proximity of *observations* to one another across the variables.

Divisive Hierarchical Method: A clustering procedure that begins with all *observations* in a single cluster, which is then divided at each step into two clusters containing the most dissimilar observations.

Explanatory Variable: The independent variable in regression analysis.

Equipment Identity Register: A database used to verify the validity of equipment being used in cellular telecommunications networks. It may provide security features such as blocking of calls from stolen cellular phones and preventing unauthorised access to the network.

***F* Statistic:** See *F-Test.*

***F*-test:** A statistical test for the additional contribution to the prediction accuracy of a variable above that of the variables already in the regression equation.

***F*-to-enter Value:** The minimum *F*-test value required when deciding on adding additional *explanatory variables* to the regression equation during the forward variable selection procedure.

***F*-to-remove Value:** An *F*-test value used to decide when to stop removing *explanatory variables* from the regression equation when employing the backward elimination variable selection procedure.

Feedforward Neural Network: A *neural network* where *nodes* in one layer are connected only to *nodes* in the next layer, and not to *nodes* in a preceding layer or *nodes* in the same layer.

Forward Model Selection: A method of variable selection in which variables are added to the model sequentially until the gain from adding another conditioning variable is insignificant.

Fraud Detection: The use of scientific tools to detect compromises to a cellular telecommunications network as part of a fraud management strategy.

Fraud Deterrence: Measures put in place to deter fraudsters from committing fraud implemented as part of a fraud management strategy.

Fraud Prevention: The process of erecting obstacles for unauthorised access to an operator's network and systems as part of a fraud management strategy.

Frequent Item Set: An *item set* satisfying minimum *support* in the process of *association rule mining.*

Global System for Mobile Communications: A digital cellular telecommunications technology deployed in Europe, North America and South Africa.

Goodness of Split: The decrease in impurity when a parent node is splitted into two descendent nodes during *classification tree* construction.

Handset: See *Mobile Station.*

Hidden Node: A *node* in one of the hidden layers of a *multilayer neural network*. It is the hidden layers and *activation function* that allow *neural networks* to represent nonlinear relationships.

Home Location Register: A database residing in a cellular telecommunications network containing subscriber and service profiles and used to confirm the identity of local subscribers.

Immediate Predecessor: If an arc is present from a node t to a node t_d in a *belief network*, then t is called an immediate predecessor of node t_d.

Impurity Function: A function calculating the ability of a *classification tree* node to distinguish between different classes.

Impurity Measure: The value returned by an *impurity function* and referred to as the *goodness of split* in *classification tree* construction.

Index-based Algorithm: An algorithm used in distance-based outlier detection employing multidimensional indexing structures, such as *R-trees* or *KD-trees*.

Input Node: A *node* in the first layer of a *multilayer neural network* representing a single variable or pattern.

Input Processor: See *Input Node*.

Intermediate Processor: See *Hidden Node*.

Internal Estimate: An estimate of classifier accuracy calculated as the proportion of *observations* misclassified when the classifier is applied to a sample of *observations* drawn from the same population from which the *learning sample* was drawn.

International Mobile Equipment Identity: A unique 15-digit number that serves as the serial number of a *cellular telephone*.

International Mobile Subscriber Identity: A unique 15-digit number that identifies a subscriber.

Item Set: A subset of items employed in the process of *association rule mining*.

KD Tree: A binary tree that recursively partitions an input space into parts, in a manner similar to a *decision tree*, acting on real-valued inputs.

Kullback-Leibler Distance: A measure of distance between two probability distributions. It may be described as the difference between the cross entropy of the two probability distributions and the entropy of one of them.

Law of Iterated Probability: A theorem stating that a multivariate distribution may be expressed as the product of marginal and conditional probility distributions.

Learning Sample: A sample of *observations* used in the learning process of data mining techniques.

Linear Regression: A statistical technique that may be used to predict the value of a *response variable* from known values of one or more *explanatory variables.*

Logistic Regression: A technique for predicting a binary *response variable* from known values of one or more *explanatory variables.*

Memorandum of Understanding: An agreement signed between all the major *global system for mobile communications* (GSM) operators to work together to promote GSM.

Min-Max Normalisation: A normalisation technique performing a linear transformation on a set of data, scaling it to a specific range, such as $[0.0, 1.0]$.

Minimal Cost-Complexity Method: A method of *classification tree* pruning, measuring tree complexity as the number of terminal nodes in the tree.

Minimum Support: See *Support.*

Minimum Support Count: The number of transactions required for an *item set* to satisfy *minimum support* in the process of *association rule mining.*

Minkowski Metric: A method of measuring the distance between two points in P dimensions using a variable scaling factor. When the scaling factor is 1 this metric measures the rectilinear distance between two points, and it measures the Euclidean distance when the scaling factor is 2.

Multivariate Analysis: A generic term used for a statistical technique that analyses a multidimensional data set.

Mobile: See *Mobile Station.*

Mobile Station: A station in a cellular telecommunications network intended to be used while in motion or during halts at unspecified points.

Mobile Subscriber Integrated Services Digital Network: The number used to call a cellular subscriber. This number consists of a country code, a *national destination code* and a subscriber number.

Mobile Switching Centre: A central switch that controls the operation of a number of *base stations.* It is a sophisticated computer that monitors all cellular calls, tracks the location of all *cellular telephones* in the system and keeps track of billing information.

National Destination Code: Part of the mobile subscriber integrated services digital network number used to identify a subscriber's cellular network operator.

Network Subsystem: A subsystem in a cellular telecommunications network that refers to the *mobile switching centre* and network registers.

Neural Network: A nonlinear predictive weighted graph model that learns through sequential processing of large samples of *observations* during which the classification errors are used to adjust weights to improve estimation.

Neuron: A node or basic building block in a *neural network.*

Node: See *Neuron.*

Observation: A record or object in a data set made up of various attributes describing the object.

Outlier: An *observation* that is substantially different from the other *observations.*

Outlier Analysis: A technique used to identify data *observations* that do not comply with the general behaviour of the data set.

Output Node: A *node* in the final layer of a multilayer *neural network* representing class membership.

Output Processor: See *Output Node.*

Overlapping Calls Detection: A fraud detection technique identifying calls from the same cellular subscriber overlapping in time in an attempt to detect the existence of two cellular telephones with identical identification codes.

Parent: See *Immediate Predecessor.*

Personal Identification Number: A code used by a *cellular telephone* in conjunction with a subscriber identity module (SIM) card to complete a call.

Premium Rate Service Fraud: A type of fraud involving a large number of calls to a premium rate service number from a subscriber's account without their knowledge.

Principle Component Analysis: The process of identifying a set of variables that define a projection encapsulating the maximum amount of variation in a data set and is orthogonal to the previous principle component of the same data set.

Probabilistic Network: See *Bayesian Network*.

Public Switched Telephone Network: The traditional landline network that cellular telecommunications networks often connect with to complete calls.

R-Tree: A tree data structure used by spatial access methods, such as indexing multi-dimensional information.

Receiver Operating Characteristic: A graphical plot of the fraction of true positives versus the fraction of false positives for a binary classifier system as its discrimination threshold is varied.

Regression: The process of attempting to predict the values of a continuous *response variable* from one or more *explanatory variables*.

Residual Mean Square: A measure of how well a regression curve fits a set of data points.

Response Variable: The dependent variable in regression analysis.

Resubstitution Estimate: An estimate of classifier accuracy using the same sample used to construct the classifier.

Saturated Model: A logistic regression model containing as many parameters as there are *observations*.

Sequencial Exception Technique: One of the techniques used in deviation-based outlier detection, simulating the way in which humans are able to distinguish unusual observations from among a series of supposedly-like observations.

Short Message Service: The transmission of short alphanumeric text-messages to and from a *cellular telephone*. These messages may be no longer than 160 alphanumeric characters and contain no images or graphics.

Signature: A multivariate probability distribution describing customer behaviour.

Similarity Coefficient: An indication of similarity between *observations* based on the presence or absence of certain characteristics.

Statistical-Based Outlier Detection: An approach to outlier detection assuming a distribution or probability model for the given data set, and which identifies outliers with respect to the model, using a discordancy test.

Strong: *Association rules* that satisfy both the *minimum support* threshold and the minimum *confidence* threshold are called strong.

Subscriber Identity Module: A card inserted into a *cellular telephone* containing subscriber-related data.

Subscription Fraud: Fraud occurring when a subscriber signs up for a service with fraudulently obtained subscriber information, or false identification.

Sum of Squared Errors: The sum of the squared prediction errors across all *observations.* It is used to denote the variance in the *response variables* not yet accounted for by a regression model.

Sum of Squared Regression: Sum of the squared differences between the mean and predicted values of the *response variable* for all *observations* in a regression equation.

Supervised Learning: The process in a neural network implementation where a known target value is associated with each input in the training set.

Support: The percentage of the total sample for which an *association rule* is valid.

Terminal Node: A node in a *classification tree* for which further splitting will not result in a decrease in impurity.

Test Sample Estimate: An estimate of classifier accuracy dividing the *learning sample* into two subsets, using one set to construct the classifier and the other to obtain the estimate.

Total Sum of Squares: Total amount of variation in the response variable of a regression equation that exists and needs to be explained by the *explanatory variables.*

Training Phase: A phase of a *neural network* implementation during which learning takes place through sequential processing of large samples of *observations* in which the classification errors are used to adjust weights in order to improve estimation.

Tumbling: A technique used by fraudsters, switching between captured *cellular telephone* identification numbers to gain access to the cellular telecommunications network.

Unsupervised Learning: The process in a neural network implementation where learning occurs when the training data lack target output values corresponding to input patterns.

V-fold Cross-Validation: A method of estimating classifier accuracy by dividing the *learning sample* into v subsets of approximately equal size, using as *learning sample* all of the subsets bar one to construct a classifier, repeated across all v subsets exclusions.

Velocity Traps: A fraud detection technique testing for call origin locations geographically far appart, but in temporal proximity.

Visitor Location Register: A network database that holds information about cellular customers using an operators's cellular telecommunications network but not subscribing to that cellular operator.

List of Reserved Symbols

α	the level of significance used during an F-test [variable selection]
a_{ki}	the net input of the i^{th} observation into node k [artificial neural network]
A	a subset of $\mathcal{X}$ obtained by repeated splitting [classification tree]
A_j	a subset of $\mathcal{X}$ for which $d(\mathbf{X}_i)$ predicts membership of class C_j [classification tree]
A_n	a signature component after call n [subscriber behaviour profiling]
$A_I, B_I, \ldots$	sets of items in I [association rule mining]
β_i	the regression coefficient of the i^{th} explanatory variable [regression analysis]
$\hat{\beta}_i$	the estimated value of a regression coefficient β_i [regression analysis]
b_i	the i^{th} category of categorical variable x_{ji} [classification tree]
$b_1(a_{ki}), b_2(a_{ki}),$ $b_3(a_{ki}), b(\cdot)$	a number of different activation functions [artificial neural network]
B	the possible values of categorical variable x_{ji} [artificial neural network]
c	the percentage of transactions in D_I containing A_I that also contain B_I [association rule mining]
C	the possible values ranging between $(-\infty, \infty)$ that continuous variable x_{ji} may take on [classification tree]
C_I	the minimum confidence threshold [association rule mining]
C_j	the j^{th} class in $\mathcal{C}$ [classification tree]
$\mathcal{C}$	a set of J classes [classification tree]
δ_{qi}	the error of the i^{th} observation at output node q [artificial neural network]
$d(\mathbf{X}_i)$	a classifier classifying observations $\mathbf{X}_i$ [classification tree]
$d^{(v)}$	a classifier constructed from the v^{th} subset of learning sample $\mathcal{L}$ using the method of V-fold cross-validation to calculate the internal estimate [classification tree]

d_{uv} the distance or similarity between clusters U and V [cluster analysis]

$d_{(uv)w}$ the minimum of distances or similarities d_{uw} and d_{vw} [cluster analysis]

d_{ik} the distance or similarity between clustered item i and item k [cluster analysis]

$d_1(\mathbf{X}_i, \mathbf{X}_j)$ the Euclidean distance between two P-dimensional observations $\mathbf{X}_i$ and $\mathbf{X}_j$ [cluster analysis]

$d_2(\mathbf{X}_i, \mathbf{X}_j)$ the Minkowski metric between two P-dimensional observations $\mathbf{X}_i$ and $\mathbf{X}_j$ [cluster analysis]

$d_3(\mathbf{X}_i, \mathbf{X}_j)$ the Gower's general similarity coefficient between two P-dimensional observations $\mathbf{X}_i$ and $\mathbf{X}_j$ [cluster analysis]

$d_{4_k}(\mathbf{X}_i, \mathbf{X}_j)$ the contribution to Gower's general similarity coefficient provided by the k^{th} variable in the two P-dimensional observations $\mathbf{X}_i$ and $\mathbf{X}_j$ [cluster analysis]

$\mathbf{D}$ the $N \times N$ symmetric matrix of distances or similarities between observations [cluster analysis]

$D(\mathbf{X}_i, y_i)$ the deviation in prediction accuracy between the current model and saturated model [logistic regression]

$D_o(k, l)$ a distance-based outlier with parameters k and l [outlier analysis]

D^2 a measure of distance between two points in the space defined by two or more correlated variables, also called the Mahalanobis distance [outlier analysis]

D_I a set of database transactions T_I [association rule mining]

ϵ_i the stochastic error at the i^{th} observation [regression analysis]

e_i the prediction error of the regression model at the i^{th} observation [regression analysis]

E the sum of squared errors across all observations [regression analysis]

E_P the sum of squared errors computed with P explanatory variables [regression analysis]

$E(y_i|\mathbf{X}_i)$ the conditional mean of the response variable y_i, given the values of explanatory variables $\mathbf{X}_i$ [logistic regression]

f_{qi} the input of the i^{th} observation into output node q [artificial neural network]

F the F-test statistic calculating the prediction improvement when adding additional explanatory variables to a regression model [variable selection]

F_o the initial distribution of observations [outlier analysis]

$g(\mathbf{X}_i)$ the logistic transformation of the logistic regression model $\pi(\mathbf{X}_i)$ [logistic regression]

$G(\mathbf{X}_i, y_i)$ the difference in deviation of models with and without explanatory variable x_{ji} [logistic regression]

h_{ki} the input of the i^{th} observation into hidden node k [artificial neural network]

H the total number of hidden nodes [artificial neural network]

H_i a hypothesis [variable selection]

$\overline{H}_i$ an alternative hypothesis [outlier analysis]

$i(t)$ the impurity measure of node t [classification tree]

i_i the i^{th} item in the universal set of items I [association rule mining]

$\Delta i(s,t)$ the decrease in impurity caused by candidate split s at tree node t [classification tree]

I a universal set of items [association rule mining]

$\mathcal{I}(\cdot)$ an indicator function defined to be 1 if the statement between the parenthesis is true, and 0 otherwise [artificial neural network]

$I(T)$ the impurity measure of tree T [classification tree]

J the number of classes contained among the response values in y_i [classification tree]

K_{I_k} a set of candidate k-item sets [association rule mining]

$l(\beta_1, \ldots, \beta_P)$ the likelihood function [logistic regression]

l_{I_i} an item set in L_I [association rule mining]

$l_{I_i}[j]$ the j^{th} item in item set l_{I_i} [association rule mining]

$\mathcal{L}$ a learning sample used when constructing data mining models

$\mathcal{L}_i$ a subset of the learning sample $\mathcal{L}$ [classification tree]

$L(\beta_1, \ldots, \beta_P)$ the log of the likelihood function $l(\beta_1, \ldots, \beta_P)$ [logistic regression]

L_I a frequent item set in association rule mining [association rule mining]

m_k the k^{th} subscriber in the set of observations [association rule mining]

M the number of input nodes [artificial neural network]

M^2 the residual mean square for estimating prediction accuracy [regression analysis]

N the number of observations in a data set

N_{C_i} the number of observations of class C_i [Bayesian decision making]

$N_{C_{ji}}$ the number of observations of class C_i having the value x_{ji} [Bayesian decision making]

N_{M_l} the maximum number of observations within radius l of an outlier [outlier analysis]

N_s the total number of subsets of the observations in $\mathcal{X}$ [outlier analysis]

N_C the confidence count [association rule mining]

N_S the support count [association rule mining]

$N(0, \sigma^2)$ a normal distribution with mean 0 and variance σ^2 [regression analysis]

η a factor scaling the step size when updating weights [artificial neural network]

O the total number of output nodes [artificial neural network]

$O(t)$ the set of parents of node t [Bayesian network]

$p_j(t)$ the proportion of observations at tree node t belonging to class C_j [classification tree]

p_R the proportion of observations in node t sent to node t_R by candidate split s [classification tree]

p_L the proportion of observations in node t sent to node t_L by candidate split s [classification tree]

$p(t)$ the resubstitution estimate of the probability that any observation falls into node t [classification tree]

P the number of explanatory variables in one observation

$P[H_i|\mathbf{X}_i]$ the conditional probability that the hypothesis H_i holds given the observation $\mathbf{X}_i$ [Bayesian decision making]

$P_{w_{ijq}}$ a conditional probability table entry [Bayesian network]

P_w the set of conditional probability table entries [Bayesian network]

P_{P_w} the probability of prediction accuracy under the conditional probability table P_w [Bayesian network]

$P_s[v_i]$ the significance probability of the value of the test statistic T_s on observation $\mathbf{X}_i$ [outlier analysis]

P_{m_k} the probability distribution describing the behaviour of subscriber m_k as a series of probabilities of cluster membership [association rule mining]

Q the number of additional explanatory variables available [variable selection]

$\mathcal{Q}$ a set of binary questions used during tree construction [classification tree]

$r(t)$ the resubstitution estimate of the probability of misclassification [classification tree]

R_g	the sum of squared regression across all observations of a regression model [variable selection]
R^2	the coefficient of determination for estimating prediction accuracy in regression models [variable selection]
$\overline{R}^2$	the adjusted coefficient of determination for estimating prediction accuracy in regression models [variable selection]
$R_c^*(d)$	the rate of misclassification when applying classifier d to a set of observations [classification tree]
$R_c(d)$	the internal estimate of $R_c^*(d)$ [classification tree]
$R_c^*\left(d^{(v)}\right)$	the rate of misclassification when applying classifier $d^{(v)}$ to a set of observations [classification tree]
$R_c^*(T)$	the rate of misclassification achieved in tree T [classification tree]
$R_c(t)$	the rate of misclassification achieved in node t of tree T [classification tree]
$R_{c\gamma}(T)$	the cost-complexity measure [classification tree]
$R_{r\gamma}(T)$	the error-complexity measure [regression tree]
s	a candidate split at tree node t [classification tree]
s^*	the candidate split s at tree node t yielding the largest decrease in impurity [classification tree]
$s^{'}$	the best surrogate split of candidate split s [classication tree]
s_I	a non-empty subset of frequent item set l_I [association rule mining]
$\mathcal{S}$	the set of candidate splits s at tree node t [classification tree]
S_y^2	the variance of response variables y in a sample of observations [variable selection]
S_i	a subset of the observations in $\mathcal{X}$ [outlier analysis]
S_I	the minimum support threshold [association rule mining]
t	a node in binary tree T [classification tree]
t_d	a descendent of node t [Bayesian network]
t_{d_j}	the j^{th} descendent of node t [Bayesian network]
t_R	the descendant to the right of tree node t [classification tree]
t_L	the descendant to the left of tree node t [classification tree]
$\{t_1\}$	a tree consisting of the root node [classification tree]
T	a binary tree [classification tree]
$\widetilde{T}$	the current set of terminal nodes in the binary tree T [classification tree]
T_t	a branch of tree T with root node t [classification tree]
$T^{'}$	a pruned subtree of tree T [classification tree]
$\lvert\widetilde{T}\rvert$	the number of terminal nodes in tree T [classification tree]

T_S	the total sum of squares across all observations of the regression model [variable selection]
T_s	a test statistic used during discordancy testing [outlier analysis]
T_I	a database transaction consisting of a set of items [association rule mining]
τ	a point in time denoting the onset of positive activity [subscriber behaviour profiling]
μ_k	a threshold value for node k [artificial neural network]
v_{ki}	the output of the i^{th} observation from the hidden node k [artificial neural network]
v_i	the value of the test statistic T_s on observation $\mathbf{X}_i$ [outlier analysis]
ϖ	the threshold value used in a fraud scoring function [subscriber behaviour profiling]
w	the rate at which old calls are aged out from a signature component [subscriber signature design]
w_{kj}	the strength of the connection from the j^{th} node to the k^{th} node [artificial neural network]
W_{qk}	the strength of the connection from hidden node k to output node q [artificial neural network]
W_{ijk}	the validity indicator of the comparison between the k^{th} variable in $\mathbf{X}_i$ and $\mathbf{X}_j$ [cluster analysis]
x_{ji}	the value of the j^{th} explanatory variable at the i^{th} observation
$\mathbf{x}_j$	N values of the i^{th} explanatory variable
$\mathbf{X}_i$	a vector of P explanatory variables of the i^{th} observation
$\mathbf{X}$	a vector of N observations
$\mathcal{X}$	the measurement space containing all possible measurement vectors $\mathbf{X}_i$
$\tilde{X}_i$	a model without explanatory variable x_{ji} [logistic regression]
y_i	the value of the response variable at the i^{th} observation
$\hat{y}_i$	the predicted value of the response variable at the i^{th} observation
$\overline{y}$	the average value of all the response variables
y_{qi}	the observed response of the i^{th} observation belonging to class q [artificial neural network]
$\hat{y}_{qi}$	the response of the i^{th} observation at output node q [artificial neural network]
$\mathbf{Y}$	the vector of N response values in the set of observations
$\pi(\mathbf{X}_i)$	the model on explanatory variables $\mathbf{X}_i$ [logistic regression]
$\hat{\pi}(\mathbf{X}_i)$	the maximum likelihood estimate of $\pi(\mathbf{X}_i)$ [logistic regression]

ψ_{ki}	the error of the i^{th} observation at hidden node k [artificial neural network]
ϕ	an impurity function calculating tree node impurity [classification tree]
γ	the complexity parameter [classification tree]
$\zeta(\mathbf{X}_i, y_i)$	the contribution of the pair $(\mathbf{X}_i, y_i)$ to the likelihood function [logistic regression]

List of Acronyms

BSC:	Base Station Controller
BTS:	Base Transceiver Station
CDR:	Call Data Record
EIR:	Equipment Identity Register
GSM:	Global System for Mobile Communications
HLR:	Home Location Register
IMEI:	International Mobile Equipment Identity
IMSI:	International Mobile Subscriber Identity
MoU:	Memorandum of Understanding
MSC:	Mobile Switching Centre
MSISDN:	Mobile Subscriber Integrated Services Digital Network
NDC:	National Destination Code
PIN:	Personal Identification Number
ROC:	Receiver Operating Characteristic
PSTN:	Public Switched Telephone Network
SIM:	Subscriber Identity Module
SMS:	Short Message Service
VLR:	Visitor Location Register

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Fraud in Mobile Telecommunication Networks

Fraud in a mobile telecommunication network refers to the illegal access of the network and the subsequent use of its services. The development of intelligent data analysis methods for fraud detection may certainly be motivated from an economic point of view. Additionally, the reputation of a network operator may suffer from an increasing number of fraud cases. The Business Day of 20 March 2003 [9] reported that globally, mobile telecommunication fraud is bigger business than international drug trafficking, with operators worldwide typically losing US $55bn a year. It is the single largest cause of revenue loss for operators, costing them between 3% and 5% of their annual revenue. In Africa alone, carriers write off R700m a year to fraud, which is expected to increase since more than thirty million Africans have access to cellular telephones, providing criminals with a very large wireless market to infiltrate [9].

Historically, earlier types of fraud involved use of technological means to acquire free access to the mobile telecommunication network. *Cloning* of cellphones by creating copies of handsets with identification numbers from legitimate subscribers was typically used as a means of gaining free access to the network. In the era of analog handsets, identification numbers could be captured easily by eavesdropping with suitable receiver equipment in public places, where cellphones were evidently used. One specific type of fraud, called *tumbling*, was quite prevalent in the United States. It exploited deficiencies in the validation of subscriber identity when a cellphone subscription was used outside the subscriber's home area. The fraudster kept switching between captured identification numbers to gain access. Early fraud detection systems examined whether two instances of one subscription were used at the same time — this was called the *overlapping calls detection mechanism*. Detection systems testing for call origin locations geographically far apart, but in temporal proximity, were called *velocity traps*. Both the *overlapping calls* and the *velocity trap* methods attempted to detect the existence of two cellphones with identical identification codes, clearly evidencing cloning. As a countermeasure to these fraud types, technological improvements were introduced [22]. However, new forms of fraud also came into existence. One of the growing types of fraud in South Africa is so-called *subscription fraud*. In subscription fraud, a fraudster obtains a subscription, possibly with false identification, and starts a fraudulent activity with no intention to pay the bill. Another kind of subscription fraud, known to most South Africans, is the theft of cellphones, where offenders steal cellphones and use them to make calls until the theft is reported and the handset is locked by the service provider. In September 2001 the media reported that MTN, one of South Africa's cellular service providers, receives on average 5 700 reported thefts of cellphones every month [32].

One way that operators may fight back is by installing fraud prevention software to detect usage anomalies quickly. Callers are dissimilar, so calls that look like fraud for one account, may be expected behaviour for another. Fraud detection must therefore be tailored to each account's own activity. However, a change in behaviour patterns is a common characteristic in nearly all fraud scenarios.

1.2 Problem Description and Thesis Objectives

The mobile telecommunication industry suffers major losses each year due to fraud, as mentioned in §1.1. Because of the direct impact of fraud on the bottom-line of network operators, the prevention and detection of fraud has become a priority. Subscription fraud is currently a major form of fraud, but as fraud detection software becomes more successful in detecting and preventing this kind of fraud, criminals are likely to discover new techniques to defraud service providers and their customers.

Modern computerised fraud management systems implement a combination of different proprietary fraud detection techniques, each one contributing to a subscriber's fraud weight, typically generating an alarm when the fraud weight exceeds a user-defined threshold value. Classification techniques — most often artificial neural networks — are routinely included in modern fraud management systems; such systems usually are not used to detect fraud, but rather to confirm fraud detected by other techniques. Fraud management systems achieve behaviour profiling by grouping subscribers according to the product to which they subscribe, thereby assuming that subscribers subscribing to a certain product exhibit similar behaviour. Detection rules and threshold values, being the heart of most fraud detection strategies, are defined by fraud analysts using a method of trial and error.

In this thesis the focus is on the use of well-known data mining techniques in the fraud detection process. The following objectives have been set:

1. To employ classification, clustering, association and probabilistic techniques to build

models for use in the fraud detection process.

2. To establish a context for well-known data mining techniques in fraud detection.

3. To evaluate and compare the performances of various data mining methodologies typically employed to detect fraud. Performance may be measured by the fraud detection rate and the false alarm rate.

4. To suggest a combination of data mining techniques that may be used to build a comprehensive fraud detection model that is capable of outperforming models based on a single data mining methodology.

1.3 Layout of Thesis Structure

This thesis consists of seven chapters. Various basic data mining methodologies used in building cellular telephone user behaviour profiles and detecting fraud are described in Chapter 2. In Chapter 3, the nature and operation of the cellular telecommunications industry is described. The chapter proceeds with a discussion of the types of fraud experienced by South African network operators, and the methods they employ to detect and prevent these fraud types. In Chapter 4, a number of seminal articles related to fraud detection in specifically cellular telecommunication networks, which influenced this thesis, are summarised. Chapter 5 provides insight into the call data collection process and the characteristics of the collected data. Chapter 6 forms the core of the thesis, where different data mining methods are applied to real data, demonstrating how user behaviour profiles may be built and how fraud may be predicted. The chapter also contains an appraisal of the performance and appropriateness of the different data mining methods in the context of the fraud detection process. A number of conclusions and recommendations are made in Chapter 7. A combination of data mining techniques are suggested in the chapter that may be used in conjunction with each other to build a comprehensive fraud detection model capable of outperforming models based on a single data mining methodology.

Chapter 2

Data Mining Methodologies

Berry, *et al.* [4] define *data mining* as the exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules. The statistical techniques of data mining include *linear and logistic regression*, *multivariate analysis*, *principle component analysis*, *decision trees*, *neural networks*, *Bayesian decision making*, *association rule mining*, *cluster analysis* and *outlier analysis*.

The data mining methodologies employed in this thesis during the analysis of cellular telephone call data and the subsequent model building process are reviewed in this chapter.

2.1 Decision Trees

Hair, *et al.* [20] define the process of constructing decision trees as a sequential partitioning of *observations* to maximise the differences on *response variables* over the different partition sets. The construction of a decision tree is a technique that generates a graphic representation of the model it produces. It is called a decision tree, because the resulting model is presented in the form of a tree structure. Decision tree problems are divided into *classification* problems and *regression* problems. In classification problems one attempts to predict values of a categorical response variable from one or more continuous and/or categorical *explanatory variables*, whilst in regression problems one attempts to predict the values of a continuous variable from one or more continuous and/or categorical explanatory variable(s) [7].

2.1.1 Classification Trees

A classifier or classification rule is a systematic method of predicting to which class an observation belongs, given a set of measurements on each observation. A more precise formulation of what is meant by a classification rule may be achieved by defining the measurements $\mathbf{X}_i = (x_{1i}, x_{2i}, \ldots, x_{Pi})$ as the measurement vector made during observation i of some process. The measurement space $\mathcal{X}$ is defined as containing all possible measurement vectors $\mathbf{X}_i$. Suppose the response variables y_i of the observations fall into J classes $C_1, C_2, \ldots, C_J$, and let $\mathcal{C}$ be the set of classes, $\mathcal{C} = \{C_1, \ldots, C_J\}$. A systematic way of predicting class membership is a rule that assigns a class membership in $\mathcal{C}$ to every measurement vector $\mathbf{X}_i$ in $\mathcal{X}$. That is, given any $\mathbf{X}_i \in \mathcal{X}$, the rule assigns one of the classes in $\mathcal{C}$ to $\mathbf{X}_i$.

A classifier or classification rule is a function $d : \mathcal{X} \mapsto \mathcal{C}$. Another way of viewing a classifier is to define A_j as the subset of $\mathcal{X}$ for which $d(\mathbf{X}_i) = C_j$, that is $A_j = \{\mathbf{X}_i : d(\mathbf{X}_i) = C_j\}$. The sets $A_1, \ldots, A_j$ are disjoint and $\mathcal{X} = \cup_j A_j$. A classifier is therefore a partition of $\mathcal{X}$ into J disjoint subsets, $A_1, \ldots, A_j$, such that, for every $\mathbf{X}_i \in A_j$, the predicted class is C_j.

In systematic classifier construction, past experience is summarised in a *learning sample* $\mathcal{L}$. This consists of the measurement data on N past observations together with their actual classifications, that is a set of data of the form $\mathcal{L} = \{(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_N, y_N)\}$ on N observations, where $\mathbf{X}_i \in \mathcal{X}$ and $y_i \in \mathcal{C}$, $i = 1, \ldots, N$ [8].

Classifier Accuracy Estimates

One way to measure the accuracy of a classifier is to test the classifier on subsequent observations whose correct classifications are known. This may be achieved by constructing d using $\mathcal{L}$, drawing another very large set of observations from the same population from which $\mathcal{L}$ was drawn and then observing the correct classification for each of those observations, and also finding the predicted classification using $d(\mathbf{X}_i)$. Let the proportion misclassified by d be denoted by $R_c^*(d)$. In actual problems, only the data in $\mathcal{L}$ are available, with little prospect of obtaining an additional large sample of classified observations. In such cases $\mathcal{L}$ is used both to construct $d(\mathbf{X}_i)$ and to estimate $R_c^*(d)$. Such estimates of $R_c^*(d)$ are referred to as *internal estimates*.

The least accurate and most commonly used internal estimate is the *resubstitution estimate*. After the classifier d is constructed, the observations in $\mathcal{L}$ are run through the classifier. The proportion of observations misclassified is the resubstitution estimate. An indicator function $\mathcal{I}(\cdot)$ is defined to be one if the statement between the parenthesis is true, and zero otherwise. The resubstitution estimate may then be formulated as

$$R_c(d) = \frac{1}{N} \sum_{(\mathbf{X}_i, y_i) \in \mathcal{L}} \mathcal{I}\left(d(\mathbf{X}_i) \neq y_i\right). \tag{2.1}$$

The problem with the resubstitution estimate is that it is computed using the same data used to construct d, instead of using an independent sample. Using the subsequent value of $R_c(d)$ as an estimate of $R_c^*(d)$ may give an overly optimistic measure of the accuracy of d.

Another internal estimate often used is the *test sample estimate*. Here the observations in $\mathcal{L}$ are partitioned into two sets, $\mathcal{L}_1$ and $\mathcal{L}_2$. Only the observations in $\mathcal{L}_1$ are used to construct d. Then the observations in $\mathcal{L}_2$ are used to estimate $R_c^*(d)$, using the expression in (2.1). The test sample approach has the drawback that it reduces the effective sample size. This is a minor difficulty if the sample size is large.

However, for smaller sample sizes, another method, called *V-fold cross-validation*, is usually preferred. The observations in $\mathcal{L}$ are randomly partitioned into V subsets of approximately equal size, denoted by $\mathcal{L}_1, \ldots, \mathcal{L}_V$. The classification procedure is applied for every $v \in \{1, \ldots, V\}$, using as the learning sample $\mathcal{L} \backslash \mathcal{L}_v$, to obtain a classifier $d^{(v)}(\mathbf{X}_i)$. Since none of the observations in $\mathcal{L}_v$ have been used in the construction of $d^{(v)}$, a test sample estimate for $R_c^*\left(d^{(v)}\right)$ is calculated, using the expression in (2.1). Finally, using the same procedure again, a classifier d is constructed using all observations in $\mathcal{L}$.

Construction of Classification Trees

Tree structured classifiers are constructed by repeated splits of subsets of $\mathcal{X}$ into two descendant subsets, beginning with $\mathcal{X}$ itself. Those subsets which are not split are called terminal subsets. The terminal subsets form a partition of $\mathcal{X}$ and are designated by a class label. The entire construction of a tree revolves around three elements:

1. Selection of the splits.
2. Decisions as to when to declare a node terminal, or to continue splitting it.
3. Assignment of each *terminal node* to a class.

Assume that the measurement vectors have the form $\mathbf{X}_i = (x_{1i}, \ldots, x_{Pi})$. Let $\mathcal{Q}$ be a set of binary questions of the form $\{\text{Is } \mathbf{X}_i \in A?\}$, where $A \subset \mathcal{X}$ is obtained by (possibly repeated) splitting of the space $\mathcal{X}$. The set of questions $\mathcal{Q}$ is defined by adhering to the following rules:

1. Each split depends on the value of a single variable.
2. For each continuous variable x_{ji}, $\mathcal{Q}$ includes all questions of the form $\{\text{Is } x_{ji} \leq C?\}$ for all C ranging over $(-\infty, \infty)$.
3. If x_{ji} is categorical, taking values in $\{b_1, b_2, \ldots, b_L\}$, then $\mathcal{Q}$ includes all questions of the form $\{\text{Is } x_{ji} \in B?\}$ as B ranges over all subsets of $\{b_1, b_2, \ldots, b_L\}$.

The idea is to select each split of a subset so that the data in each of the descendant subsets is purer than the data in the parent subset. A so–called *goodness of split* criterion is derived from a so–called *impurity function* ϕ, defined on the set of all J-tuples $(p_1(t), \ldots, p_J(t))$, where $p_j(t)$, $j \in \{1, \ldots, J\}$, is the proportion of observations at node t, $\mathbf{X}_i \in t$, belonging to class C_j and satisfying $p_j(t) \geq 0$, $\sum_j p_j(t) = 1$, with the properties that:

1. ϕ is a maximum only at the point $\left(\frac{1}{j}, \frac{1}{j}, \ldots, \frac{1}{j}\right)$,
2. ϕ achieves its minimum only at the points $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)$.

Given an impurity function ϕ, an *impurity measure* $i(t)$ at node t is defined as

$$i(t) = \phi\left(p_1(t), p_2(t), \ldots, p_J(t)\right).$$

If a candidate split s at node t sends a proportion p_R of the observations in t to descendant subset t_R and the proportion p_L to descendant subset t_L, the decrease in impurity is defined as

$$\Delta i(s,t) = i(t) - p_R i(t_R) - p_L i(t_L),$$

which is referred to as the goodness of the split s of t. At the node t all candidate splits in $\mathcal{S}$ are considered so as to find a split s^* that yields the largest decrease in impurity, that is for which

$$\Delta i(s^*, t) = \max_{s \in \mathcal{S}} \Delta i(s,t).$$

After a certain amount of splitting has been performed, the set of splits used, together with the order in which they were performed, determines a binary tree T. The current set of terminal nodes are denoted by $\widetilde{T}$. The tree impurity $I(T)$ is defined as

$$I(T) = \sum_{t \in \widetilde{T}} i(t)p(t),$$

where $p(t)$ is the resubstitution estimate of the probability that any observation falls into node t, and is defined by $p(t) = \sum_j p_j(t)$, where $p_j(t)$ is the resubstitution estimate for the probability that an observation will both be in class C_j and falls into node t. Tree growing is terminated when a node t is reached in which no significant decrease in impurity is possible. Such a node t then becomes a terminal node. This may be achieved by setting a threshold $\kappa > 0$, and declaring a node t terminal if

$$\max_{s \in \mathcal{S}} \Delta(s,t) < \kappa. \tag{2.2}$$

A so–called *class assignment rule* assigns a class $C_j,\ j \in \{1, \ldots, J\}$, to every terminal node $t \in \widetilde{T}$. The class assigned to node $t \in \widetilde{T}$ is denoted by $j(t)$. The class assignment of a terminal node is determined by

$$p_j(t) = \max_i \{p_i(t)\},$$

in which case node t is designated as a class C_j terminal node. If the maximum is achieved for two or more different classes, C_j is assigned arbitrarily as any one of the maximising classes.

Some observations in $\mathcal{L}$ may be incomplete in terms of values for certain explanatory variables. This problem may be overcome by the use of surrogate splits. The idea is to define a measure of similarity between any two candidate splits s and s' of a node t. If the best split of t is the candidate split s on the variable x_{ji}, the candidate split s' on the variables other than x_{ji} is found that is most similar to s, and s' is called the best surrogate for s. The second best surrogate, third best, and so on are defined similarly. If an observation does not include x_{ji} in its set of explanatory variables, the decision as to whether it goes to t_L or t_R is made by using the best surrogate split.

The resubstitution estimate of the probability of misclassification, $r(t)$, given that an observation falls into node t, is defined by

$$r(t) = 1 - \max_j \{p_j(t)\}.$$

The resubstitution estimate for the overall misclassification cost, $R_c^*(T)$, is given by

$$\begin{aligned} R_c^*(T) &= \sum_{t \in \widetilde{T}} r(t)p(t) \\ &= \sum_{t \in \widetilde{T}} R_c(t), \end{aligned}$$

where $R_c^*(T)$ is the tree misclassification cost and $R_c(t)$ is the node misclassification cost.

The splitting termination rule, given by inequality (2.2), typically produces unsatisfactory results [8]. A more satisfactory procedure is to grow a very large tree T_{max} by letting the splitting procedure continue until all terminal nodes are either small, or pure, or contain only identical measurement vectors. Here, pure means that the node observations are all in one class. The large tree T_{max}, may then selectively be pruned, producing a sequence of subtrees of T_{max}, and eventually collapsing to the tree $\{t_1\}$ consisting of the root node.

A branch T_t of T with root node $t \in T$ consists of node t and all descendants of t in T. Pruning a branch T_t from T consists of deleting from T all descendants of t, that is, cutting off all of T_t, except its root node. The tree pruned in this way is denoted by $T - T_t$. If T' is obtained from T by successively pruning off branches, then T' is called a pruned subtree of T, denoted by $T' < T$. Even for a moderately sized tree, T_{max}, there is a potentially large number of subtrees and an even larger number of distinct ways of pruning up to the root node $\{t_1\}$. A selective pruning procedure is necessary, that is, a selection of a reasonable number of subtrees, decreasing in size, such that each subtree selected is the best subtree in its size range.

The so–called *minimal cost-complexity method* of pruning results in a decreasing sequence of subtrees. The complexity of any subtree $T < T_{max}$, is defined as $|\widetilde{T}|$, the number of terminal nodes in T. Let $\gamma \geq 0$ be a real number called the complexity parameter and define the cost-complexity measure $R_{c\gamma}(T)$ as

$$R_{c\gamma}(T) = R_c(T) + \gamma|\widetilde{T}|. \tag{2.3}$$

For each value of γ, that subtree $T(\gamma) < T_{max}$ is found which minimises $R_{c\gamma}(T)$, that is

$$R_{c\gamma}\left(T(\gamma)\right) = \min_{T \leq T_{max}} \{R_{c\gamma}(T)\}.$$

If γ is small, the penalty for having a large number of terminal nodes is small and $T(\gamma)$ will be large. As the penalty γ per terminal node increases, the minimising subtrees $T(\gamma)$ will have fewer terminal nodes. For γ sufficiently large, the minimising subtree $T(\gamma)$ will consist of the root node only. The minimal cost-complexity method of pruning results in a decreasing sequence of subtrees $T_1 > T_2 > \ldots > \{t_1\}$, where $T_k = T(\gamma_k)$ and $\gamma_1 = 0$. The problem is now reduced to selecting one of these subtrees as the optimum-sized tree [8]. The best subtree, T_{k_0}, is a subtree minimising the estimate of the misclassification cost.

2.1.2 Regression Trees

In regression, an observation consists of data $(\mathbf{X}_i, y_i)$ where $\mathbf{X}_i$, the measurement vector, lies in a measurement space $\mathcal{X}$, and y_i, the response variable of the i^{th} observation, is a real-valued number. With regression, construction of a predictor $d(\mathbf{X}_i)$ and the determination of its accuracy are achieved in the same way as in classifier construction, as described in §2.1.1; the only difference being that a classifier predicts class membership, while regression predicts a real-valued number.

A regression tree is constructed by partitioning the space $\mathcal{X}$ by a sequence of binary splits into terminal nodes. In each terminal node t, the predicted response value $y(t)$ is constant. Starting with a learning sample $\mathcal{L}$, three elements are necessary to determine a tree predictor:

1. A method to select a split at every intermediate node,
2. A rule for determining when a node is terminal, and
3. A rule for assigning a value $y(t)$ to every terminal node t.

In order to assign a value to each terminal node, the resubstitution estimate for the misclassification cost,

$$R_r(d) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - d(\mathbf{X}_i)\right)^2,$$

is calculated. Then $y(t)$ is taken to minimise $R_r(d)$. The value of $y(t)$ that minimises $R_r(d)$ is the average of y_i for all observations $(\mathbf{X}_i, y_i)$ falling into t. Thus, the extremal value of $y(t)$ in question is given by

$$\overline{y}(t) = \frac{1}{N(t)} \sum_{\mathbf{X}_i \in t} y_i,$$

where the sum is taken over all y_i such that $\mathbf{X}_i \in t$, and where $N(t)$ is the total number of observations in t. The error of a regression tree T is given by

$$R_r(T) = \sum_{t \in \widetilde{T}} R_r(t),$$

where, $R_r(t)$, the error of node t is given by

$$R_r(t) = \frac{1}{N} \sum_{\mathbf{X}_i \in t} (y_i - \overline{y}(t))^2 .$$

Given any set of candidate splits $\mathcal{S}$ of a current terminal node t in $\widetilde{T}$, the best split s^* of t is a split in $\mathcal{S}$ which decreases $R_r(T)$ most. For any split s of t into descendant subsets t_L and t_R, let $\Delta R_r(s,t) = R_r(t) - R_r(t_L) - R_r(t_R)$. The best split, such that

$$\Delta R_r(s^*, t) = \max_{s \in \mathcal{S}} \{\Delta R_r(s,t)\},$$

is taken.

Minimal error-complexity pruning in regression trees is achieved in exactly the same way as minimal cost-complexity pruning in classification trees. The result of minimal error-complexity pruning is a decreasing sequence of trees $T_1 > T_2 > \ldots > \{t_1\}$, with $\{t_1\} < T_{max}$, and a corresponding increasing sequence of γ values $0 = \gamma_1 < \gamma_2 < \ldots$, such that, for $\gamma_k \leq \gamma < \gamma_{k+1}$, T_k is the smallest subtree of T_{max} minimising $R_{r\gamma}(T)$, the error-complexity measure of tree T as given by the expression in (2.3).

2.2 Variable Selection

Variable selection methods are used mainly in exploratory situations, where many explanatory variables have been measured and a final model explaining the response variable has not been reached or established [1].

Suppose y_i is a variable of interest, depending in some (possibly complex) way on a set of potential explanatory variables or predictors $(x_{1i}, \ldots, x_{Pi})$. The problem of variable selection, or subset selection as it is often called, arises when modelling the relationship between y_i and a subset of $(x_{1i}, \ldots, x_{Pi})$, where there is uncertainty about which subset to use. Such a situation is of particular interest when P is large and $(x_{1i}, \ldots, x_{Pi})$ is thought to contain many redundant or irrelevant variables [17].

The variable selection problem is most familiar in the context of linear regression, where attention is restricted to linear models. Hair, *et al.* [20] describe linear regression analysis as a statistical technique that may be used to analyze the relationship between a single response variable and several explanatory variables. The objective of linear regression analysis is to use the explanatory variables whose values are known to predict the single response variable selected by the researcher. Each explanatory variable is weighted by the regression analysis procedure to ensure optimal prediction of the response variable from the set of explanatory variables. The weights denote the relative contribution of the explanatory variables to the overall prediction and facilitate interpretation as to the influence of each variable in making the prediction, although correlation among the explanatory variables complicates the interpretive process. The set of weighted explanatory variables forms the regression variate, a linear combination of the explanatory variables that best predicts the response variable.

2.2.1 The Regression Model

Suppose P explanatory variables are used to predict the response variable $\mathbf{Y}$, and N observations of the form $(y_i, x_{1i}, x_{2i}, \ldots, x_{Pi})$ are available, where x_{ji} is the value of the j^{th} explanatory variable at the i^{th} observation, and y_i is the value of the response variable at the i^{th} observation. Linear regression models assume a relationship between $\mathbf{Y}$ and the P explanatory variables of the form

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_P x_{Pi} + \epsilon_i, \tag{2.4}$$

where ϵ_i is a stochastic error term, with mean 0, representing noise in the data. The errors ϵ_i are assumed to be independent and identically normally distributed with a constant variance σ^2; that is for all $i = 1, \ldots, N$

$$\epsilon_i \sim N(0, \sigma^2).$$

Suppose β_j $(j = 0, 1, \ldots, P)$ is estimated by $\hat{\beta}_j$, then the prediction for y_i is given by

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \ldots + \hat{\beta}_P x_{Pi}.$$

The prediction error of the regression model is defined by $e_i = y_i - \hat{y}_i$, for all $i = 1, \ldots, N$ [42].

2.2.2 Criteria for Variable Selection

Any variable selection procedure requires a criterion for deciding how many and which variables to select for the prediction of a response variable. The least squares method of estimation minimises the residual sum of squares, also called the *sum of squared errors*. Hair, *et al.* [20] define the sum of squared errors as the sum of squared prediction errors (residuals) across all observations, denoted by E. This quantity is used to denote the variance in the response variable not yet accounted for by the regression model. Hence

$$E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{N} e_i^2. \tag{2.5}$$

Small values of E indicate that the least squares profile fits the data well. Therefore, an implicit criterion for variable selection is the value of E. A related criterion to assess prediction accuracy is the *sum of squared regression*, which is the sum of squared differences between the mean and predicted values of the response variable for all observations. This quantity is denoted by R_g and represents the amount of improvement in explanation of the response variable attributable to the explanatory variables, as more explanatory variables are added to the regression equation. That is,

$$R_g = \sum_{i=1}^{N} (\hat{y}_i - \overline{y})^2. \tag{2.6}$$

The *total sum of squares*, denoted by T_S, is the total amount of variation that exists and needs to be explained by the explanatory variables. The so–called baseline is calculated by summing the squared differences between the mean and actual values for the response variables across all observations, that is

$$T_S = \sum_{i=1}^{N} (y_i - \overline{y})^2 = \sum_{i=1}^{N} (\hat{y}_i - \overline{y})^2 + \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = R_g + E.$$

In deciding between alternative subsets of variables, the subset producing the smaller value of E would typically be selected. Prediction accuracy may also be expressed by the coefficient of determination (R^2), given by

$$R^2 = \frac{R_g}{T}.$$

This quantity is a measure of the proportion of the variance of the response variable about its mean that is explained by the explanatory variables. The higher the value of R^2, the greater the explanatory power of the regression equation, and therefore the better the prediction of the response variable [20]. Note that

$$E = \sum_{i=1}^{N} (\hat{y}_i - \overline{y})^2 (1 - R^2).$$

Therefore, minimising E is equivalent to maximising the *coefficient of determination* (R^2). The value of R^2 will never decrease by including additional variables, therefore all the explanatory variables will be selected if the criterion of maximising R^2 is used. The *adjusted coefficient* of determination, denoted by $\overline{R}^2$, will reduce this bias and is related to R^2 by the relationship

$$\overline{R}^2 = R^2 - \frac{P(1-R^2)}{N-P-1},$$

where P is the number of explanatory variables and N is the sample size. The idea is that those explanatory variables that maximise $\overline{R}^2$ are chosen [1].

Another method of variable selection is to minimise the *residual mean square*, defined as

$$M^2 = \frac{E}{N-P-1}.$$

This quantity is related to the value of $\overline{R}^2$ by means of the relationship

$$\overline{R}^2 = \frac{1-M^2}{S_y^2},$$

where

$$S_y^2 = \sum_{i=1}^{N} \frac{(\hat{y}_i - \overline{y})^2}{N-1}.$$

The so–called *general F-test* is the basis for several selection procedures. Suppose the explanatory variables $(x_{1i}, x_{2i}, \ldots, x_{Pi})$ are used in the regression equation. Suppose also that measurements on Q additional variables $(x_{P+1,i}, x_{P+2,i}, \ldots, x_{P+Q,i})$, are available. Before deciding whether any of the additional variables should be included, the hypothesis that, as a set, the Q variables do not improve the prediction of the response variable, is tested. If the regression equation in the population has the form of equation (2.4), the hypothesis

$$H_0 : \beta_{P+1} = \beta_{P+2} = \ldots = \beta_{P+Q} = 0$$

is tested. To perform the test, an equation that includes all $P+Q$ variables is first obtained, and the residual sum of squares (E_{P+Q}) is computed. Similarly, an equation that includes only the first P variables and the corresponding residual sum of squares (E_P). Then the test statistic is computed as

$$F = \frac{(E_P - E_{P+Q})/Q}{E_{P+Q}/(N-P-Q-1)}.$$

The numerator measures the improvement in the equation by using the additional Q variables. The hypothesis is rejected if the computed value of F exceeds the tabled value $F(1-\alpha)$ at a level of significance α, with Q and $N-P-Q-1$ degrees of freedom [1].

2.2.3 General Approaches to Variable Selection

There are several approaches to assist in finding the best regression model.

2.2.3.1 Forward Selection Method

In the forward selection method the best single variable is selected by choosing the variable with the highest absolute correlation with y_i. The partial correlation coefficients are examined to find an additional explanatory variable that explains the largest statistically significant potion of the error remaining from the first regression equation. The variable selected will maximise the F statistic for testing that the partial correlation coefficient is zero. The forward selection method proceeds in this manner, each time adding one variable to the variables previously selected, until a specified termination rule is satisfied. The most commonly used termination rule is based on the F-test of the hypothesis that the partial correlation of the variable entered is equal to zero. The termination rule terminates the process of entering variables when the computed value of F is less than a specified value. This cutoff value is called the *minimum F-to-enter value* [1, 20].

2.2.3.2 Backward Elimination Method

The backward elimination method begins with all the variables in the equation and proceeds by eliminating the least useful variables one at a time. For each variable, the F statistic testing that the variable's coefficient is zero, is computed. The F statistic here is called the *computed F-to-remove value*. The variable with the smallest computed F-to-remove value is a candidate for removal. The maximum F-to-remove value is specified. The termination rule terminates the process of removing variables when the minimum F-to-remove value is greater than some threshold [1].

2.2.3.3 Stepwise Procedure

The stepwise variable selection procedure is a combination of the forward selection and backward elimination methods. At step 0 only $\overline{y}$ is included. At step 1 the variable with highest computed F-to-enter value is selected. At step 2 a second variable with highest computed F-to-enter value is entered, if the highest computed F-to-enter value is greater than the minimum F-to-enter value. After the second variable is entered, the F-to-remove value is computed for both variables. If either of them is lower than the maximum F-to-remove value, the variable is removed. If not, a third variable is included if its computed F-to-enter value is large enough. In successive steps this process in repeated. For a given equation, variables with small enough computed F-to-remove values are removed, and the variables with large enough computed F-to-enter values are included. The process terminates when no variables may be deleted or added [1].

2.3 Logistic Regression

The goal in logistic regression is to find the best fitting, and most parsimonious model, to describe the relationship between a response or outcome variable, and a set of explanatory or predictor variables [24]. Logistic regression is a special form of regression, one in which the response variable is a nonmetric, dichotomous variable. Although some differences exist, the general manner of interpretation is similar to that of linear regression [20], as described in §2.2.

2.3.1 Difference between Logistic and Linear Regression

The key quantity in any regression analysis is the mean value of the response variable, given the values of the explanatory variables. This value is called the *conditional mean*, and is expressed as $E(y_i|\mathbf{X}_i)$, where y_i denotes the response variable and $\mathbf{X}_i$ denotes the value of the explanatory variables, as before. In the linear regression case, this mean is expressed as an equation linear in $\mathbf{X}_i$, that is

$$E(y_i|\mathbf{X}_i) = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_P x_{Pi}.$$

This expression implies that it is possible for $E(y_i|\mathbf{X}_i)$ to take on any value as $\mathbf{X}_i$ ranges between $-\infty$ and $+\infty$. When the response variable is a dichotomous variable, the mean must be greater than or equal to zero, and less than or equal to one, that is $0 \leq E(y_i|\mathbf{X}_i) \leq 1$. The logistic regression model is given by

$$\pi(\mathbf{X}_i) = \frac{e^{\beta_0 + \beta_1 x_{1i} + \ldots + \beta_P x_{Pi}}}{1 + e^{\beta_0 + \beta_1 x_{1i} + \ldots + \beta_P x_{Pi}}}, \tag{2.7}$$

where $\pi(\mathbf{X}_i) = E(y_i|\mathbf{X}_i)$. The logistic transformation is central in the study of logistic regression, and is defined in terms of $\pi(\mathbf{X}_i)$, as

$$\begin{aligned} g(\mathbf{X}_i) &= \ln\left[\frac{\pi(\mathbf{X}_i)}{1 - \pi(\mathbf{X}_i)}\right] \\ &= \beta_0 + \beta_1 x_{1i} + \ldots + \beta_P x_{Pi}. \end{aligned}$$

The significance of this transformation is that $g(\mathbf{X}_i)$ has many of the desirable properties of a linear regression model. The logistic transformation, $g(\mathbf{X}_i)$, is linear in its parameters, may be continuous, and may range from $-\infty$ to $+\infty$, depending on the range of and properties of $\mathbf{X}_i$ [24].

The conditional distribution of the response variable in linear regression is different to that in logistic regression. In linear regression models, it is assumed that an observation of the response variable may be expressed as $y_i = E(y_i|\mathbf{X}_i) + \epsilon_i$, where ϵ_i indicates the observation's deviation from the conditional mean. It is assumed that ϵ_i follows a normal distribution with mean equal to zero and some variance that is constant across all values of the explanatory variables. It follows that the conditional distribution of the response variable, given $\mathbf{X}_i$, will be normal with mean $E(y_i|\mathbf{X}_i)$, and the variance is constant. This is not the case with a dichotomous response variable. In this situation, the value of the response variable, given $\mathbf{X}_i$, may be expressed as $y_i = \pi(\mathbf{X}_i) + \epsilon_i$, where ϵ_i may assume one of two possible values. If $y_i = 1$, $\epsilon_i = 1 - \pi(\mathbf{X}_i)$, with probability $\pi(\mathbf{X}_i)$. If $y_i = 0$, $\epsilon_i = -\pi(\mathbf{X}_i)$, with probability $1 - \pi(\mathbf{X}_i)$. Hence, ϵ_i has a distribution with mean equal to zero and variance equal to $\pi(\mathbf{X}_i)[1 - \pi(\mathbf{X}_i)]$ [24].

2.3.2 Fitting the Logistic Regression Model

Suppose a sample of N independent observations of the pair $(\mathbf{X}_i, y_i), i = 1, 2, \ldots, N$ exists, where y_i denotes the value of the dichotomous response variable, and $\mathbf{X}_i$ is the value of the explanatory variables in the i^{th} observation. To fit the logistic regression model in (2.7) to a data set, the values of the unknown parameters, $(\beta_1, \ldots, \beta_P)$, must be estimated. The maximum likelihood method is used to estimate the logistic regression model. In general, this method yields values for the unknown parameters which maximises the probability of obtaining the observed set of data.

If y_i is coded as zero or one, then the conditional probability that y_i is equal to 1 given $\mathbf{X}_i$, is provided by the expression for $\pi(\mathbf{X}_i)$, given in expression (2.7). This is denoted as $P[(y_i = 1)|\mathbf{X}_i]$. The conditional probability that y_i is equal to zero given $\mathbf{X}_i$, $P[(y_i = 0)|\mathbf{X}_i]$, is given by $1 - \pi(\mathbf{X}_i)$. For those pairs $(\mathbf{X}_i, y_i)$, for which $y_i = 1$, the contribution to the likelihood function is $\pi(\mathbf{X}_i)$, and for those pairs for which $y_i = 0$, the contribution to the likelihood function is $1 - \pi(\mathbf{X}_i)$. The contribution of the pair $(\mathbf{X}_i, y_i)$ to the likelihood function may be expressed as

$$\zeta(\mathbf{X}_i, y_i) = \pi(\mathbf{X}_i)^{y_i}[1 - \pi(\mathbf{X}_i)]^{1-y_i}. \tag{2.8}$$

The observations are assumed to be independent, hence, the likelihood function is obtained as the product of terms given in expression (2.8), that is

$$l(\beta_1, \ldots, \beta_P) = \prod_{i=1}^{N} \zeta(\mathbf{X}_i, y_i). \tag{2.9}$$

The principle of maximum likelihood states that the values which maximise the expression in (2.9) should be used as estimates of $(\beta_1, \ldots, \beta_P)$. It is mathematically easier to work with the logarithm of the quantity $l(\beta_1, \ldots, \beta_P)$ in (2.9), defined as

$$\begin{aligned} L(\beta_1, \ldots, \beta_P) &= \ln[l(\beta_1, \ldots, \beta_P)] \\ &= \sum_{i=1}^{N} \{y_i \ln[\pi(\mathbf{X}_i)] + (1 - y_i) \ln[1 - \pi(\mathbf{X}_i)]\}. \end{aligned} \tag{2.10}$$

To find values $(\beta_1, \ldots, \beta_P)$ that maximise $L(\beta_1, \ldots, \beta_P)$, the function is differentiated with respect to $(\beta_1, \ldots, \beta_P)$, and the resulting expressions set to zero. The resultant equations are

$$\sum_{i=1}^{N} [y_i - \pi(\mathbf{X}_i)] = 0 \tag{2.11}$$

and

$$\sum_{i=1}^{N} \mathbf{X}_i [y_i - \pi(\mathbf{X}_i)] = 0. \tag{2.12}$$

Equations (2.11) and (2.12) are nonlinear in $(\beta_1, \ldots, \beta_P)$, and are therefore solved numerically using iterative methods [24].

2.3.3 Testing for the Significance of the Coefficients

After estimating the coefficients of the explanatory variables in a response variable prediction model, an assessment of the significance of the variables in the model is performed. One approach to testing for the significance of the coefficient of an explanatory variable in any model, is to compare the prediction accuracy of a model that includes the variable in question, and that of a model that does not include that variable, with observed values of the response variable.

In linear regression, the assessment of the significance of a coefficient is performed by constructing an analysis of variance table. In this table the total sum of squares, denoted by T, is partitioned into the residual sum of squares, denoted by E, and the sum of squared regression, denoted by R_g. The observed value is denoted by y_i, and the predicted value for the i^{th} set of explanatory variable observation values under the model, is denoted by $\hat{y}_i$. This comparison is evaluated in (2.5). Under the model not containing explanatory variables, the only parameter is β_0, and $\hat{\beta}_0 = \overline{y}$, the mean of the response variable. In this case $\hat{y}_i = \overline{y}$, and E is equal to the total variance. When explanatory variables are included in the model, any decrease in E will be due to the fact that the slope coefficient for the explanatory variables is not zero. A change in the value of E is due to the regression source of variability, denoted R_g in (2.6). A large value for R_g suggests that the explanatory variable is important, whereas a small value suggests that the explanatory variable is not helpful in predicting the response.

The guiding principle in logistic regression is the same as in linear regression: The observed values of the response variable are compared to the predicted values obtained from models with and without the variable in question. In logistic regression, comparison of observed values of the response variable to predicted values, is based on the log likelihood function, defined in (2.10). To understand this comparison better, note that the observed value of the response variable is also a predicted value resulting from a *saturated model*. A saturated model is one that contains as many parameters as there are observations. The comparison between observed and predicted values of the regression model, using the log likelihood function, is based on the expression

$$D(\mathbf{X}_i, y_i) = -2\ln\left[\frac{L_c}{L_s}\right], \tag{2.13}$$

where L_c is the log likelihood of the current model, and L_s is the log likelihood of the saturated model. The reason for using minus twice the log is of a mathematically technical nature, and is necessary to obtain a quantity whose distribution is known and may thus be used for hypothesis testing purposes [24]. Such a test is called the likelihood ratio test. Using (2.10), the expression in (2.13) becomes

$$D(\mathbf{X}_i, y_i) = -2\sum_{i=1}^{N}\left[y_i \ln\left(\frac{\hat{\pi}(\mathbf{X}_i)}{y_i}\right) + (1-y_i)\ln\left(\frac{1-\hat{\pi}(\mathbf{X}_i)}{1-y_i}\right)\right], \tag{2.14}$$

where $\hat{\pi}(\mathbf{X}_i)$ is the maximum likelihood estimate of $\pi(\mathbf{X}_i)$. The statistic, $D(\mathbf{X}_i, y_i)$, in expression (2.14) is called the *deviation*. The deviation in logistic regression plays the same role as the residual sum of squares plays in linear regression. To assess the significance of an explanatory variable, the values of $D(\mathbf{X}_i, y_i)$ for equations with and without the explanatory variable, are compared. The change in $D(\mathbf{X}_i, y_i)$ is obtained as

$$\begin{aligned} G(\mathbf{X}_i, y_i) &= D(\tilde{X}_i, y_i) - D(X_i, y_i) \\ &= -2\ln\left[\frac{\tilde{L}_i}{L_i}\right], \end{aligned} \tag{2.15}$$

where $\tilde{\mathbf{X}}_i$ is the model without explanatory variable x_{ji}, $\mathbf{X}_i$ is the model with explanatory variable x_{ji}, L_i is the log likelihood of the model with the explanatory variable x_{ji}, and $\tilde{L}_i$ is the log likelihood of the model without the explanatory variable x_{ji}. For the case of a single explanatory variable, it is known that when that variable is not in the model, the maximum likelihood estimate of β_0 is $\ln(n_1/n_0)$, where $n_1 = \sum_{i=1}^{N} y_i$, and $n_0 = \sum_{i=1}^{N}(1-y_i)$, and the predicted value is the constant, n_1/N. In this case the value of $G(\mathbf{X}_i, y_i)$ is given by

$$\begin{aligned} G(\mathbf{X}_i, y_i) &= -2\ln\left[\frac{\left(\frac{n_1}{N}\right)^{n_1}\left(\frac{n_0}{N}\right)^{n_0}}{\prod_{i=1}^{N}\hat{\pi}(\mathbf{X}_i)^{y_i}(1-\hat{\pi}(\mathbf{X}_i))^{(1-y_i)}}\right] \\ &= 2\sum_{i=1}^{N}[y_i\ln(\hat{\pi}(\mathbf{X}_i) + (1-y_i)\ln(1-\hat{\pi}(\mathbf{X}_i)] - [n_1\ln(n_1) + n_0\ln(n_0) - N\ln(N)]. \end{aligned}$$

Under the hypothesis that $\beta_1, \ldots, \beta_P$ is equal to zero, the statistic $G(\mathbf{X}_i, y_i)$ follows a chi-square distribution with one degree of freedom [24].

2.4 Artificial Neural Networks

Hair, *et al.* [20] state that neural networks are one of the tools most likely to be associated with data mining. Patterned after the workings of the neural system of the brain, a neural network attempts to learn, by means of repeated trials, how to organise itself so as to achieve optimal prediction. The neural network model is composed of *nodes*, also called *neurons*, which act as *inputs*, *outputs*, or *intermediate processors*. Each node is connected to a set of neighbouring nodes by means of a series of *weighted paths*, similar to the weights in a regression model. Based on a learning paradigm, the model takes the input data of the first observation, and makes an initial prediction based on the weights of the network. The prediction error is assessed, and then the model attempts to modify the weights so as to improve prediction, and then moves on to the next observation. This cycle repeats itself for each observation in what is termed the *training phase*, when the model is being calibrated. After calibration, the model may be used on a separate sample of observations to assess its external validity.

2.4.1 Basic Concepts of Neural Networks

There are three basic types of neural networks: multilayer perceptron networks, radial basis function networks, and Kohonen networks. The multilayer perceptron model is the most commonly used and is the type described in this section.

The most basic element in a neural network is a node, a self-contained processing unit that acts in parallel with other nodes in the neural network. Each connection from another node has an assigned weight. A weight w_{kj} is interpreted as the strength of the connection from the j^{th} node to the k^{th} node. The node processes the incoming data by creating a weighted sum value in which each input value is multiplied by its respective weight. Thus the net input into node k during observation i is given by

$$a_{ki} = \sum_j w_{kj} x_{ji} + \mu_k,$$

where x_{ji} denotes the output value of the i^{th} observation from node j, and μ_k is a threshold value for node k.

Each node takes its net input and applies an *activation function* to it. For example, the output of the j^{th} node is $b\left(a_{ki}\right)$, where $b(\cdot)$ is the activation function. The two common choices for activation functions, are the threshold function

$$b_1(a_{ki}) = \begin{cases} 1 & \text{if } a_{ki} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

or sigmoidal functions, such as

$$b_2(a_{ki}) = \frac{1}{1 + e^{-a_{ki}}} \tag{2.16}$$

or

$$b_3(a_{ki}) = \tanh(a_{ki}).$$

A layer of a multilayer network is composed of nodes that perform similar tasks. A *feedforward network* is one where nodes in one layer are connected only to nodes in the next layer, and not to nodes in a preceding layer or nodes in the same layer. The first layer of a multilayer network consists of the *input nodes*, denoted by x_{ji}. An input node represents a single variable or pattern. Metric variables require only one node for each variable. Nonmetric variables have to be encoded, which means that each value of the nonmetric variable should be represented by a binary variable. The last layer contains the *output nodes*, denoted by y_{qi}. The outputs represent membership of one of the q classes. All other nodes in the model are called *hidden nodes*, denoted h_{ki}, and together constitute the hidden layers. It is the hidden layers and activation function that allow neural networks to represent nonlinear relationships. A feedforward network may have a number of hidden layers with a variable number of hidden nodes per layer [40].

2.4.2 Neural Network Learning

The weights in neural networks are adjusted to solve the problem presented to the network. Learning or training is the term used to describe the process of finding the values of these weights. The two types of learning associated with neural networks are *supervised learning* and *unsupervised learning*. Supervised learning occurs when there is a known target value associated with each input in the training set. The output of the network is compared with the target value, and the difference is used to train the network. Unsupervised learning occurs when the training data lack target output values corresponding to input patterns. The network must learn to group input patterns based on some common feature. One of the most common algorithms used for training neural networks, in the context of supervised learning, is *backpropagation*.

The backpropagation algorithm is a method to find weights for a multilayered feedforward network. To accomplish learning, some form of an objective function or performance metric is required. The goal is to use the objective function to optimise the weights. The most common performance metric used in neural networks is the sum of squared errors, defined as

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{q=1}^{O}(y_{qi} - \hat{y}_{qi})^2, \tag{2.17}$$

where the subscript i indexes observations (with a total of N observations), where the subscript q indexes output nodes (with a total of O output nodes), where y is the observed response, and where $\hat{y}$ is the model response.

The process of passing information through the multilayer feedforward neural network starts with the input values being presented to the input layer. The input nodes perform no operation on this information, but simply pass it onto the hidden nodes. The input to the k^{th} hidden node is

$$h_{ki} = \sum_{j=1}^{M} w_{kj} x_{ji},$$

where M is the total number of input nodes. The k^{th} hidden node applies an activation function,

$$v_{ki} = b(h_{ki}) = \frac{1}{1 + e^{-h_{ki}}},$$

to its net inputs and outputs, assuming that the activation function $b(\cdot)$ is the sigmoidal function, defined in (2.16). Similarly, an output node q receives a net input of

$$f_{qi} = \sum_{k=1}^{H} W_{qk} v_{ki},$$

where H is the number of hidden nodes, and W_{qk} represents the weight from hidden node k to output q. The node then outputs the quantity

$$\hat{y}_{qi} = b(f_{qi}) = \frac{1}{1 + e^{-f_{qi}}}.$$

The goal is to find the set of weights w_{kj}, the weights connecting the input nodes to the hidden nodes, and W_{qk}, the weights connecting the hidden nodes to the output nodes that minimise the objective function, the sum of squared errors in (2.17). The objective function is a function of the unknown weights w_{kj} and W_{qk}. Therefore, the partial derivative of the objective function with respect to weights w_{kj} and W_{qk} represents the rates of change of the objective function with respect to a unit change of those weights. The method to find values for the weights is an iterative process, evaluating the partial derivatives of the objective function with respect to the weights, and adjusting the weights in a direction down the slope, continuing in this manner until the error function no longer decreases.

To update the weight W_{qk}, the quantity

$$\begin{aligned} \Delta W_{qk} &= -\eta \frac{\partial E}{\partial W_{qk}} \\ &= -\eta \frac{\partial E}{\partial \hat{y}_{qi}} \frac{\partial \hat{y}_{qi}}{\partial f_{qi}} \frac{\partial f_{qi}}{\partial W_{qk}} \\ &= -\eta [(-1)(y_{qi} - \hat{y}_{qi})] \hat{y}_{qi} (1 - \hat{y}_{qi}) v_{ki} \qquad (2.18) \end{aligned}$$

is computed, where η simply scales the step size. The weights are updated as

$$W'_{qk} = W_{qk} + \Delta W_{qk}.$$

To update the weight w_{kj}, the quantity

$$\begin{aligned}\Delta w_{kj} &= -\eta \frac{\partial E}{\partial w_{kj}} \\ &= -\eta \sum_{q=1}^{O} \frac{\partial E}{\partial \hat{y}_{qi}} \frac{\partial \hat{y}_{qi}}{\partial f_{qi}} \frac{\partial f_{qi}}{\partial v_{ki}} \frac{\partial v_{ki}}{\partial h_{ki}} \frac{\partial h_{ki}}{\partial w_{kj}} \\ &= -\eta \sum_{q=1}^{O} (y_{qi} - \hat{y}_{qi}) \hat{y}_{qi} (1 - \hat{y}_{qi}) W_{qk} v_{ki} (1 - v_{ki}) x_{ji}\end{aligned}$$

is computed, and substituted into the iterative relationship $w'_{kj} = w_{kj} + \Delta w_{kj}$. Note that there is a summation over the number of output nodes. This is because each hidden node is connected to all the output nodes.

The method described above may be summarised as follows in algorithmic fashion [40]:

Algorithm 1. *Backpropagation Algorithm*

1. *Initialise the weights to small random values.*
2. *Choose an observation i and propagate it forward. This yields values for v_{ki} and $\hat{y}_{qi}$, the outputs from the hidden layer and output layer.*
3. *Compute the output errors: $\delta_{qi} = (y_{qi} - \hat{y}_{qi}) b'(f_{qi})$.*
4. *Compute the hidden layer errors: $\psi_{ki} = \sum_{q=1}^{O} \delta_{qi} W_{qk} v_{ki} (1 - v_{ki})$.*
5. *Compute $\Delta W_{qk} = \eta \delta_{qi} v_{ki}$ and $\Delta w_{kj} = \eta \psi_{ki} x_{ji}$*
6. *Repeat steps 2–5 for each observation.*

2.4.3 Neural Networks vs Standard Statistical Techniques

Sarle [34] argues that artificial neural networks are nothing more than nonlinear regression and discriminant models that may be implemented with standard statistical software. Many artificial neural networks are similar or identical to popular statistical techniques[1], especially where the emphasis is on prediction of complicated phenomena, rather than on explanation. Sarle argues that artificial neural networks learn in much the same way that many statistical algorithms perform estimation, but usually much more slowly than statistical algorithms.

[1] Generalised linear regression models, polynomial regression models, nonparametric regression models and discriminant analysis, projection pursuit regression models, principle component analysis, and cluster analysis.

Multilayer perceptron models are general-purpose, flexible, nonlinear models that, given enough hidden nodes and enough data, are able to approximate virtually any function to any desired degree of accuracy. The complexity of the multilayer perceptron model may be varied by varying the number of hidden layers and the number of hidden nodes in each hidden layer. With a small number of hidden nodes, a multilayer perceptron model is a parametric model that provides a useful alternative to polynomial regression. With a moderate number of hidden nodes, the multilayer perceptron model may be considered a quasi-parametric model, similar to projection pursuit regression models. A multilayer perceptron model with one hidden layer is essentially the same as the projection pursuit regression model, except that a multilayer perceptron model uses a predetermined functional form for the activation function in the hidden layer, whereas projection pursuit models use a flexible nonlinear smoother. If the number of hidden nodes is allowed to increase with the sample size, a multilayer perceptron model becomes a nonparametric sieve that provides a useful alternative to methods, such as kernel regression and smoothing splines. Multilayer perceptron models are especially valuable, because the complexity of these models may be varied from a simple parametric model to a highly flexible, nonparametric model.

2.5 Bayesian Decision Making

Bayesian analysis is concerned with the basic problem of assessing some underlying state of nature that is in some way uncertain. On the basis of whatever evidence does exist, some action or actions are to be chosen from among various possible alternatives. Suppose that there exists a set of mutually exclusive and exhaustive events that are considered possible. It is known in advance that one, and only one, of these events will actually occur, but there is uncertainty about which one of these it will be. Bayesian analysis involves assigning a probability to each of these events on the basis of whatever evidence is currently available. If additional evidence is subsequently obtained, then the initial probabilities are revised on the basis of this new evidence by means of Bayes' Theorem. The initial probabilities are known as prior probabilities in that they are assigned before the acquisition of the additional evidence bearing on the problem. The probabilities which result from the revision process are known as posterior probabilities [31].

Bayesian classifiers are statistical classifiers, that are able to predict class membership probabilities, such as the probability that a given sample belongs to a particular class. Let $\mathbf{X}_i$ be an observation whose class label is unknown and let H_i be some hypothesis, such as, that the observation $\mathbf{X}_i$ belongs to a specified class. For classification problems, the probability that the hypothesis H_i holds, given the observation $\mathbf{X}_i$, namely $P[H_i|\mathbf{X}_i]$, must be determined. The conditional probability $P[H_i|\mathbf{X}_i]$ can be calculated by Bayes' theorem, namely

$$P[H_i|\mathbf{X}_i] = \frac{P[\mathbf{X}_i|H_i]P[H_i]}{P[\mathbf{X}_i]}.$$

2.5.1 Naive Bayesian Classification

In a naive Bayesian classification process, each observation is represented by a P-dimensional feature vector, $\mathbf{X}_i = (x_{1i}, x_{2i}, \ldots, x_{Pi})$, denoting measurements made on P attributes.

Suppose that there are J classes, $C_1, C_2, \ldots, C_J$. Given an unknown observation, $\mathbf{X}_i$, the naive Bayesian classifier predicts that $\mathbf{X}_i$ belongs to the class having the highest posterior probability, conditioned on $\mathbf{X}_i$. That is, the naive Bayesian classifier assigns an unknown observation $\mathbf{X}_i$ to the class C_i if and only if

$$P[C_i|\mathbf{X}_i] > P[C_j|\mathbf{X}_i] \text{ for } 1 \leq j \leq J,\ j \neq i. \tag{2.19}$$

Thus $P[C_i|\mathbf{X}_i]$ is maximised. The class C_i for which $P[C_i|\mathbf{X}_i]$ is maximised is called the maximum posteriori hypothesis. By Bayes' theorem it holds that

$$P[C_i|\mathbf{X}_i] = \frac{P[\mathbf{X}_i|C_i]P[C_i]}{P[\mathbf{X}_i]},$$

where $P[\mathbf{X}_i]$ is the prior probability of observation $\mathbf{X}_i$, $P[C_i]$ is the prior probability of the observations belonging to class C_i and $P[\mathbf{X}_i|C_i]$ the posterior probability of observation $\mathbf{X}_i$ conditioned on class C_i.

The value $P[\mathbf{X}_i]$ is independent of class. Therefore, only the product $P[\mathbf{X}_i|C_i]P[C_i]$ needs to be maximised. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P[C_1] = P[C_2] = \ldots = P[C_J]$, and therefore $P[\mathbf{X}_i|C_i]$ will be maximised. Otherwise, $P[\mathbf{X}_i|C_i]P[C_i]$ will be maximised. The class prior probabilities may be estimated by $P[C_i] = \frac{N_{C_i}}{N}$, where N_{C_i} is the number of observations of class C_i, and N is the total number of observations.

Given data sets with many attributes, it may be computationally expensive to compute $P[\mathbf{X}_i|C_i]$. In order to reduce computational expense in evaluating $P[\mathbf{X}_i|C_i]$, a naive assumption of class conditional independence is made. In this assumption one presumes that the values of the attributes are conditionally independent of one another, given the class label of the observation. Thus,

$$P[\mathbf{X}_i|C_i] = \prod_{j=1}^{P} P[x_{ji}|C_i].$$

The probabilities $P[x_{1i}|C_i], P[x_{2i}|C_i], \ldots, P[x_{Pi}|C_i]$ may be estimated from the training samples, where

1. $P[x_{ji}|C_i] = \frac{N_{C_{ji}}}{N_{C_i}}$, if x_{ji} is categorical, where $N_{C_{ji}}$ is the number of observations of class C_i having the value x_{ji}, and N_{C_i} is the number of observations belonging to C_i.

2. The attribute is typically assumed to have a Gaussian distribution if x_{ji} is continuous, so that
$$P[x_{ji}|C_i] = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_{ji}-\mu)^2}{2\sigma^2}},$$
where μ and σ are the mean and standard deviation, respectively, given the values for attribute x_{ji} for observations of class C_i.

2.5.2 Bayesian Belief Networks

In practice, however, dependencies may exist between variables. Bayesian belief networks specify joint conditional probability distributions in order to accommodate such dependencies. They allow class conditional dependencies to be defined between subsets of variables, by providing a graphical model of causal relationships, on which learning may be performed. These networks are also known as *belief networks*, *Bayesian networks*, or *probabilistic networks*.

A belief network is defined by two components. The first is a directed acyclic graph, in which each node represents a random variable and each arc represents a probabilistic dependence. If an arc is drawn from a node t to a node t_d, then t is called a *parent* or *immediate predecessor* of t_d, and t_d is called a *descendent* of t. Each variable is conditionally independent of its nondescendents in the graph, given its parents. The variables may be discrete or continuous. They may correspond to actual attributes given in the data or to hidden variables believed to form a relationship.

The second component defining a belief network consists of one conditional probability table for each variable. The conditional probability table for t_d specifies the conditional distribution $P[t_d|O(t_d)]$, where $O(t_d)$ is the set of parents of t_d.

The joint probability of any tuple $(x_{1i}, \ldots, x_{Pi})$ corresponding to the variables or attributes of $t_{d_1}, \ldots, t_{d_P}$ is computed by

$$P[x_{1i}, \ldots, x_{Pi}] = \prod_{j=1}^{P} P[x_{ji}|O(t_{d_j})],$$

where the values for $P[t_d|O(t_d)]$ correspond to the entries in the conditional probability table for t_{d_j}.

A node within the network may be selected as an output node, representing a class label attribute. There may be more than one output node. Inference algorithms for learning may also be applied on the network. The classification process, rather than returning a single class label, returns a probability distribution for the class label attribute, that is, predicting the probability for membership to each class [21].

2.5.3 Training Bayesian Belief Networks

During the learning or training phase of a belief network, a number of scenarios are possible. The network structure may be given in advance or inferred from the data, and the network variables may be observable or hidden in all or some of the observations.

If the network structure is known and the variables are observable, then training the network is straightforward. It consists of computing the conditional probability table entries, as is done when computing the probabilities involved in a naive Bayesian classification process.

When the network structure is given and some of the variables are hidden, then a method of gradient descent is typically used to train the belief network. The objective is to learn the values of the conditional probability table entries. Let $\mathcal{X}$ be a set of N observations, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$. Let $P_{w_{ijq}}$ be a conditional probability table entry for the variable $t_{d_j} = y_{qi}$ having the parents $O(t_{d_j}) = x_{ji}$. $P_{w_{ijq}}$ is viewed as a weight, analogous to the weights in hidden nodes of neural networks. The set of weights is collectively referred to as P_w. The weights are initialised to random probability values. At each iteration of the gradient descent method, the weights are updated and eventually converge to a local optimum solution.

The method searches for the $P_{w_{ijq}}$ values that best model the data, based on the assumption that each possible setting of P_w is equally likely. The goal is thus to maximise $P_{P_w}[\mathcal{X}] = \prod_{k=1}^{N} P_{P_w}[\mathbf{X}_k]$. This is achieved by following the gradient of $\ln P_{P_w}[\mathcal{X}]$, which makes the problem simpler. Given the network structure and initialised $P_{w_{ijq}}$, the algorithm proceeds as follows:

Algorithm 2. *Bayesian Belief Network Training Algorithm*

1. *The gradients are computed: For each i, j, q the derivatives*

$$\frac{\partial \ln P_{P_w}[\mathcal{X}]}{\partial P_{w_{ijq}}} = \sum_{k=1}^{N} \frac{P[t_{d_j} = y_{qi}, O(t_{d_j}) = x_{ji} | \boldsymbol{X}_k]}{P_{w_{ijq}}} \tag{2.20}$$

are computed. The probability on the right-hand side of (2.20) is to be calculated for each training sample $\boldsymbol{X}_k$ in $\mathcal{X}$. When the variables represented by t_{d_j} and $O(t_{d_j})$ are hidden for some $\boldsymbol{X}_k$, then the corresponding probability on the right-hand side of (2.20) may be computed from the observed variables of the sample using standard algorithms for Bayesian network inference.

2. *A small step is taken in the direction of the gradient, and the weights are updated by*

$$P_{w_{ijq}} \leftarrow P_{w_{ijq}} + (l)\frac{\partial \ln P_{P_w}[\mathcal{X}]}{\partial P_{w_{ijq}}}, \tag{2.21}$$

where l is the learning rate representing the step size.

3. *The weights are renormalised: Because the weights $P_{w_{ijq}}$ are probability values, they must be between 0 and 1, and $\sum_j P_{w_{ijq}}$ must be equal to 1 for all i, q. These criteria are achieved by renormalising the weights after they have been updated [21].*

4. *Steps 1-3 are repeated until the values $P_{w_{ijq}}$ maximise P_{P_w}.*

2.6 Cluster Analysis

Exploratory procedures are often quite helpful in understanding the complex nature of multivariate relationships. Searching available data for a structure of groupings is an important exploratory technique [28]. Groupings may provide an informal means for assessing dimensionality, identifying outliers, and suggesting interesting hypotheses concerning relationships between data points. Cluster analysis is distinct from classification methods in that the latter pertains to a known number of groups, where the operational objective is to assign new observations to one of these groups. Cluster analysis, however, is a more primitive technique in that no assumptions are made concerning the number of groupings or the group structure of the data. It is an analytical technique for developing meaningful subgroups of observations, in which the objective is to classify a sample of observations into a small number of mutually exclusive groupings, based on the similarities and differences among the observations [28]. Cluster analysis usually involves three steps: (i) the *measurement* of some form of similarity or association among the observations, so as to determine how many groupings justifiably exist in the sample, (ii) the *partitioning* of observations into clusters, and (iii) the *profiling* of the variables to determine their composition [20].

2.6.1 Hierarchical Clustering Methods

The concept of similarity is fundamental to cluster analysis. So–called inter-observation similarity is a measure of correspondence, or resemblance, between observations to be clustered. Inter-observation similarity may be measured in a variety of ways, but three methods dominate the applications of cluster analysis: *correlation measures*, *distance measures*, and *association measures* [20].

The Euclidean distance between two P-dimensional observations $\mathbf{X}_i = (x_{1i}, x_{2i}, \ldots, x_{Pi})$ and $\mathbf{X}_j = (x_{1j}, x_{2j}, \ldots, x_{Pj})$ is

$$\begin{aligned} d_1(\mathbf{X}_i, \mathbf{X}_j) &= \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \ldots + (x_{Pi} - x_{Pj})^2} \\ &= \sqrt{(\mathbf{X}_i - \mathbf{X}_j)'(\mathbf{X}_i - \mathbf{X}_j)}. \end{aligned}$$

Another distance measure often used is the so–called *Minkowski metric*, given by

$$d_2(\mathbf{X}_i, \mathbf{X}_j) = \left[\sum_{k=1}^{P} |x_{ki} - x_{kj}|^m\right]^{\frac{1}{m}}.$$

For $m = 1, d_2(\mathbf{X}_i, \mathbf{X}_j)$ measures the rectilinear distance between two points in P dimensions. For $m = 2, d_2(\mathbf{X}_i, \mathbf{X}_j)$ becomes the Euclidean distance.

Gower's [19] general similarity coefficient is one of the most popular measures of proximity for mixed data types, and normalises the meaningful variables to the range $[0.0, 1.0]$. Gower's general similarity coefficient $d_3(\mathbf{X}_i, \mathbf{X}_j)$ compares two P-dimensional observations $\mathbf{X}_i = (x_{1i}, x_{2i}, \ldots, x_{Pi})$ and $\mathbf{X}_j = (x_{1j}, x_{2j}, \ldots, x_{Pj})$, and is defined as

$$d_3(\mathbf{X}_i, \mathbf{X}_j) = \frac{\sum_{k=1}^{P} W_{ijk} d_{4_k}(\mathbf{X}_i, \mathbf{X}_j)}{\sum_{k=1}^{P} W_{ijk}}, \tag{2.22}$$

where $d_{4_k}(\mathbf{X}_i, \mathbf{X}_j)$ denotes the contribution provided by the k^{th} variable, and W_{ijk} is usually 1 or 0, depending upon whether or not the comparison is valid for the k^{th} variable [19]. For ordinal and continuous variables, Gower [19] defines the value of $d_{4_k}(\mathbf{X}_i, \mathbf{X}_j)$ as

$$d_{4_k}(\mathbf{X}_i, \mathbf{X}_j) = 1 - \frac{|x_{ik} - x_{jk}|}{r_k},$$

where r_k is the range of values for the k^{th} variable. For continuous variables, $d_4(\mathbf{X}_i, \mathbf{X}_j)$ ranges between 1, for identical values $x_{ik} = x_{jk}$, and 0, for two extreme values of x. The value of $d_{4_k}(\mathbf{X}_i, \mathbf{X}_j)$ for nominal variables is 1 if $x_{ik} = x_{jk}$, and 0 if $x_{ik} \neq x_{jk}$. For binary variables, $d_{4_k}(\mathbf{X}_i, \mathbf{X}_j) = 1$ if observations i and j both have attribute k present, or 0 otherwise [21, 19].

When observations cannot be represented by meaningful P-dimensional measurements, pairs of items are often compared on the basis of the presence or absence of certain characteristics. The presence or absence of a characteristic may be described mathematically by introducing a binary variable, which assumes the value 1 if the characteristic is present and the value 0 if the characteristic is absent. *Similarity coefficients* are defined by arranging the frequencies of matches and mismatches for items i and k in the form of a contingency table. In the contingency table shown in Table 2.1, a represents the frequency of $1-1$ matches, and b is the frequency of $1-0$ matches, and so forth. The most common similarity coefficient used, defined in terms of the frequencies in the contingency table, is

$$\frac{a+d}{p},$$

in which equal weights for $1-1$ and $0-0$ similarities are applied [28].

		Item k		
		1	0	Totals
Item i	1	a	b	$a+b$
	0	c	d	$c+d$
	Totals	$a+c$	$b+d$	$p=a+b+c+d$

Table 2.1: *Similarity contingency table.*

Methods of Clustering

Hierarchical clustering techniques proceed either by a series of successive mergers, termed *agglomerative hierarchical methods*, or by a series of successive divisions, termed *divisive hierarchical methods* [20]. Agglomerative hierarchical methods start with the individual observations — hence there are initially as many clusters as observations. The most similar observations are first grouped together, and these initial groups are merged according to their similarities. Eventually, as the similarity decreases, all subgroups are fused into a single cluster. Divisive hierarchical methods work in the opposite direction. An initial single group of observations is divided into two subgroups, such that the observations in one subgroup are far from the observations in the other in some sense. These subgroups are then further divided into dissimilar subgroups. The process continues until there are as many subgroups as observations [28]. In both agglomerative or divisive hierarchical clustering, one may specify the desired number of clusters as a termination condition. The agglomerative hierarchical clustering algorithm for grouping N observations, is described by the following steps:

Algorithm 3. *Agglomerative Hierarchical Clustering Algorithm*

1. *The algorithm starts with N clusters, each containing a single observation and an $N \times N$ symmetric matrix of distances or similarities $\boldsymbol{D} = \{d_{ik}\}$.*

2. *The distance matrix is searched for the nearest or most similar pair of clusters. The distance between the most similar clusters, U and V, is given by d_{uv}.*

3. *Clusters U and V are merged into a new cluster, labelled (UV). The entries in the distance matrix are updated by deleting the rows and columns corresponding to clusters U and V, and by adding a row and column giving the distances between cluster (UV) and the remaining clusters.*

4. *Steps 2 and 3 are repeated $N-1$ times, until all observations are in a single cluster or until the termination condition is met.*

Four popular agglomerative methods used to develop clusters are:

1. Single linkage,
2. Complete linkage,
3. Average linkage and
4. McQuitty's Similarity Analysis.

Single Linkage

The inputs to single linkage algorithms may be distances or similarities between pairs of observations. Groups are formed from the individual observations by merging the nearest neighbours, where the term nearest neighbour indicates smallest distance or largest similarity. Initially, the smallest distance in $\mathbf{D} = \{d_{ik}\}$ must be found, and the corresponding clusters merged, say U and V, to obtain the cluster (UV). In Step 3 of the general algorithm (Algorithm 3), the distances between (UV) and any other cluster W are computed by

$$d_{(UV)W} = \min\{d_{UW}, d_{VW}\}.$$

The quantities d_{UW} and d_{VW} are the distances between the nearest neighbours of clusters U and W and clusters V and W, respectively.

Complete Linkage

Complete linkage clustering proceeds in much the same manner as single linkage, with one important exception. At each stage, the distance, or similarity, between clusters is determined by distance, or similarity, between two elements, one from each cluster, that are most distant. Complete linkage ensures that all items in a cluster are within some maximum distance, or minimum similarity, of each other. The general agglomerative algorithm again starts by finding the minimum entry in $\mathbf{D} = \{d_{ik}\}$ and merging the corresponding clusters, such as U and V, to obtain cluster (UV). In Step 3 of the general algorithm (Algorithm 3), the distances between (UV), and any other cluster W, are computed by

$$d_{(UV)W} = \max\{d_{UW}, d_{VW}\}.$$

The quantities d_{UW} and d_{VW} are distances between the most distant members of clusters U and W, and clusters V and W, respectively.

Average Linkage

In the average linkage approach one treats the distances between two clusters as the average distance between all pairs of items, where one member of a pair belongs to each cluster. The input to the average linkage algorithm may be distances or similarities. Average linkage algorithms begin by searching the distance matrix $\mathbf{D} = \{d_{ik}\}$ to find the nearest, or most similar, clusters, for example, U and V. These clusters are merged to form the cluster (UV). In Step 3 of the general agglomerative algorithm (Algorithm 3), the distances between (UV), and any other cluster W, are determined by

$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)} N_W},$$

where d_{ik} is the distance between item i in the cluster (UV) and item k in the cluster W, and $N_{(UV)}$ and N_W are the number of items in clusters (UV) and W, respectively.

McQuitty's Similarity Analysis

The inputs to McQuitty's Similarity Analysis may be distances or similarities between pairs of clusters. Groups are formed from the individual observations by merging the nearest neighbours, where the term nearest neighbour is taken to mean neighbour with largest similarity. Initially, the largest similarity in $\mathbf{D} = \{d_{ik}\}$ must be found, and the corresponding clusters merged, say U and V, to get the cluster (UV). In Step 3 of the general algorithm (Algorithm 3), the similarities between (UV) and any other cluster W are computed by

$$d_{(UV)W} = (d_{UW} + d_{VW})/2.$$

The quantities d_{UW} and d_{VW} measure the similarity between clusters U and W, and clusters V and W, respectively.

2.6.2 Non–hierarchical Clustering Methods

Non–hierarchical clustering techniques are designed to group items, rather than variables, into a collection of K clusters. The number of clusters, K, may either be specified in advance, or determined as part of the clustering procedure [28]. Non–hierarchical methods start from either an initial partition of items into groups, or an initial set of seed points, which form the nuclei of clusters. One way to start the process is to select seed points randomly from among the items, or to partition the items randomly into initial groups.

K-means Method

The K-means algorithm assigns each item to the cluster having the nearest centroid. This process consists of three steps.

Algorithm 4. *K-means Algorithm*

1. *The items are partitioned into K initial clusters.*

2. *The algorithm proceeds through the list of items, assigning an item to the cluster whose centroid is nearest. The centroid for the cluster receiving the item, and the cluster losing the item, are recalculated.*

3. *Step 2 is repeated until no more reassignments take place.*

2.7 Outlier Analysis

Very often, there exist observations in a data set that do not comply with the general behaviour or model of the set of data. Such observations, which are grossly different from or inconsistent with the remaining set of data, are called *outliers*.

Many data mining algorithms attempt to minimise the influence of outliers or eliminate them all together. However, this could result in the loss of important hidden information. In other words, the outliers themselves may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity.

Outlier mining may be described in general terms as follows: Given a set of N observations, and k, the expected number of outliers, the objective is to find the top k observations that are considerably dissimilar, exceptional, or inconsistent with the remaining data. The outlier mining problem may be viewed as two subproblems: to define which data may be considered inconsistent in a given data set, and to find an efficient method to mine the outliers so defined [21].

2.7.1 Statistical-Based Outlier Detection

In the statistical approach to outlier detection one assumes a distribution or probability model for the given data set, and then identifies outliers with respect to the model, using a so–called *discordancy test*. A statistical discordancy test examines two hypotheses, a working hypothesis, and an alternative hypothesis. A working hypothesis, H_i, is a statement that the entire data set of N observations comes from an initial distribution model, F_o, that is,

$$H_i : \mathbf{X}_i \in F_o, \text{where } i = 1, 2, \ldots, N.$$

The hypothesis is retained if there is no statistically significant evidence supporting its rejection. A discordancy test verifies whether an observation $\mathbf{X}_i$ is significantly large, or small, in relation to the distribution F_o. Different test statistics have been proposed for use in discordancy tests, depending on the available knowledge about the data. Assuming that some statistic T_s has been chosen for discordancy testing, and that the value of the statistic for observation $\mathbf{X}_i$ is v_i, the distribution of T_s is constructed. The so–called significance probability $P_s[v_i] = P[T_s > v_i]$ is evaluated. If some $P_s[v_i]$ is sufficiently small, then $\mathbf{X}_i$ is discordant and the working hypothesis is rejected for that value of i. An alternative hypothesis, $\overline{H}_i$, which states that $\mathbf{X}_i$ comes from another distribution model is adopted. The result is dependent on which model F_o is chosen, since $\mathbf{X}_i$ may be an outlier under one model and a perfectly valid value under another.

A major drawback of the statistical approach to outlier detection, is that most tests are for single attributes. Moreover, the statistical approach requires knowledge about parameters of the data set, such as the data distribution. Statistical methods do not guarantee that all outliers will be found for the cases where no data-specific test was developed, or where the observed distribution cannot be modelled adequately by means of any standard distribution [21].

2.7.2 Distance–Based Outlier Detection

An observation $\mathbf{X}_i$ is a *distance–based outlier* with parameters k and l, denoted by $D_o(k, l)$, if at least a fraction $0 \leq k \leq 1$ of the observations in $\mathcal{X}$ lie at a distance greater than l from observation $\mathbf{X}_i$. In distance–based outlier detection the ideas behind discordancy testing for various standard distributions are generalised. For many discordancy tests, it may be shown that if an observation $\mathbf{X}_i$ is an outlier according to the given test, then $\mathbf{X}_i$ is also a $D_o(k, l)$ outlier for some suitably defined k and l. Distance–based outlier detection requires the user to set values for both k and l. Finding suitable values for these parameters may involve a tedious trial and error process [21, 3].

Several efficient algorithms for mining distance–based outliers have been developed, of which the *index–based algorithm* is one. The index–based algorithm uses multidimensional indexing structures, such as *R-trees* or *KD-trees*, to search for neighbours of each observation $\mathbf{X}_i$ within a radius l around that observation. Let N_{M_l} be the maximum number of observations within this radius of an outlier. Then, once $N_{M_l} + 1$ neighbours of an observation $\mathbf{X}_i$ are found, it is clear that $\mathbf{X}_i$ is not an outlier [21].

2.7.3 Deviation–Based Outlier Detection

In a *deviation–based outlier detection* approach, one does not use statistical tests or distance–based measures to identify exceptional observations. Instead, one identifies outliers by examining the main characteristics of observations in a group. Observations that deviate from this description are considered outliers. One of the techniques used to detect outliers, is the *sequencial exception technique*. This technique simulates the way in which humans are able to distinguish unusual observations from among a series of supposedly-like observations. It uses implicit redundancy of the data. Given a set $\mathcal{X}$ of N observations, a sequence of subsets, $(S_1, S_2, \ldots, S_{N_s})$, of these observations with $2 \leq N_s \leq N$ is built such that

$$S_{j-1} \subset S_j, \text{ where } S_j \subseteq \mathcal{X} \text{ for all } s \leq j \leq N.$$

Instead of assessing the dissimilarity of the current subset with respect to its complementary set, the algorithm selects a sequence of subsets from the set for analysis. For every subset, it determines the dissimilarity difference of the subset with respect to the preceding subset in the sequence. The dissimilarity difference may be computed by any function returning a low value if the observations are similar to one another, and a higher value if the opposite is true [21].

2.8 Association Rule Mining

Progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as basket data. A record in such a data set typically consists of the transaction date and the items bought during the transaction [2]. In association rule mining one searches for interesting relationships among items in a given data set [21].

2.8.1 Basic Concepts in Association Rule Mining

Let $I = \{i_1, i_2, \ldots .i_m\}$ be a universal set of items. A subset of items in I is referred to as an *item set*. An item set that contains k items is called a *k-item set*. Let D_I, the task relevant data, be a set of database transactions where each transaction T_I is a set of items such that $T_I \subseteq I$. Each transaction is associated with an identifier, called the *transaction ID*. Let A_I be any set of items. A transaction T_I is said to contain A_I if and only if $A_I \subseteq T_I$. The frequency of occurrence of an item set is the number of transactions that contain the item set. An item set satisfies *minimum support* if the frequency of occurrence of the item set is greater than or equal to the product of the minimum support threshold and the total number of transactions in D_I. An *association rule* is an implication of the form $A_I \Rightarrow B_I$, where $A_I \subset I$, $B_I \subset I$, and $A_I \bigcap B_I = \emptyset$. The rule $A_I \Rightarrow B_I$ holds in the transaction set D_I with *support* c, where c is the percentage of transactions in D_I that contain $A_I \bigcup B_I$. This is taken to be an estimate of the probability $P[A_I \bigcup B_I]$. The rule $A_I \Rightarrow B_I$ is said to have *confidence* c in the transaction set D_I if c is the percentage of transactions in D_I containing A_I that also contain B_I. This is taken to be an estimate of the conditional probability, $P[A_I|B_I]$. Rules that satisfy both a minimum support threshold, denoted S_I, and a minimum confidence threshold, denoted C_I, are called *strong*. The number of transactions required for an item set to satisfy minimum support is referred to as the *minimum support count*. If an item set satisfies minimum support, then it is called a *frequent item set*. The set of frequent k-item sets is denoted by L_{I_k} [21, 2].

Association rule mining is a two-step process:

1. *Finding all frequent item sets*: By definition, each of these item sets will occur at least as frequently as a pre-determined minimum support count.

2. *Generating strong association rules from the frequent item sets*: By definition, these rules must satisfy minimum support and minimum confidence.

2.8.2 The Apriori Algorithm

The Apriori algorithm is an algorithm for mining frequent item sets for boolean association rules. The algorithm uses prior knowledge of frequent item set properties. Apriori employs an iterative approach known as level-wise search, were k-item sets are used to explore $(k+1)$-item sets. First, the set of frequent 1-item sets is found. This set is denoted L_{I_1}, and is used to find L_{I_2}, the set of frequent 2-item sets, which is used to find L_{I_3}, and so on, until no more frequent k-item sets can be found.

To improve the efficiency of the level-wise generation of frequent item sets, an important property called the *Apriori property* is used to reduce the search space. In order to use the Apriori property, all non-empty subsets of a frequent item set must also be frequent. This property is based on the following observation. By definition, if an item set I does not satisfy the minimum support threshold, S_I, then I is not frequent, that is, $P[I] < S_I$. If an item A_I is added to the item set I, then the resulting item set cannot occur more frequently than I. Therefore, $I \bigcup A_I$ is not frequent either, that is, $P[I \bigcup A_I] < S_I$.

A two-step process, consisting of join and prune actions, is used in the Apriori algorithms to find L_{I_k}, using $L_{I_{k-1}}$:

Algorithm 5. *Apriori Algorithm*

1. *The join step. To find L_{I_k}, a set of candidate k-item sets is generated by joining $L_{I_{k-1}}$ with itself. This set of candidates is denoted K_{I_k}. Let l_{I_1} and l_{I_2} be item sets in $L_{I_{k-1}}$. The notation $l_{I_i}[j]$ refers to the j^{th} item in l_{I_i}. By convention, the Apriori algorithm assumes that items within an item set are sorted in lexicographic order. The join, denoted $L_{I_{k-1}} \bowtie L_{I_{k-1}}$, is performed, where members of $L_{I_{k-1}}$ are joinable if their first $(k-2)$ items are in common. That is, members l_{I_1} and l_{I_2} of $L_{I_{k-1}}$ are joined if $(l_{I_1}[1] = l_{I_2}[1]) \wedge (l_{I_1}[2] = l_{I_2}[2]) \wedge \ldots \wedge (l_{I_1}[k-2] = l_{I_2}[k-2]) \wedge (l_{I_1}[k-1] < l_{I_2}[k-1])$, where $\wedge$ reads "and". The condition $(l_{I_1}[k-1] < l_{I_2}[k-1])$ simply ensures that no duplicates are generated. The resulting item set formed by joining l_{I_1} and l_{I_2} is $l_{I_1}[1]l_{I_1}[2] \ldots l_{I_1}[k-1]l_{I_2}[k-1]$.*

2. *The prune step. K_{I_k} is a superset of L_{I_k}, that is, its members may or may not be frequent, but all of the frequent k-item sets are included in K_{I_k}. A scan of the*

database to determine the count of each candidate in K_{I_k} would result in the determination of L_{I_k}. K_{I_k}, however, may be very large. To reduce the size of K_{I_k}, the Apriori property is used as follows. Any $(k-1)$-item set that is not frequent cannot be a subset of a frequent k-item set. Hence, if any $(k-1)$-subset of a candidate k-item set is not in $L_{I_{k-1}}$, then the candidate cannot be frequent either and so can be removed from K_{I_k}.

2.8.3 Generating Association Rules from Frequent Item Sets

Once the frequent item sets from transactions in a database D_I have been found, strong association rules may be generated from them. This may be achieved by using the following equation for confidence, where the conditional probability is expressed in terms of the item set support count

$$N_C(A_I \Rightarrow B_I) = P[A_I|B_I] = \frac{N_S(A_I \cup B_I)}{N_S(A_I)},$$

with N_C the rule of confidence, where N_S is the support count, where $N_S(A_I \bigcup B_I)$ is the number of transactions containing the item sets $A_I \bigcup B_I$, and where $N_S(A_I)$ is the number of transactions containing the item set A_I. For each frequent item set l_I, all non-empty subsets of l_I are generated. For each non-empty subset s_I of l_I, the rule $s_I \Rightarrow (l_I - s_I)$ is given, if $\frac{N_S(l_I)}{N_S(s_I)} \geq C_I$, where C_I is the minimum confidence threshold [21].

2.9 Chapter Summary

In this chapter, a number of data mining methodologies were reviewed, starting with a discussion on decision trees (focussing on classification and regression trees), followed by an examination of the regression model and its role in performing variable selection. This was succeeded by a summary of logistic regression and the differences between linear and logistic regression. Background on artificial neural networks and the learning of these networks using the backpropagation algorithm was also provided, followed by a review of Bayesian decision making (focussing on naive Bayesian classification and Bayesian believe networks). The remaining sections focussed on cluster analysis, outlier analysis and association rule mining. The data mining methodologies reviewed in this chapter are all applied to real call data records later in this thesis.

Chapter 3

The Cellular Telecommunications Industry

Just 20 years after the launch of the world's first commercial cellular services, there were more cellular telephone than fixed-line telephone users globally, and nearly as many people had a cellular telephone than had a television. Cellular communications experience faster growth rates in low-income countries. Low- and middle-income countries therefore account for a rising share of the world cellular market.

Africa has been the fastest growing cellular market in the world during the period 2000 – 2005. The first cellular telephone call in Africa was made in Zaire in 1987. In 2005 there were more than 52 million cellular telephone users in the continent, compared to about 25 million fixed lines — in 19 African countries, cellular telephones account for at least three quarters of all telephones. At the end of 2003, there were 6.1 cellular telephone subscribers for every 100 inhabitants in Africa, compared with 3 fixed line subscribers per 100 inhabitants [18].

In June 2004 South Africa had 18.7 million cellular subscribers, with a projected 19 million in 2006 [37]. The South African market is dominated by Vodacom and MTN, with a third license awarded in June 2001 to Cell C. The South African cellular telephone market was worth R 23 billion in 2005 and was estimated to grow to approximately R 54 billion by 2007 [37]. More than 5 500 Vodacom base stations were in place in 2005 to provide to 60% of the geographical area of the country. Together the three networks covered more than 71% of the population geographically in April 2005 [37].

3.1 Cellular Network Architecture

The *Global System for Mobile Communications* (GSM) network consists of three major parts (a graphical representation may be found in Figure 3.1): The *mobile station* is carried by the subscriber, the *base station subsystem* controls the radio link with the mobile station and the *network subsystem*, whose main part is the *Mobile services Switching Centre* (MSC), performs the switching of calls between mobile users, and between mobile and fixed line users. The mobile station, also referred to as *mobile* or *cellular telephone* or *handset*, consists of the physical equipment, such as the radio transceiver, display and digital signal processors, and a smart card called the *Subscriber Identity Module* (SIM). The mobile station is uniquely identified by the *International Mobile Equipment Identity* (IMEI). The SIM card provides personal mobility, so that the user may have access to all subscribed services irrespective of the location of the telephone and independent of the use of a specific telephone, and contains the *International Mobile Subscriber Identity* (IMSI) used to identify the subscriber to the system. The SIM card may be protected against unauthorised use by a password or personal identity number.

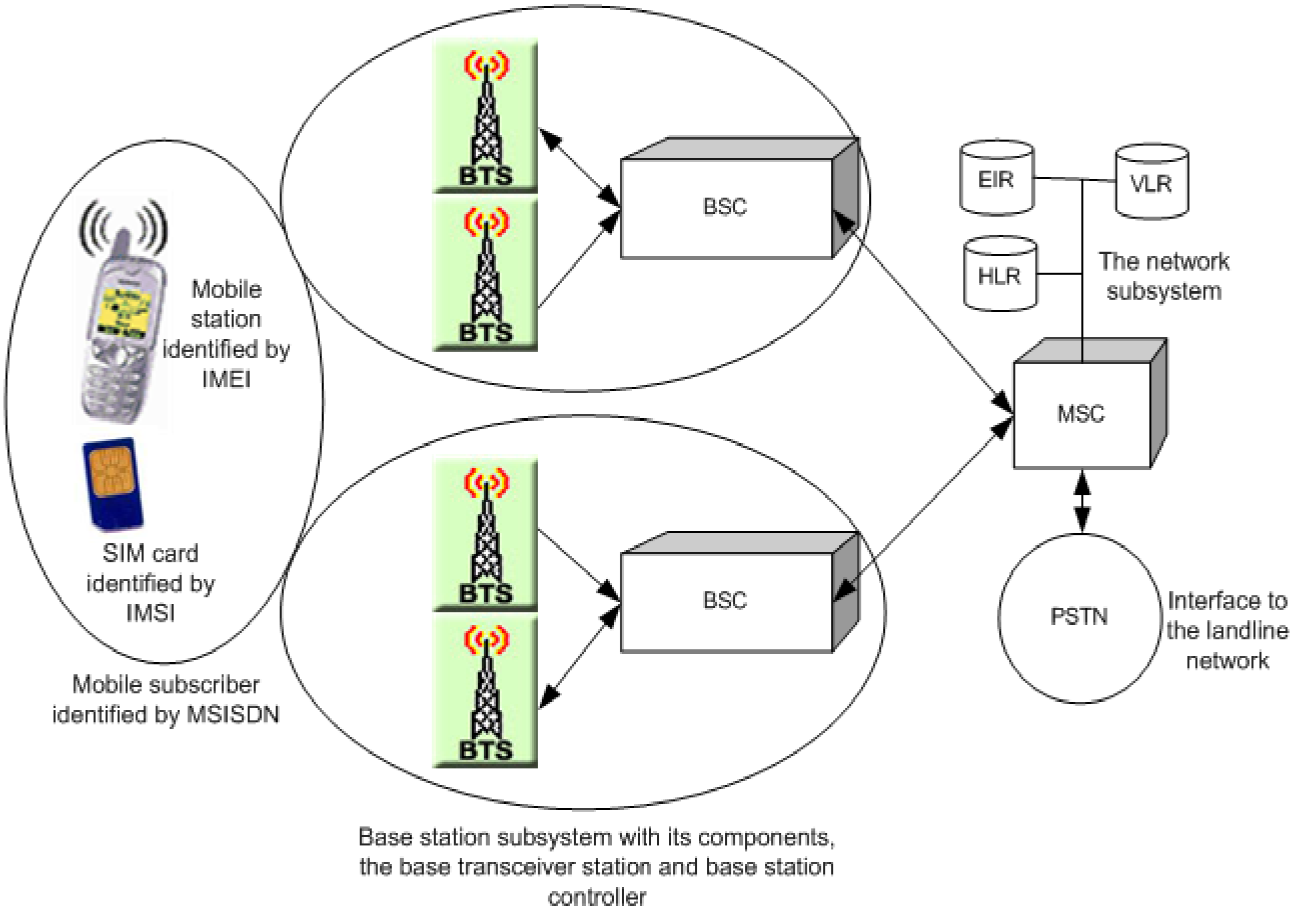
BTS
BTS
BSC
EIR
VLR
HLR
The network subsystem
Mobile station identified by IMEI
SIM card identified by IMSI
Mobile subscriber identified by MSISDN
MSC
BTS
BTS
BSC
PSTN
Interface to the landline network
Base station subsystem with its components, the base transceiver station and base station controller

Figure 3.1: *Graphical representation of the interaction between different components in a cellular telecommunications network.*

The directory number dialled to reach a mobile subscriber is called the *Mobile Subscriber Integrated Services Digital Network* (MSISDN). This number includes a country code and a *National Destination Code* (NDC), which identifies the subscriber's operator or service provider. When a mobile subscriber makes a call, the nearest cell site transceiver, depending on the best signal strength received, makes a radio connection with the cellular telephone. This connection is made possible by the base station for the cell site. The base station subsystem is composed of two parts, the *Base Transceiver Station* (BTS) and the *Base Station Controller* (BSC). The BTS houses the radio transceivers that define a cell and handles the radio-link with the mobile station. There are typically a large number of BTSs deployed in a large urban area. The call is then routed through the base station's transceiver to the *Mobile Switching Centre* (MSC). The mobile switch queries several databases before permitting a call to ensure that the caller is allowed on the network. The call is processed and routed to the *Public Switched Telephone Network* (PSTN) if the other party in the call is not subscribed to the same network operator as the caller. Calls between subscribers of the same network operator are routed via the cellular network. The *Home Location Register* (HLR) and *Visitor Location Register* (VLR), together with the MSC, provide the call-routing capabilities of the network. The HLR contains all the administrative information of each subscriber registered in the corresponding network, along with the current location of the mobile station. The location of the mobile station is typically in the form of the signalling address of the VLR associated with the mobile station. The VLR contains selected administrative information from the HLR, necessary for call control and provision of the subscribed services, for each mobile telephone currently located in the geographical area controlled by the VLR. The *Equipment Identity Register* (EIR) is a database that contains a list of all valid mobile equipment on the network, where each mobile station is identified by its IMEI. An IMEI is marked as invalid if it has been reported stolen or is not approved to be used on the GSM network [36].

3.2 Cellular Network Operations

Computer systems are used to administer subscribers, mediate call data records and bill each billable call. The *subscriber administration and provisioning system* is used by the network operator and service providers to administer each subscriber. Network operators offer a wide range of products catering for different groups of subscribers, based on their network usage patterns. For example, a subscriber making most of its calls during off-peak periods will benefit from one of the products in the leisure group, having a low monthly subscription fee, with a bundle of free call units during off-peak periods, but expensive peak call rates. On the other hand, a subscriber making a large number of calls during peak periods will benefit more from a product in the high business category, having a more expensive monthly subscription, but with low peak call rates. Prepaid subscribers do not pay a fixed monthly subscription fee and are not billed at the end of each billing period, but must have a positive account balance before allowed to make a call. Postpaid subscribers, on the other hand, pay a monthly subscription fee, depending on the product subscribed to and additional services provisioned for, and are billed at the end of each billing period for the use of these services. All subscribers are provisioned by default with the basic telephony service allowing it to make and receive voice calls. Additional services may be provisioned, some at no additional cost, as part of the subscribed product, or on request. Most products include provisioning of services allowing the subscriber to send and receive text messages (SMSs). The data service, for example, allowing the subscriber to send and receive data over the cellular network, may be provisioned at no additional cost on most products, when requested.

The subscriber administration and provisioning system maintains a set of attributes indicating the subscriber's suspension status on the network. The subscriber may be suspended from creating and/or partaking in any network traffic, making international calls or making and receiving calls while in a foreign country. When a subscriber's mobile telephone is reported stolen the subscriber administration and provisioning system may be used to blacklist the subscriber's SIM card and mobile telephone, barring a fraudster from making and receiving calls. The subscribed product, provisioned services, payment method and other subscriber-related attributes are used to create a profile for each subscriber. The HLR, mediation system and billing system are updated with each subscriber's latest profile.

When an outgoing call is made, the VLR tests whether the subscriber is allowed to make that call. For example, if the subscriber is suspended from international dailing, an outgoing call to an international destination will not be allowed by the VLR. The HLR also tracks individual devices via the EIR and is used to track mobile telephones that have been reported stolen, reporting and preventing their use on the cellular network. The MSC generates a call data record for every call originating or terminating in the cellular network. This call data record contains basic information describing the call, including the unique identifier of the subscriber on the cellular network, the duration of the call, services used while making the call, which base station facilitated the call, and other characteristics describing the call. The mediation system aggregates the call data records received from the MSC, matches them with a service and price-per-unit, and finally generates a standardised call data record ready to be billed by the billing system.

3.3 Cellular Telecommunications Fraud

There are many different definitions of telecommunications fraud [26]. However, there seems to be a general consensus that telecommunications fraud involves the theft of services or deliberate abuse of cellular telephone networks. Furthermore, it is accepted that in these circumstances the perpetrator's intention is to avoid completely or at least reduce the charges that would legitimately have been charged for the services used. This is only a small part of the fraud problem, since the majority of telecommunications fraud that is committed, is for own profit. On occasion, this avoidance of call charges will be achieved through the use of deception in order to fool billing and customer care systems into invoicing the wrong party. Telecommunications fraud has been identified as the single largest cause of revenue loss for network operators, with figures averaging between 3 and 5 percent of an operator's annual revenue [26]. Fraud is the most significant threat to the communications business, eroding profit margins, consuming network capacity and jeopardising customer relationships.

Cellular telephone theft is the largest type of telecommunications fraud in South Africa, as is the case in other developing countries. A cellular telephone stolen from a legitimate subscriber may be used until the theft is reported and the SIM card is locked for further use on the cellular network. During mid 1995 the *Memorandum of Understanding* (MoU), the controlling and regulating document for GSM standards, defined a system whereby GSM networks may identify each cellular telephone's IMEI number. If the IMEI number is listed as stolen, the number may be flagged in the EIR, preventing access from the relevant unit to the network, regardless of which SIM card is inserted into the telephone.

South African network operators pay connection bonusses to service providers, who, in turn, use these bonusses to subsidise the price of cellular telephones as an added incentive to sign airtime contracts. Using fraudulent means, operators with false identity documents often buy discounted cellular telephones and sell the handset at a profit, holding on to the SIM card to be sold separately. The user of the SIM card has several days and sometimes weeks in which to make as many calls as desired before the accounts department realises that it is a bad account and disconnects the SIM card. Corrupt service providers also take advantage of the connection incentive scheme by connecting non-existing subscribers, thereby receiving an incentive bonus for the fraudulent connection.

Subscription fraud is the most common form of fraud worldwide. Perpetrators of this type of fraud apply for a service and once activated, immediately use it for national and international calls with no intention to pay for the calls made. Subscription fraud is almost always associated with *call selling*, also known as 'phreaking'. Criminal gangs increasingly use 'phreaking' as a means of setting up their own cut-price telephone service which they then proceed to sell to other criminals, to illegal immigrants and to refugees. One of Germany's more notorious 'phreaking' cases occurred when six fraudsters bought six cellular telephones for cash, thereby avoiding a credit check. They hired a hotel room and advertised calls to Vietnam at reduced rates. Once the customers arrived, they were asked to pay DM100 for 30-minute telephone calls. The cellular telephones were in use 24 hours a day and it took six weeks for them to be disconnected before they ran up bills for millions of Deutschmark [39].

Call forwarding manipulation consists of setting up a local number to forward calls to an international destination. A local number, number A, dials the call forwarding set number, number B, which forwards the call to the international number, number C. The call between number A and B is charged at local rates against subscriber A's account, while the forwarded call between number B and C are charged at international rates against subscriber B's account. Another scenario of call forwarding manipulation is where collusion exists between the caller and a call centre operator. The caller calls the call centre operator known to him free of charge, while the operator forwards the call to a third party.

The owner of a premium rate service receives revenue from users calling the number. *Premium rate service fraud* involves a high number of calls made to the premium rate service number from a subscriber's account without their knowledge, or from a number where there is no intention to pay for the calls.

3.4 Cellular Telecommunications Fraud Detection

A successful fraud management strategy consists of three elements: Prevention, Detection and Deterrence.

The purpose of a *fraud prevention* strategy is to erect obstacles to unauthorised access to the operator's network and systems. Technical solutions may include the issuing of password and PIN codes, performing pre-call validation using call operators or PINs, and the use of authentication and encryption procedures. In addition to technical solutions, network operators also implement business and procedural solutions. These include subscriber identity verification, requiring deposits before providing a service and implementing call restrictions for new subscribers.

When prevention mechanisms fail, *fraud detection* is used to detect compromises to the cellular network. Fraud has a common pattern of unusual or unexpected subscriber behaviour. Tools for monitoring subscriber behaviour include high-usage alerts, alerting fraud analysts when a subscriber's usage is above a pre-defined threshold value, and maintaining subscriber behaviour profiles, identifying significant changes in subscriber behaviour.

The purpose of *fraud deterrence* is to discourage criminals from committing fraud. Proactive fraud monitoring and the prosecution of perpetrators may contribute to deterring criminals from committing fraud.

3.5 Chapter Summary

The reader was provided with a basic understanding of the cellular telecommunications industry in this chapter, in which the architecture and operation of a cellular network, telecommunications fraud experienced by cellular network operators, and the required elements of a successful fraud management strategy were described. The focus in this thesis is on the use of mathematical and statistical techniques aiding in the process of *fraud detection* (one of the three elements of a successful fraud management strategy introduced in §3.4). The following chapter (Chapter 4) is dedicated to a concise literature survey of such fraud detection aids proposed for use in computerised fraud management systems in the cellular telecommunications industry.

Chapter 4

Literature on Fraud Detection

Summarising account activity is a major step in the design of a fraud detection system, because it is rarely practical to access or analyse *all* the call records for an account every time it is evaluated for fraud. A common approach is to reduce the call records for an account to several statistics that are recomputed during each period of fraud detection. Account summaries may be compared to threshold values during each period, and an account whose summary exceeds a threshold value may be queued to be analysed for fraud, by hand.

4.1 Fixed-time Fraud Detection

Taniguchi, *et al.* [38] present three methods to detect fraud, which may be combined to improve fraud detection performance. These methods are used to compute subscriber specific statistics for each period. First, a *feed-forward neural network* based on supervised learning is used to distinguish between the classes of fraudulent behaviour and legitimate behaviour, by means of a non-linear discriminative function. The problem with supervised learning is to adapt the neural network weights so that the input-output mapping corresponds to the input-output samples. The feed-forward mapping of a three-layer feed forward network is defined as

$$\sum_{k=0}^{H} W_{qk} b \left(\sum_{j=0}^{M} w_{kj} x_{ji} \right),$$

where b is a non-linear mapping, where W_{qk} is the weight of the link between the q^{th} output node and the k^{th} node in the hidden layer, and where w_{kj} is the weight of the link between the the j^{th} input node and the k^{th} node in the hidden layer. The feed-forward network used by Taniguchi, *et al.* [38] consists of five hidden units and a binary output. In order to constrain the complexity of the mapping, Taniguchi, *et al.* [38] use a weight decay regularisation. Secondly, density estimation is applied to model the past behaviour of each subscriber and to detect any abnormalities, based on past behaviour. The problem with probability density estimation is that a probability density function $P[x]$ of the mobile telephone subscriber's past behaviour has to be found or estimated, given a finite number of data points drawn from that density. Taniguchi, *et al.* [38] estimate the probability density function and compute the probability of current usage with the model. To model the probability density function, they use a Gaussian mixture model, which is a sum of weighted component densities of Gaussian form,

$$P[x] = \sum_{k=1}^{H} P[x|k]P[k],$$

where $P[x|k]$ is the k^{th} component density of Gaussian form and $P[k]$ is its mixing proportion. Finally, *Bayesian networks* are used by Taniguchi, *et al.* [38] to define probabilistic models under the assumptions of fraud and legitimate. Bayes' rule is used to invert these measures so as to calculate the probability for fraud, given the subscriber's behaviour. The data used in all three approaches are based on call data records, which are call records stored for billing purposes.

Fawcett, *et al.* [13] introduce the notion of activity monitoring, which typically involves monitoring the behaviour of a large population of entities for interesting events requiring further attention. The goal of activity monitoring is to issue alarms accurately and in a timely fashion. Cellular telephone fraud detection is a typical activity monitoring problem. The task is to scan a large set of accounts, examining the calling behaviour of each, and to issue an alarm when an account appears to have been defrauded. Let each $\mathbf{X} \in \mathcal{X}$ represent a customer account, where $\mathcal{X}$ is a set of data streams, and $\mathbf{X}$ is an ordered set of data items $(\mathbf{X}_1, \ldots, \mathbf{X}_N)$. Each $\mathbf{X}_i$ $(i = 1, \ldots, N)$ represents the detail of a cellular telephone call. Fawcett, *et al.* [13] define activity monitoring as the task of analysing the data streams in order to detect the occurrence of interesting behaviour, which they refer to as positive activity. Let τ be a point in time denoting the onset of positive activity. For the data items $\mathbf{X}$, τ designates the beginning of a contiguous subsequence $\mathbf{X}_\tau = (\mathbf{X}_{p_1}, \ldots, \mathbf{X}_{p_m})$, such that $time(\mathbf{X}_i) \geq \tau,\ \mathbf{X}_i \in \mathbf{X}_\tau$. The goal of activity monitoring is to give an indication that the sequence is exhibiting positive activity, called an alarm. An alarm ϖ represents the point in time when it is issued. Fawcett, *et al.* [13] assert that the goal of activity monitoring is not to identify *all* positive activity, nor to classify each $\mathbf{X}_i$ as positive or negative. Rather, the goal is to identify in a timely fashion that positive activity has begun. Alarming earlier may be more beneficial, but after a first alarm, a second alarm on the same sequence may add no value. Fawcett, *et al.* [13] define a scoring function $s(\tau, \varpi, \mathbf{X})$ which returns the benefit of an alarm ϖ on a given sub-sequence, with respect to a given τ. Positive activity may be defined in terms of this scoring function as the sub-sequence of $\mathbf{X}$ for which $s(\tau, time(\mathbf{X}_j), \mathbf{X}) > 0$. One possible scoring function is to count the number of fraudulent calls that would have been made, had the fraud not been detected, *i.e.*

$$s(\tau, \varpi, \mathbf{X}) = \left|\{\mathbf{X}_i \in \mathbf{X} \mid \mathit{call\_start}(\mathbf{X}_i) \geq \varpi\}\right|.$$

To evaluate activity monitoring performance, Fawcett, *et al.* [13] used so–called *Receiver Operating Characteristic* (ROC) analysis with minor modifications. ROC analysis depicts the tradeoff between true positive classifications and false positive ones, which is similar to the goal of activity monitoring.

Fawcett, *et al.* [14] define a method for choosing account-specific threshold values, rather than universal threshold values that apply to all customers. This procedure takes daily traffic summaries for a set of accounts that experienced at least 30 days of fraud-free traffic, before being hit by fraudulent activity, and applies a machine learning algorithm to each account separately in order to develop a set of rules that distinguishes between fraudulent and legitimate activity for the account. The superset of rules for all accounts is then pruned by keeping only those that apply to a number of accounts, with possibly different threshold values for different accounts. The final set of rules, therefore, covers most accounts, with the understanding that most of the final rules may be irrelevant for most accounts, but that all the final rules are relevant for at least some accounts. This fraud detection system separates calls by account, computes account summaries, and then compares account summaries to account-specific threshold values that were previously computed from training data. The account-specific threshold values may be updated periodically by re-fitting trees and sequentially selecting the account summaries to use.

However, account-specific threshold values have limitations. A procedure that requires a fixed period of uncontaminated traffic for training purposes does not apply to accounts that experience fraudulent activity before the training period is over. Moreover, rules that are good for one time period may not be relevant for future time periods, because account calling behaviour typically changes over time, and may even be seasonal.

Researchers at Bell Laboratories [11] agree with Fawcett, *et al.* [14] that fraud detection must be tailored to each account's own activity, but their goals for fraud detection are more ambitious. First, they argue that fraud detection should be event-driven, not time-driven, so that fraud may be detected as it is happening, not at fixed points in time that are unrelated to account activity. Secondly, fraud detection methods should have memory and use all past calls on an account, weighting recent calls more heavily but not ignoring earlier calls. Thirdly, fraud detection methods should be able to learn from the calling pattern on an account and adapt to legitimate changes in calling behaviour. Finally, fraud detection methods should be self-initialising, so that they may be applied to new accounts that do not have enough data for training.

4.2 Real-time Fraud Detection

The fraud detection algorithms of researchers at Bell Laboratories are based on tracking the behaviour of each customer by means of a multivariate probability distribution that may be used to predict the customer's next legitimate call. The estimate of this distribution is termed a *signature*, because it captures all that is known about the customer's current transaction behaviour. Signature design involves choosing a set of marginal and conditional distributions that is best for detecting fraud. In order to detect fraudulent activity quickly, it is important to be able to assign a meaningful initial signature or predictive distribution to new customers.

4.2.1 Account Signatures

Cahill, *et al.* [10] propose a fraud detection method based on the tracking of account behaviour, using so-called account signatures. An account signature might describe which call durations, times-between-calls, days-of-week and times-of-day, terminating numbers, and payment methods are likely for the account and which are unlikely for the account. That is, the call variables for each call are described by a multivariate probability distribution, and an account signature is an estimate of the multivariate probability distribution. Cahill, *et al.* [10] propose use of the *law of iterated probability* to reduce the complexity of the multivariate distribution. Designing a signature amounts to eliminating conditioning variables that do not matter from the product of probabilities, given by the law of iterated probabilities. They assume that all signature components are represented by histograms in which the labels of the bins are not fixed. The histogram bins are selected so that, on average, it is as easy as possible to distinguish between legitimate calls and fraudulent calls. For a signature variable, this may be accomplished by maximising the average weighted *Kullback-Leibler distance* from the histogram for an account in the priming data to the histogram for the fraudulent data. For deciding which conditional distributions to keep in the signature, Cahill, *et al.* [10] compute the p-value for a Chi-square test for each account in the training data and keep only the conditioning variables that are statistically significant for the majority of accounts and highly statistically significant for at least certain accounts. Conditioning variables are added sequentially, until the incremental benefit from any of the remaining variables is too small to be statistically significant for a majority of accounts. To keep the signature current, they propose exponentially weighted moving averaging for updating signature components. The updated signature component, based on call $n + 1$, is given by

$$A_{n+1} = (1 - w)A_n + w\mathbf{X}_{n+1}, \tag{4.1}$$

where A_n denotes the account's signature component after call n, where $\mathbf{X}_{n+1}$ captures the attribute of call $n+1$, represented by a vector of 0's except for a 1 in the bin that contains the observed value of the call, and where w denotes the rate at which old calls are aged out of the signature component, which determines the effect of the current call on the component. To detect fraudulent activity, using signatures, Cahill, *et al.* [10] use a scoring method. They compare the call's attribute probability under the account signature to its probability under a fraudulent signature. The higher the call score, the more suspicious the call. For a call to obtain a high score, it has to be unexpected for the account.

Scott [35] outlines a paradigm for designing network intrusion detection systems based on stochastic models. Different networks have individual characteristics that should be considered, but rather than focussing on one detailed algorithm for a single type of network, Scott emphasises global aspects of intrusion detection common to most networks. Most networks share three qualities that set intrusion detection apart from other discrimination problems: criminal intrusion is rare, the networks to be screened typically generate massive amounts of data, and screening is to be done in real time. Scott [35] recommends Bayesian reasoning as the foundation for network intrusion detection systems. The most often cited quality of Bayesian reasoning, is its ability to include prior information, but Scott [35] notes that the greatest advantage of Bayesian methods is that they simplify the logic of building a coherent system. Let $\mathbf{X}$ represent observed data for an account and let the complementary events C and U denote whether fraudulent activity was present on the account while the data was generated, or whether the account was controlled solely by the user. In this case Bayes' theorem may be expressed as the posterior odds ratio

$$\frac{P[C|\mathbf{X}]}{P[U|\mathbf{X}]} = \frac{P[\mathbf{X}|C]P[C]}{P[\mathbf{X}|U]P[U]}.$$

The quantity $P[\mathbf{X}|C]/P[\mathbf{X}|U]$ is known as the Bayes factor for fraudulent activity. Bayes factors codify the evidence of an intrusion contained in the data. The prior distribution $P[C]$ is an important means of limiting false alarms in network intrusion detection. Scott [35] cited the paper by Cahill, *et al.* [10] as a good method for modelling telephone traffic. The method is based on account signatures, as explained above. The signature system collects information on telephone call characteristics believed to be good discriminators of customer and criminal behaviour. The system discretises continuous information and models the resulting multivariate categorical observations, using graphical methods. A signature is kept for each account, and a separate signature is maintained as an intruder profile. The signature system assigns to each incomming call a score interpretable as the log Bayes odds ratio in favour of the call having been produced by an intruder.

Hollmén, *et al.* [23] present a real-time fraud detection system, but one which is based on a *stochastic generative model*. In the generative model, they incorporate a variable *victimised*, which indicates whether or not the account has been victimised by a fraudster and a second variable *fraud*, which indicates whether or not the fraudster is currently performing fraud. Both variables are hidden. They have an observed variable *call*, which indicates whether a call is being made, or not. The transition probabilities from no-call to call and from call to no-call are dependent on the state of the variable fraud. Overall, they obtain a regime-switching stochastic time-series model which uses a Markov chain to implement switching, where the variables in the time-series are binary, and the switching variable has a hierarchical structure. Hollmén, *et al.* [23] argue that the benefit of a hierarchical structure is that it allows modelling of the time-series at different time scales. At the lowest hierarchical structure, dynamical behaviour of the individual calls is modelled, at the next level the transition from normal behaviour to fraudulent behaviour, and at the next level the transition to being victimised.

4.2.2 Designing Customer Behaviour Profiles

Chen, *et al.* [12] describe a method for reducing a database of transaction records when interest lies in the behaviour of the people making the transactions. They propose the use of histograms to summarise transaction behaviour dynamically, by discretising the variables, and updating the bin counts whenever a new call is made. Problems with this approach are that the intervals must be chosen, and choices which seem appropriate for one customer may be inappropriate for another. In addition, standard histograms give equal importance to recent and old transactions, and the customer's pattern of behaviour for any one variable may depend on other variables, so that a multidimensional histogram may be required. Signature designing starts with the collection of transaction records from a representative set of customers during a fixed period. It is proposed that one target signature is computed from the data of all customers with that target behaviour, and that these customers should be removed from the priming data, which is then used for designing customer specific signatures. The next step in signature designing is to choose histogram bin intervals. The task of binning signature variables is to choose the bin separation points so that the coarsened distribution is as close as possible to the original distribution for each customer. Using the priming data, bin separation points are chosen to maximise the *Kullback-Leibler distance* in order to balance the signature's ability to identify members of the target group, and its ability to avoid misclassification of a customer, who does not belong to the target group. The vector of transaction parameters for a customer may be written as a joint probability distribution, which may, in turn, be written as a product of one-dimensional distributions. The next step in designing a signature is to reduce this product of one-dimensional distributions by ignoring certain conditioning variables. Chen, *et al.* [12] state that, generally, X should be conditioned on Y if the distribution of X varies significantly with Y, which happens if the conditional distributions of X given Y differ significantly from the distribution of X that does not condition on Y. They propose a method of *forward model selection* for this task, where variables are added to the model sequentially until the gain from adding another conditioning variable is insignificant. It is important that a customer is assigned an initial signature as soon as the customer begins to transact. The task of assigning initial signatures resembles customer segmentation, where initial signature components constituting an initial signature, is assigned to the customer, based on the information available at the time of the assignment. A key feature of a signature is that it may be updated sequentially from current records. Chen, *et al.* [12] propose exponentially weighted moving averaging for updating signature components, where updating requires only the most recent histogram, the number of transactions made up to that point in time and a current transaction. A fixed weight is defined that controls the extent to which the signature component is affected by a new transaction and the rate at which the previous transaction is aged out.

The predictive model of each customer is updated with each transaction that the customer makes. Standard dynamic updating algorithms may often be used when the variable being updated may be modelled as a random draw from its signature component.

Lambert, *et al.* [29] argue that chronological variables are not observed at random, but in order; so all the Monday transactions for the week have to be observed before all the Tuesday transactions for the week, for example. They use a dynamic Poisson process to derive an approximation to a posterior mean that is almost as easy to compute as a linear estimate, and that predicts accurately on both real and simulated data. In other words, they derive a sequential estimator for timing distributions that is based on a Poisson model with periodic rates which may evolve over time. Updating histograms sequentially is not difficult when observations are randomly sampled, but when behaviour changes over time, a histogram of relative frequencies is inappropriate, because recent transactions have no more influence on the histogram than do old transactions. Evolving behaviour is tracked better by an exponentially weighted moving average. Timing variables, however, are not randomly sampled, and this makes unweighted averages and exponentially weighted moving averages inappropriate estimators. The key idea is to estimate the transaction rate for a period at the time of the transaction and then to estimate the probability for the period. Lambert, *et al.* [29] define an event-driven estimator, which is very close to the exact maximum likelihood estimator of a simulated Poisson process, as an approximate posterior mean of the transaction rate under a simple dynamic Poisson model.

Chen, *et al.* [11] introduce an incremental quantile estimator, based on stochastic approximation, to track call duration for a set of callers. The incremental estimate depends on its previous estimate and the current set of measurements, and require only a few arithmetic operations. They compare the performance of the exponentially weighted moving average to other incremental quantile estimators by means of a simulation study, when measurements are either normal or exponential, and the parameters of the distributions are either constant over time or changing linearly over time. As an application of the incremental quantile estimator, Chen, *et al.* [11] apply exponentially weighted stochastic approximation to track call duration for a set of telecommunication customers.

4.3 Industry Tested Fraud Detection Concepts

Moreau, *et al.* [30] suggest two approaches to user profiling, one which employs absolute analysis, and one which employs differential analysis of call data records. Call data records are transmitted to the network operator by the cells or switches with which the mobile telephone was communicating at the time, due to proximity. Moreau, *et al.* [30] state that existing fraud detection systems tend to interrogate sequences of call data records, comparing a function of the various fields by means of fixed criteria, known as triggers. A trigger, if activated, raises an alert status, which cumulatively would lead to an investigation by the network operator. Such fixed trigger systems perform what is known as an absolute analysis of call data records, and are good at detecting the extremes of fraudulent activity. Another approach to the problem is to perform a differential analysis. Here the behavioural patterns of the mobile telephone is monitored, and its most recent activities compared with a history of its usage. Criteria may then be derived to be used as triggers that are activated when usage patterns of a particular mobile telephone change significantly over a short period of time. As an initial approach to differential usages systems, the information of call data records are extracted and stored in record format. This process requires two windows over the sequence of transactions for each user. The shorter window is called the current user profile, and the longer window, the user profile history. When a new call data record arrives for a given user, the oldest entry from the user profile history is discarded, and the oldest entry from the current user profile moves to the back of the user profile history queue. The new record encoded from the incomming call data record then joins the back of the current user profile queue. Moreau, *et al.* [30] identify the following call data record components as the most relevant measures with respect to fraud detection, which should continually be picked out of the call data records and incorporated into the user profiles:

- an identification number, identifying the telephone user,
- the location of mobile originating calls,
- the duration of a call,
- an indicator to distinguish between national and international calls and
- the number dialled.

The following features are derived from the extracted call data record components:

- the number of mobile originated national calls per time interval,
- the number of mobile originated international calls per time interval,
- the total duration of mobile originated national calls per time interval,
- the total duration of mobile originated international calls per time interval,
- the number of hot (in the sense of high fraud probability) destinations per time interval,
- serveral statistical measures per time interval and
- fraud ranking.

Even over a long period of time, user behaviour should be efficiently storable, without loss of essential information. For this purpose Moreau, *et al.* [30] use two different methods of user profiling for the short-term window and for the long-term window. While user information gathered in the recent history is represented by current user profiles, information collected in the long term is stored in the user profile history. For the short-term window, several current user profiles may be stored to keep more detailed information for a longer time period. Moreau, *et al.* [30] maintain user behaviour profiles similar to that of Cahill, *et al.* [10] as defined in (§4.1).

4.4 Chapter Summary

The reader was provided with an overview of research performed in the field of cellular telecommunications fraud detection in this chapter. South African cellular network operators mainly rely on rule-based techniques to detect fraud, but modern fraud management systems making use of more sophisticated mathematical and statistical techniques to detect fraud are being implemented world-wide. The fraud detection literature considered in this chapter were divided according to the ability of fraud detection techniques to detect fraud at *fixed points in time* (§4.1) or in *real-time* (§4.2). Fixed-time fraud detection techniques, such as the classification of accounts as fraudulent or legitimate using *artificial neural networks*, *classification trees* and *Bayesian classification* are often found in computerised fraud management systems and will also be applied to real call data records later in this thesis. Real-time fraud detection techniques found in the literature make use of *account behaviour profiles*, also called *account signatures*. Real call data records will also be used later in this thesis to build account behaviour profiles employing cluster analysis and association rule mining. This chapter closed with a section (§4.3) providing insight into current implementation and maintenance of user behaviour profiles in the industry.

Chapter 5

Cellular Telephone Call Data

Call data records (CDRs) are produced by telephone switches on a per-call basis and contain all the infomation of the telephone call, be it from or to a subscriber belonging to the cellular network. Information included in a call data record are the telephone numbers involved in the call, the date and time of the call, the duration of the call, call features used (such as conference call or three way calling), identification of the cell transmitting the call to the subscriber's telephone, and more. Call data records form the source of billable records, containing the data necessary for billing systems to rate a particular call and bill the subscriber. Call data records do not contain tariff information, but mediation systems use the telephone number to enrich call data records by adding the subscriber's tariff to each record. Billing systems then use call data records and the tariff and duration information they contain to calculate a rate for each call.

5.1 Data Collection

The data set considered in this thesis originates from one of South Africa's cellular network operators, and consists of three components. The main data component consists of 2 127 261 call data records, giving the detail of each voice and text message (SMS) transaction, originating from and terminating on a subscriber's telephone. The data set contains transactions executed between April and September 2003, for 500 prepaid and 500 postpaid subscribers, chosen randomly from the operator's set of active subscribers. The attributes contained in each call data record are defined in Table 5.1. Call data record examples, with the subscriber's telehone number (MSISDN) and other party number encrypted (in order to protect the identity of the subscriber) are given in Table 5.2.

The description assigned to each cell may be used to derive its location, which is the second data component provided. Table 5.3 provides an example of cell descriptions. The third data component provides the tariff at which the subscriber's calls are rated, and examples may be found in Table 5.4. Meaning is given to call transaction types, by

Attribute Name	**Attribute Definition**
MSISDN	Acronym for Mobile Subscriber Integrated Services Digital Network (ISDN) number. The mobile subscriber's international telehone number.
Other_Party_Number	Other party present in the transaction. In the case of mobile originating transactions, is it the number dialled by the subscriber, and in the case of mobile terminating transactions, is it the party dialling the subscriber.
Location_Area_Code	Identification of a set of cells that are typically served by the same MSC.
Cell_Id	Identification of the cell closest to the cellular telephone when the call was made.
Call_Date	Date and time of transaction.
Call_Duration	Duration, measured in seconds, of mobile originating and mobile terminating calls. Call duration for SMS transactions is set to 0.
Call_Charge	Amount, measured in Rands and cents, charged for the transaction.
Call_Transaction_Type	Differentiation label for transaction types. Transaction types include mobile originating calls (MOC), mobile terminating calls (MTC), SMS originating from mobile subscriber's telephone (SMSO), and SMS terminating on mobile subscriber's telephone (SMSMT).
Inter_Seq_Num	The switch technology used by the network operator separate calls of long duration into shorter duration segments. Inter_Seq_Num joins these segments to form a call data record.
Subscriber_Type	Identification of the subscriber as a Prepaid or Postpaid subscriber.
Fraud_Ind	Classification of the call data record as a fraudulent or legitimate call.

Table 5.1: *Definitions of the attributes contained in each call data record.*

assigning a description to each type, which may be found in Table 5.5. Only voice calls originating from a subscriber's telephone, identified by transaction type 1, are used in further processing of the data, reducing the data set to 908 153 records, necessitated by resource requirements of complex data mining algorithms when applied to large sets of data.

The two kinds of fraud most often detected by cellular networks, are dealer- and subscription fraud. The incentives paid by cellular networks to dealers for obtaining new subscribers, and retaining current subscribers, are the main motivation behind committing dealer fraud. The dealer incentive scheme specifies that a subscriber must generate traffic on the network for the dealer to qualify for an incentive for that subscriber. In order to meet this requirement, dealers sometimes provision non-existing subscribers, and make one call on each subscriber's account, lasting only a few seconds. After the initial call, no additional traffic is generated for these subscribers. Call data records, 80 in total, with a duration of 3 seconds, and no charge, were added to the set of observations, exemplifying typical subscriber behaviour when this type of fraud is present (see Table

MSISDN	Other Party Number	Location Area Code	Cell Id	Call Date	Call Duration	Call Charge	Transaction Type	Inter Sequence Number	Subscriber Type	Fraud Ind
27556714598	114430336	141	11971	2003-04-03 20:46:28	6	0.00	1	0	P	0
27556714598	114430336	141	11971	2003-04-03 20:46:48	5	0.08	1	0	P	0
27556714598	114430336	141	11971	2003-04-03 20:47:04	6	0.00	1	0	P	0
27556714598	114430336	141	11971	2003-04-03 20:47:22	33	0.53	1	0	P	0
27556714598	558609273	141	11971	2003-04-05 18:19:48	39	0.62	1	0	P	0
27556794104	722637085	133	53041	2003-04-05 14:19:51	51	0.82	1	0	P	0
27552586369	722373047	182	13361	2003-04-03 16:39:44	9	0.54	1	0	P	0
27556774599	27835948801	181	11043	2003-04-16 14:43:08	14	0.84	1	0	P	0
27556734669	556606456	151	13183	2003-04-16 14:50:05	6	0.00	1	0	P	0
27556764681	721550814	411	21141	2003-04-05 16:18:31	17	0.27	1	0	P	0
27553200275	835067695	131	54332	2003-04-08 15:04:22	40	0.86	1	0	C	0
27553200315	556436568	131	54332	2003-04-08 15:11:49	35	0.75	1	0	C	0
27553200141	121	409	46027	2003-04-05 14:07:35	5	0.00	1	0	C	0
27553200377	437265876	409	20631	2003-04-05 14:08:37	59	0.76	1	0	C	0
27553200710	724330118	208	23302	2003-04-05 16:37:46	13	0.76	1	0	C	0
27553200453	722520017	305	36140	2003-04-05 13:45:24	2	0.02	1	0	C	0
27553200453	722520017	305	36140	2003-04-05 13:49:48	2	0.02	1	0	C	0
27553200453	555109492	305	36140	2003-04-05 13:51:01	45	0.55	1	0	C	0
27553200453	722520017	305	36140	2003-04-05 14:03:05	13	0.16	1	0	C	0
27552585562	27731661465	182	10272	2003-04-03 17:42:48	67	5.70	1	0	P	0
27556734669	55660556606456	151	13183	2003-04-16 14:47:30	131	7.86	1	0	P	0
27552585651	847139077	182	15711	2003-04-03 17:48:14	130	8.55	1	0	P	0
27556784674	27835551676	134	43069	2003-04-15 14:35:48	110	6.60	1	0	P	0
27556784656	836905719	151	13312	2003-04-03 20:33:54	236	7.35	1	0	P	0
27556724675	27552491000	182	44015	2003-04-03 16:31:03	85	5.70	1	0	P	0
27556764699	559531065	151	10713	2003-04-15 14:44:16	71	5.70	1	0	P	0
27556738557	312625852	306	31562	2003-04-15 14:51:09	103	5.70	1	0	P	0
27552535929	27834941373	202	26832	2003-04-23 20:34:39	917	24.80	1	0	P	0
27556768844	847412296	138	54680	2003-04-23 17:52:18	493	25.65	1	0	P	0
27556714598	725435717	141	11971	2003-04-07 21:53:27	1310	20.96	1	0	P	0
27556768888	555525587	161	11442	2003-04-11 11:46:56	346	20.76	1	0	P	0
27556754627	27837248154	142	10981	2003-04-30 18:17:11	404	24.24	1	0	P	0
27552586369	557669446	182	14182	2003-04-30 19:16:23	345	20.70	1	0	P	0
27552535929	554883818	183	17330	2003-04-01 16:05:27	465	22.80	1	0	P	0
27556768883	556872292	305	32172	2003-04-24 10:09:18	565	28.50	1	0	P	0
27556764646	935722516201	182	13091	2003-04-14 13:26:26	216	60.00	1	0	P	0
27552535929	834941373	202	26832	2003-04-24 19:44:52	1780	67.30	1	0	P	0
27556748904	926622324395	501	27200	2003-04-16 10:11:11	681	60.00	1	0	P	0
27556774667	447940371653	183	19162	2003-04-03 21:39:53	360	60.00	1	0	P	0
27556784675	9442089814465	152	12992	2003-08-27 16:55:05	333	60.00	1	0	P	0
27552535929	834941373	202	26832	2003-08-26 19:39:42	1180	57.00	1	0	P	0
27556709049	9447762838377	131	59491	2003-08-30 19:51:23	426	102.24	1	0	P	0
27553200526	215579086	205	26892	2003-04-07 09:17:48	1509	60.43	1	1	C	0
27553200907	836037663	161	13251	2003-04-01 12:59:15	1271	50.95	1	0	C	0
27553200099	9971503500026	409	24652	2003-04-08 16:55:39	847	96.07	1	0	C	0
27553200925	933621728161	133	53260	2003-04-09 10:51:55	1442	118.67	1	0	C	0

Table 5.2: *Extract from the set of call data records.*

Location		**Cell ID**
Area Code	**Cell ID**	**Description**
202	21411	Mowbray_1
202	26962	Batavia_2
202	46542	UCT_MC2
202	61560	Ysterplaat_Mobile
161	11121	Diepsloot-1
161	40644	Mellis_Court_MC2
161	40678	Cresta_Centre_MC4
161	14913	Didata_Campus-3
141	16232	Corporate_Park-2
141	16233	Corporate_Park-3
141	11082	Randjiesfontein-2
141	10622	Kyalami-2
141	11411	Sunninghill_Centre-1
141	11413	Sunninghill_Centre-3

Table 5.3: *Extract from the set of cell id descriptions.*

5.6). Subscription fraud occurs when a subscriber signs up for a service with fraudulently obtained subscriber information, or false identification. Cellphone theft, where a criminal makes calls on a subscriber's account, may also be seen as a type of subscription fraud. Subscription fraud may usually be identified by a large number of expensive calls, often to international destinations. Call data records for 30 subscribers were added to the set of observations, exemplifying subscription fraud (see Table 5.7).

MSISDN	**Tariff**	**Tariff Usage Group**
27552535929	4UP	Prepaid
27556748904	WNP	Leisure
27553200099	200	Low Business
27553200377	100	Low Business
27556768883	BUS	Low Business
27556754627	500	High Business
27553200453	S1T	High Business
27556768844	T1K	High Business

Table 5.4: *Extract from subscriber tariffs, grouping each tariff according to expected usage.*

Transaction Type	**Short Code**	**Description**
1	MOC	Mobile Originating Calls
2	MTC	Mobile Terminating Calls
29	CF	Call Forward
30	SMSO	SMS Mobile Originating
31	SMSMT	SMS Mobile Terminating

Table 5.5: *Call transaction type description.*

5.2 Data Preparation

Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically large sizes, often several terabytes or more. There are a number of data preprocessing techniques available to clean, integrate, transform and reduce the data. Data cleaning may be applied to remove noise and correct inconsistencies in the data. Data integration merges data from multiple sources into a coherent data store. Data transformation processes, such as normalisation, may be applied, improving the accuracy and efficiency of mining algorithms involving distance measurements. Data reduction may reduce the data set size by aggregating, eliminating redundant features, or clustering, for instance. These data processing techniques, when applied prior to mining, may improve the overall quality of the patterns mined and the time required for the actual mining substantially [21].

Switch technology used by the particular network operator in question, segments calls exceeding 1 780 seconds into separate call data records, each one no longer than 1 780 seconds, with the sum of the call segments not exceeding 3 591 seconds. Calls exceeding 3 591 seconds are terminated by the switch. Segmented call data records were merged during data preparation, resulting in one call data record per call for all calls. The attribute *Inter_Seq_Num*, forming part of a call data record, is used to identify segmented calls, and merge the segments into one call data record for each call.

To aid in the mining process, new attributes were constructed from the given set of attributes, and were added to the data set. The network operator's marketing brochures were used to assign a value to each subscription tariff, creating a new attribute *Subscriber_Tariff*. The attribute *Call_Date* was used to derive a binary attribute, *Peak_Ind*, indicating the rate period during which the call was made, replacing the attribute *Call_Date*. Calls made during the peak period are charged more than those made during the off-peak period. The peak period for cellular customers in South Africa is between $07h00$ and $20h00$, excluding weekends and public holidays. The attributes *Cell_Id* and *Location_Area_Code* were used to create a new attribute, *Cell_Location* (replacing these two attributes) deriving the cell's approximate location, as a distance in kilometres between

MSISDN	Other Party Number	Location Area Code	Cell ID	Call Date	Call Duration	Call Charge	Call Destination	Fraud Ind
27559990001	214171835	111	13051	2005-07-08 01:41:22	3	0.00	South Africa	1
27559990002	834541575	111	13051	2005-07-09 04:06:47	3	0.00	South Africa	1
27559990003	112	111	13051	2005-07-10 06:32:11	3	0.00	South Africa	1
27559990004	214171835	111	13051	2005-07-11 08:57:36	3	0.00	South Africa	1
27559990005	834541575	111	13051	2005-07-12 11:23:00	3	0.00	South Africa	1
27559990006	112	111	13051	2005-07-13 13:48:24	3	0.00	South Africa	1
27559990007	214171835	111	13051	2005-07-14 16:13:50	3	0.00	South Africa	1
27559990008	834541575	111	13051	2005-07-15 18:39:14	3	0.00	South Africa	1
27559990009	112	111	13051	2005-07-16 21:04:38	3	0.00	South Africa	1
27559990010	214171835	111	13051	2005-07-17 23:30:03	3	0.00	South Africa	1
27559990011	834541575	111	13051	2005-07-19 01:55:27	3	0.00	South Africa	1
27559990012	112	111	13051	2005-07-20 04:20:52	3	0.00	South Africa	1
27559990013	214171835	111	13051	2005-07-21 06:46:16	3	0.00	South Africa	1
27559990014	834541575	111	13051	2005-07-22 09:11:40	3	0.00	South Africa	1
27559990015	112	111	13051	2005-07-23 11:37:05	3	0.00	South Africa	1
27559990016	214171835	111	13051	2005-07-24 14:02:29	3	0.00	South Africa	1
27559990017	834541575	111	13051	2005-07-25 16:27:53	3	0.00	South Africa	1
27559990018	112	111	13051	2005-07-26 18:53:18	3	0.00	South Africa	1
27559990019	214171835	111	13051	2005-07-27 21:18:43	3	0.00	South Africa	1
27559990020	834541575	111	13051	2005-07-28 23:44:08	3	0.00	South Africa	1
27559990021	112	111	13051	2005-07-30 02:09:32	3	0.00	South Africa	1
27559990022	214171835	111	13051	2005-07-31 04:34:56	3	0.00	South Africa	1
27559990023	834541575	111	13051	2005-08-01 07:00:21	3	0.00	South Africa	1
27559990024	112	111	13051	2005-08-02 09:25:45	3	0.00	South Africa	1
27559990025	214171835	111	13051	2005-08-03 11:51:09	3	0.00	South Africa	1
27559990026	834541575	111	13051	2005-08-04 14:16:34	3	0.00	South Africa	1
27559990027	112	111	13051	2005-08-05 16:41:58	3	0.00	South Africa	1
27559990028	214171835	111	13051	2005-08-06 19:07:23	3	0.00	South Africa	1
27559990029	834541575	111	13051	2005-08-07 21:32:47	3	0.00	South Africa	1
27559990030	112	111	13051	2005-08-08 23:58:11	3	0.00	South Africa	1
27559990031	214171835	111	13051	2005-08-10 02:23:37	3	0.00	South Africa	1
27559990032	834541575	111	13051	2005-08-11 04:49:01	3	0.00	South Africa	1
27559990033	112	111	13051	2005-08-12 07:14:25	3	0.00	South Africa	1
27559990034	214171835	111	13051	2005-08-13 09:39:50	3	0.00	South Africa	1
27559990035	834541575	111	13051	2005-08-14 12:05:14	3	0.00	South Africa	1
27559990036	112	111	13051	2005-08-15 14:30:39	3	0.00	South Africa	1
27559990037	214171835	111	13051	2005-08-16 16:56:03	3	0.00	South Africa	1
27559990038	834541575	111	13051	2005-08-17 19:21:27	3	0.00	South Africa	1
27559990039	112	111	13051	2005-08-18 21:46:52	3	0.00	South Africa	1
27559990040	214171835	111	13051	2005-08-20 00:12:16	3	0.00	South Africa	1
27559990041	834541575	111	13051	2005-08-21 02:37:40	3	0.00	South Africa	1
27559990042	112	111	13051	2005-08-22 05:03:05	3	0.00	South Africa	1
27559990043	214171835	111	13051	2005-08-23 07:28:30	3	0.00	South Africa	1
27559990044	834541575	111	13051	2005-08-24 09:53:55	3	0.00	South Africa	1
27559990045	112	111	13051	2005-08-25 12:19:19	3	0.00	South Africa	1
27559990046	214171835	111	13051	2005-08-26 14:44:43	3	0.00	South Africa	1
27559990047	834541575	111	13051	2005-08-27 17:10:08	3	0.00	South Africa	1
27559990048	112	111	13051	2005-08-28 19:35:32	3	0.00	South Africa	1
27559990049	214171835	111	13051	2005-08-29 22:00:56	3	0.00	South Africa	1
27559990050	834541575	111	13051	2005-08-31 00:26:21	3	0.00	South Africa	1
27559990051	112	111	13051	2005-09-01 02:51:45	3	0.00	South Africa	1
27559990052	214171835	111	13051	2005-09-02 05:17:10	3	0.00	South Africa	1
27559990053	834541575	111	13051	2005-09-03 07:42:34	3	0.00	South Africa	1
27559990054	112	111	13051	2005-09-04 10:07:58	3	0.00	South Africa	1
27559990055	214171835	111	13051	2005-09-05 12:33:24	3	0.00	South Africa	1
27559990056	834541575	111	13051	2005-09-06 14:58:48	3	0.00	South Africa	1
27559990057	112	111	13051	2005-09-07 17:24:12	3	0.00	South Africa	1
27559990058	214171835	111	13051	2005-09-08 19:49:37	3	0.00	South Africa	1
27559990059	834541575	111	13051	2005-09-09 22:15:01	3	0.00	South Africa	1
27559990060	112	111	13051	2005-09-11 00:40:26	3	0.00	South Africa	1

Table 5.6: *Extract from call data records exemplifying dealer fraud.*

MSISDN	Other Party Number	Location Area Code	Cell ID	Call Date	Call Duration	Call Charge	Call Destination	Fraud Ind
27554435190	992215887322	301	47512	2003-08-28 13:34:34	500	171.21	Pakistan	1
27554435190	992215887330	301	47512	2003-08-28 16:34:34	400	151.21	Pakistan	1
27554435190	992215887220	301	47509	2003-08-28 12:34:34	440	160.17	Pakistan	1
27552546298	991215854874	301	47509	2003-08-10 12:34:34	3591	1292.21	India	1
27552546298	991215854130	301	47509	2003-08-10 11:34:34	359	129.51	India	1
27552546298	991215854120	301	47509	2003-08-10 15:34:34	3590	1256.21	India	1
27552546298	834435190	301	47509	2003-08-10 17:34:34	3590	120.21	South Africa	1
27552546298	834435190	301	47509	2003-08-10 19:34:34	3590	120.21	South Africa	1
27552546298	834435190	301	47518	2003-08-10 21:34:34	3590	120.21	South Africa	1
27552546298	834435200	301	47518	2003-08-10 22:34:34	3590	120.21	South Africa	1
27552585886	27557280552	111	13161	2003-09-23 19:40:42	41	2.85	South Africa	1
27552585886	27726060672	111	13161	2003-09-24 13:55:21	4	2.85	South Africa	1
27552585886	27558594554	111	13161	2003-09-25 16:35:39	12	1.51	South Africa	1
27552585886	27836365342	111	13161	2003-09-28 12:12:41	11	2.02	South Africa	1
27552585886	27558594554	111	13161	2003-09-28 14:19:34	34	1.51	South Africa	1
27552585886	27558336696	111	13161	2003-09-25 20:45:48	71	1.18	South Africa	1
27552585886	27553203320	111	13161	2003-09-22 11:14:25	4	1.51	South Africa	1
27552585886	5575040201	111	13161	2003-09-24 09:52:41	68	1.49	South Africa	1
27552585886	27555010630	111	13161	2003-09-26 18:23:48	2	1.51	South Africa	1
27552585886	27437451055	111	13161	2003-09-30 07:53:47	37	1.51	South Africa	1
27552585886	27558336696	111	13161	2003-09-26 17:19:01	74	2.26	South Africa	1
27552585886	27558336696	111	13161	2003-09-27 08:46:48	20	0.79	South Africa	1
27552585886	27126530213	111	13161	2003-09-27 06:52:13	30	0.79	South Africa	1
27552585886	5551461113	111	13161	2003-09-30 08:35:12	36	0.79	South Africa	1
27552585886	27554554342	111	13161	2003-09-24 10:06:04	316	8.30	South Africa	1
27552585886	27557263357	111	13161	2003-09-25 17:51:22	16	1.51	South Africa	1
27552585886	8474615656	111	13161	2003-09-24 08:44:15	135	3.44	South Africa	1
27552585886	27835449801	111	13161	2003-09-30 12:24:26	12	0.56	South Africa	1
27552585886	27835449801	111	13161	2003-09-25 22:43:12	8	0.12	South Africa	1
27552585886	27835449801	111	13161	2003-09-30 13:16:36	11	0.51	South Africa	1
27552585886	27583035251	111	13161	2003-09-26 15:52:44	159	4.53	South Africa	1
27552585886	27559223296	111	13161	2003-09-25 12:43:36	45	1.51	South Africa	1
27552585886	27589133254	111	13161	2003-09-25 09:33:25	5	1.51	South Africa	1
27552585886	27835847216	111	13161	2003-09-27 19:09:44	23	0.92	South Africa	1
27552585886	27837610583	111	13161	2003-09-24 17:11:56	20	2.41	South Africa	1
27552585886	27837080508	111	13161	2003-09-22 18:21:31	79	3.61	South Africa	1
27552585886	27722439951	111	13161	2003-09-27 16:27:28	27	0.79	South Africa	1
27552585886	27559723610	111	13161	2003-09-24 10:14:50	225	6.32	South Africa	1
27552585886	27722479391	111	13161	2003-09-23 20:54:37	21	0.27	South Africa	1
27552585886	27555641270	111	13161	2003-09-24 18:20:06	22	1.58	South Africa	1
27552585886	27555641270	111	13161	2003-09-24 18:13:20	62	2.37	South Africa	1
27552585886	27555641270	111	13161	2003-09-24 18:46:34	49	1.58	South Africa	1
27552585886	27832363020	111	13161	2003-09-30 19:26:41	56	2.41	South Africa	1
27552585886	27555641270	111	13161	2003-09-27 16:28:13	20	0.79	South Africa	1
27552585886	27832363020	111	13161	2003-09-24 18:36:28	14	2.41	South Africa	1
27552585886	27415554939	111	13161	2003-09-28 19:50:30	46	2.37	South Africa	1
27552585886	27115527700	111	13161	2003-09-28 13:28:07	28	1.54	South Africa	1
27552585886	27557226209	111	13161	2003-09-27 16:29:31	2	0.79	South Africa	1
27552585886	27557959701	111	13161	2003-09-23 09:23:10	5	1.54	South Africa	1
27552585886	27557226209	111	13161	2003-09-27 14:59:34	67	1.18	South Africa	1
27552585886	27556511401	111	13161	2003-09-28 16:26:48	43	1.54	South Africa	1
27552585886	27724895389	111	13161	2003-09-25 15:10:59	113	3.08	South Africa	1
27552585886	27725702297	111	13161	2003-09-29 10:25:52	20	0.44	South Africa	1
27552585886	27836856959	111	13161	2003-09-29 18:32:49	57	2.02	South Africa	1
27552585886	27413604533	111	13161	2003-09-23 19:32:57	48	1.54	South Africa	1
27552585886	27553329144	111	13161	2003-09-24 10:35:16	59	3.54	South Africa	1
27552585886	27553334106	111	13161	2003-09-23 16:42:20	32	2.85	South Africa	1

Table 5.7: *Extract from call data records exemplifying subscription fraud.*

the provincial capital in which the cell is located, and a fixed point, taken as Johannesburg [16]. The country and area code were extracted from the number dialled, contained in the attribute *Other_Party_Number*, and were used to determine the distance of each call, storing the result in attribute *Call_Distance* [25]. The attribute *Subscriber_Type*, taking the values C (contract/postpaid) or P (prepaid), was transformed into a binary attribute, *Prepaid_Ind*, with 1 indicating a prepaid subscriber, and 0 a postpaid subscriber.

The final set of attributes was grouped into explanatory and response variables, and a variable name was assigned to each. The results are shown in Table 5.8, and a subset of observations, using these variable names, are given in Table 5.9.

Attribute Name	**Variable Name**	**Variable Type**	**Variable Data Type**
Prepaid_Ind	$\mathbf{x}_1$	Explanatory	Binary
Subscriber_Tariff	$\mathbf{x}_2$	Explanatory	Nominal
Peak_Ind	$\mathbf{x}_3$	Explanatory	Binary
Cell_Location	$\mathbf{x}_4$	Explanatory	Continuous
Call_Distance	$\mathbf{x}_5$	Explanatory	Continuous
Call_Charge	$\mathbf{x}_6$	Explanatory	Continuous
Call_Duration	$\mathbf{x}_7$	Explanatory	Continuous
Call_Transaction_Type	$\mathbf{x}_8$	Explanatory	Categorical
Fraud_Ind	$\mathbf{Y}$	Response	Binary

Table 5.8: *Variable definitions and types.*

Distance-based mining algorithms, such as neural networks or clustering techniques, provide better results if the data to be analysed have been normalised, that is, scaled to a specific range, such as $[0.0, 1.0]$. When one variable takes many more values than another, it will typically outweigh distance measurements taken on the other variables if left unnormalised [21]. Continuous variables in the data set were therefore normalised using *min-max normalisation*. In min-max normalisation one performs a linear transformation on the original data. Suppose that $\min(\mathbf{x}_j)$ and $\max(\mathbf{x}_j)$ are the minimum and maximum values of the variable $\mathbf{x}_j$. Min-max normalisation maps the i-th value x_{ji} of $\mathbf{x}_j$ to x'_{ji} in the new range $[\min(\mathbf{x}'_j), \max(\mathbf{x}'_j)]$ by means of transformation

$$x'_{ji} = \frac{x_{ji} - \min(\mathbf{x}_j)}{\max(\mathbf{x}_j) - \min(\mathbf{x}_j)}(\max(\mathbf{x}'_j) - \min(\mathbf{x}'_j)) + \min(\mathbf{x}'_j). \tag{5.1}$$

Min-max normalisation preserves the relationship among the original data values [21].

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{Y}$
1	1	0	0	0	1.44	90	1	0
1	1	0	0	0	0.08	5	1	0
1	1	0	0	0	0.53	33	1	0
1	1	0	0	0	0.62	39	1	0
1	1	1	442	0	1.62	27	1	0
1	1	1	0	0	2.85	28	1	0
1	1	1	442	0	3.30	55	1	0
1	1	0	0	0	1.55	33	1	0
1	1	0	0	0	4.65	122	1	0
1	1	0	395	0	2.42	151	1	0
1	1	0	395	0	0.82	51	1	0
1	1	0	1 265	0	0.27	17	1	0
1	1	0	395	0	0.91	57	1	0
1	1	1	0	0	8.55	130	1	0
1	1	0	239	0	0.53	33	1	0
1	1	1	0	0	2.85	2	1	0
1	1	1	159	0	2.85	3	1	0
1	1	1	395	0	6.60	110	1	0
1	1	0	0	0	16.65	579	1	0
1	1	1	0	0	2.40	40	1	0
1	1	0	0	451	10.00	83	1	0
1	1	1	300	304	16.68	139	1	0
1	1	0	159	276	3.12	26	1	0
1	1	0	159	276	2.16	18	1	0
1	1	0	0	451	31.92	266	1	0

Table 5.9: *Extract from the final set of call data records, grouping attributes into explanatory and response variables.*

5.3 Variable Selection

Variable selection techniques are usually the first step in statistical analyses, and assists in making a choice among numerous explanatory variables for inclusion into predictive models. In this section regression techniques are applied to select a subset of explanatory variables from the data set best predicting the response variable.

5.3.1 Linear Regression

Linear regression analysis is only useful when the response variable is continuous. This supports the introduction of a new response variable, termed fraud weight and given the variable name $\mathbf{Y}'$, indicating the severity of fraud for each transaction. Fraud weight is set to 0 for legitimate calls, and equal to call charge for fraudulent calls. To obtain a preliminary impression of the data, the descriptive statistics of the data set are examined, as well as the Pearson's correlation between response variable $\mathbf{Y}^{'}$ and the continuous explanatory variables, as shown in Table 5.10.

Variable Name	**Minimum**	**Maximum**	**Mean**	**Standard Deviation**	**Correlation**
$\mathbf{x}_2$	1	1 490	443.1	366.262	−0.006
$\mathbf{x}_4$	0	1 265	319.71	405.382	−0.002
$\mathbf{x}_5$	0	13 097	5.17	210.639	0.173
$\mathbf{x}_6$	0	1 292.21	1.681	5.692	0.849
$\mathbf{x}_7$	0	3 591	65.17	124.414	0.113

Table 5.10: *Descriptive statistics of the continuous explanatory variables $\boldsymbol{x}_2$ (Subscriber Tariff), $\boldsymbol{x}_4$ (Cell Location), $\boldsymbol{x}_5$ (Call Distance), $\boldsymbol{x}_6$ (Call Charge) and $\boldsymbol{x}_7$ (Call Duration).*

Statistics of categorical variables $\mathbf{x}_1$ (Prepaid Indicator) and $\mathbf{x}_3$ (Peak Indicator) were examined by means of frequency tables, which may be found in Table 5.11 and Table 5.12 respectively. Note that the response variable $\mathbf{Y}^{'}$ is most highly correlated with the explanatory variable $\mathbf{x}_6$ (Call Charge). This is to be expected, since fraud weight is derived from call charge. It is possible to derive a regression equation using all the explanatory variables. However, since some of the explanatory variables are strongly

Value	**Frequency**	**Percentage of Total**
0	680 779	75.0
1	226 948	25.0
Total	907 727	100.0

Table 5.11: *Frequency table for the categorical variable $\boldsymbol{x}_1$ (Prepaid Indicator) in the set of call data records.*

Value	**Frequency**	**Percentage of Total**
0	359 403	39.59
1	548 324	60.41
Total	907 727	100.00

Table 5.12: *Frequency table for the categorical variable $\boldsymbol{x}_3$ (Peak Indicator) in the set of call data records.*

interrelated, it is more efficient to use only a subset of explanatory variables to derive the regression equation.

The variable selection problem may be described as considering certain subsets of explanatory variables and selecting that subset that either maximises or minimises an appropriate criterion. Two obvious subsets are the best single variable and the complete set of explanatory variables. The problem lies in selecting an intermediate subset that may lie somewhere between both these extremes. Stepwise regression using the forward selection method, as described in §2.2.3.1, is employed for making this choice. The forward selection method starts by choosing the explanatory variable with the highest absolute correlation with $\mathbf{Y}^{'}$. In the set of observations used, $\mathbf{x}_6$ (Call Charge), is chosen as the best single predictor of $\mathbf{Y}^{'}$. The partial correlations between $\mathbf{Y}^{'}$ and each of the other explanatory variables, after removing the linear affect of $\mathbf{x}_6$ (Call Charge), are shown in Table 5.13. Explanatory variable $\mathbf{x}_7$ (Call Duration) is the next variable chosen by the forward selection method.

	Correlation
$\mathbf{x}_2$	0.126
$\mathbf{x}_4$	−0.041
$\mathbf{x}_5$	−0.017
$\mathbf{x}_7$	−0.718

Table 5.13: *Partial pearson's correlation matrix after removing $\boldsymbol{x}_6$ (Call Charge) in the forward selection method.*

Afifi, *et al.* [1] suggested that a minimum F-to-enter value be used as stopping rule, but other criteria may also be used as the basis for a stopping rule. The coefficient of determination (R^2) may be used, terminating the process when the increase in R^2 is a very small amount. Alternatively, the series of adjusted coefficient of determination ($\overline{R}^2$) values may be examined, and the process terminated when $\overline{R}^2$ is maximised. The forward selection method introduces one variable into the regression model at each step, computing the F-to-enter value. The computed F-to-enter value, coefficient of determination (R^2) and adjusted coefficient of determination ($\overline{R}^2$), computed at each step of the forward selection method, are shown in Table 5.14. Using the coefficient of determination (R^2) as stopping rule, and terminating the process when no increase in R^2 is achieved by adding an additional variable into the regression model, results in four variables being added to the model, namely $\mathbf{x}_6$ (Call Charge), $\mathbf{x}_7$ (Call Duration), $\mathbf{x}_2$ (Subscriber Tariff) and $\mathbf{x}_5$ (Call Distance). The inclusion of $\mathbf{x}_6$ (Call Charge), $\mathbf{x}_7$ (Call Duration) and $\mathbf{x}_5$ (Call Distance) into the regression model confirms the assumption that fraudsters committing subscription fraud make expensive calls of long duration, some to foreign destinations, since they will not be held responsible for the charges generated. The correlation matrix in Table 5.10

	Variable Entered	F-to-enter	$\mathbf{R}^2$	$\overline{\mathbf{R}}^2$
1	$\mathbf{x}_6$	2 335 677	0.720	0.720
2	$\mathbf{x}_7$	2 895 539	0.864	0.864
3	$\mathbf{x}_2$	2 000 287	0.869	0.869
4	$\mathbf{x}_5$	1 521 224	0.870	0.870
5	$\mathbf{x}_4$	1 217 470	0.870	0.870

Table 5.14: *Selection of variables $\boldsymbol{x}_6$ (Call Charge), $\boldsymbol{x}_7$ (Call Duration), $\boldsymbol{x}_2$ (Subscriber Tariff), $\boldsymbol{x}_5$ (Call Distance) and $\boldsymbol{x}_4$ (Cell Location) in the forward selection method, using linear regression.*

confirms the assumption that long duration expensive calls are good indicators of fraud. Variable $\mathbf{x}_2$ (Subscriber Tariff), on the other hand, may be a good indicator of dealer fraud, since incentives paid to dealers are based on the subscriber's tariff.

5.3.2 Logistic Regression

It is often the case that the response variable is discrete, taking on two or more possible values. Over the decade 1995 – 2005 the logistic regression model has become, in many fields, the standard method of analysis in this situation [24]. The observations used here, depicting the behaviour of ligitimate and fraudulent subscribers in a cellular telephone network, is a typical example of a set of explanatory variables describing a binary response variable.

The variable selection process starts with a model that does not include any of the explanatory variables, as described in §2.2.3.1. At each step the explanatory variable with the largest score statistic, whose significance value is less than 0.05, is selected. Table 5.15 indicates the score statistics of each explanatory variable at the start of the process. The first step of this method selects explanatory variable $\mathbf{x}_6$ (Call Charge) for inclusion. The selection of variable $\mathbf{x}_6$ (Call Charge) produces the logistic regression model summarised in

Variable	Score Statistic	Degrees of Freedom	Significance
$\mathbf{x}_2$	77.399	1	0
$\mathbf{x}_4$	66.459	1	0
$\mathbf{x}_5$	8 999.084	1	0
$\mathbf{x}_6$	81 648.927	1	0
$\mathbf{x}_7$	4 216.900	1	0

Table 5.15: *Variable score statistics for the logistic regression forward select method of the continuous explanatory variables $\boldsymbol{x}_2$ (Subscriber Tariff), $\boldsymbol{x}_4$ (Cell Location), $\boldsymbol{x}_5$ (Call Distance), $\boldsymbol{x}_6$ (Call Charge) and $\boldsymbol{x}_7$ (Call Duration).*

Table 5.16, with an associated log–likelihood estimate calculated as $L(\hat{\beta}_1) = -1\,350.287$. The maximum likelihood estimates of β_0 and β_1 are $\hat{\beta}_0 = -8.947$ and $\hat{\beta}_1 = 0.063$. The fitted values are given by

Variable	Estimated Coefficient	Standard Error
$\mathbf{x}_6$	0.063	0.003
Constant	–8.947	0.088

Table 5.16: *Fitting the logistic regression model to the data.*

$$\hat{\pi}(\mathbf{x}_6) = \frac{e^{-8.947+0.063\mathbf{x}_6}}{1+e^{-8.947+0.063\mathbf{x}_6}}$$

and the estimated logistic transformation, $\hat{g}(\mathbf{X}_i)$, is given by

$$\hat{g}(\mathbf{x}_6) = -8.947 + 0.063\mathbf{x}_6.$$

After estimating the coefficients of the logistic regression model the significance of the variables are assessed. One approach is to compare the observed values of the response variable to the predicted values obtained from models with and without the variable in question, calculating this statistic as $G(\mathbf{x}_6, y_i) = 643.262$. The logistic regression process continues in this fasion, adding one additional explanatory variable at each step, and testing the significance of the coefficients. Table 5.17 indicates the variables included in the model at each step and the statistics calculated. The logistic regression model evaluated in step 4 of Table 5.17 consists of a similar set of variables as the set selected using linear regression, with the exception of variable $\mathbf{x}_5$ (Call Distance). Increasing the significance threshold value from 0.05 to 0.076 results in an additional step of the variable selection process, adding variable $\mathbf{x}_5$ (Call Distance) to the logistic regression model.

Model	$\mathbf{X}_i$	$G(\mathbf{X}_i, y_i)$	$L(\beta_1, \ldots, \beta_P)$
1	$\mathbf{x}_6$	643.262	$-1\,350.287$
2	$\mathbf{x}_6, \mathbf{x}_4$	95.844	$-1\,302.365$
3	$\mathbf{x}_6, \mathbf{x}_4, \mathbf{x}_2$	34.507	$-1\,285.112$
4	$\mathbf{x}_6, \mathbf{x}_4, \mathbf{x}_2, \mathbf{x}_7$	4.837	$-1\,282.894$

Table 5.17: *Logistic regression model summary indicating the variables included at each step of the model building process.*

The variable selection methods applied in this section were not used to exclude certain variables from further analysis, but rather to provide insight into the characteristics of the data used in this thesis. It is clear from both the methods applied that explanatory variable $\mathbf{x}_6$ (Call Charge) is the best single predictor of fraud.

5.4 Chapter Summary

This chapter was devoted to describing the call data records collected from one of South Africa's cellular network operators. The call data record attributes were defined in Table 5.1 and an extract of call data records given in Table 5.2. Artificially created call data records signifying fraudulent behaviour were introduced to the set of call data records in §5.1. Data preparation techniques were discussed in §5.2. New attributes were constructed from the given set of attributes and added to the data set to aid in the mining process. Linear and logistic regression techniques were applied in §5.3 as part of forward variable selection, providing insight into the characteristics of the call data set. The data will be transformed further in the next chapter, in which well-known data mining methods are applied to the results in order to aid in the fraud detection process.

Chapter 6

Application of Fraud Detection Methods to Call Data

The data mining methodologies described in Chapter 2 are applied in this chapter to the data set prepared in Chapter 5. The methods of decision trees (§6.1), artificial neural networks (§6.2), Bayesian decision making (§6.3), cluster analysis (§6.4), outlier analysis (§6.5) and association rule mining (§6.6) are applied, discussed and their fraud detection abilities are assessed.

6.1 Decision Trees

Most fraud management systems implemeted by cellular service providers use rule-based methods to detect and prevent fraud. The rules entered into these systems are derived from fraud scenarios experienced by the service provider, or by using trial and error methods. However, decision trees may also play a significant role in deriving rules to be used in rule-based fraud management systems, using call data records classified as fraudulent or legitimate to train decision trees. The rules derived from the decision trees may then be entered into rule-based fraud management systems to detect and prevent future fraudulent attempts using similar methods.

The observations in the reduced data set described in Chapter 5 were further processed by calculating daily statistics for each subscriber. The calculated statistics for each subscriber per day, included call count, total call duration, maximum call duration and call duration standard deviation. The same set of statistics were calculated for call charge. The relevant variable names and definitions are listed in Table 6.1. The daily statistics per subscriber were averaged to a set of statistics per subscriber. An excerpt of the daily subscriber statistics may be found in Table 6.2. The daily subscriber statistics consist of 1 085 observations, including 85 classified as fraudulent.

The tree derived in this section (see Appendix A.1 for a description of the computer

Attribute Name	Variable Name	Variable Type
Call_Count	$\mathbf{x}_1$	Explanatory
Call_Duration_Summ	$\mathbf{x}_2$	Explanatory
Call_Duration_Max	$\mathbf{x}_3$	Explanatory
Call_Duration_Stddev	$\mathbf{x}_4$	Explanatory
Call_Charge_Summ	$\mathbf{x}_5$	Explanatory
Call_Charge_Max	$\mathbf{x}_6$	Explanatory
Call_Charge_Stddev	$\mathbf{x}_7$	Explanatory
Fraud_Ind	$\mathbf{Y}$	Response

Table 6.1: *Classification tree variable definition.*

programs) is a classification tree, as defined in §2.1.1, with measurement vector $\mathbf{X}_i$, and response variables $y_i \in \mathbf{Y}$, where $N = 1\,085$. The response variables y_i of the observations fall into one of two classes, C_1 or C_2, belonging to $\mathcal{C}$, indicating call behaviour as respectively fraudulent or legitimate. The classification tree assigns class membership in $\mathcal{C}$ to every measurement vector $\mathbf{X}_i$ in $\mathcal{X}$. The learning sample $\mathcal{L}$, consisting of 1 085 observations, was used to build the classification tree, $\mathbf{d}(\mathbf{X}_i)$. The rules describing the classification tree are as follows, with the confidence measure given in brackets:

1. IF $x_2 > 3.5$ and $x_5 \leq 283.965$ THEN legitimate (99.9%)

2. IF $x_2 \leq 3.5$ THEN fraudulent (100%)

3. IF $x_2 > 3.5$ and $x_5 > 283.965$ THEN fraudulent (100%).

Rule 2 shows that fraudulent subscribers make calls of total daily duration less than or equal to 3.5 seconds. This rule is able to detect dealer fraud, of which examples are shown in Table 6.2, MSISDN 2777990001 to 27779990009. Rule 3 attempts to detect subscription fraud, identified by calls with a daily duration of more than 3.5 seconds and a daily charge of more than R283.97. Rule 1 classifies the remaining subscribers as legitimate.

Applying these rules to the learning sample $\mathcal{L}$ results in one misclassification. The V-fold cross-validation method, as defined in §2.1, was employed to estimate the accuracy of the classification tree. The observations in $\mathcal{L}$ were randomly partitioned into $V = 5$ subsets of approximately equal size. The test sample estimates for the rates of misclassification, using $\mathcal{L} \setminus \mathcal{L}_v$ to obtain the classifier $d^{(v)}$ and $\mathcal{L}_v$ to estimate the accuracy of the classifier, $v \in \{1, \ldots, 5\}$, were calculated as $R_c(d^{(1)}) = \frac{0}{219}$, $R_c(d^{(2)}) = \frac{2}{216}$, $R_c(d^{(3)}) = \frac{1}{217}$, $R_c(d^{(4)}) = \frac{0}{218}$ and $R_c(d^{(5)}) = \frac{1}{215}$. The proportion misclassified by classification tree d is therefore estimated as $R_c(d) = \frac{4}{1085}$. The classification quality may be inspected in more detail via a confusion matrix, comparing the observed response variables y_i with the responses generated by the classification tree $\hat{y}_i$. The confusion matrix, given in Table

MSISDN	Call Count	Summed Call Duration	Max Call Duration	Stddev Call Duration	Summed Call Charge	Max Call Charge	Sttdev Call Charge	Fraud Ind
27772530985	1.286	121	102.714	14.186	3.586	3.586	0.192	0
27774435191	26	55282	3590	953.23	3847.53	403.92	67.823	1
27772534709	2.889	108.889	54.611	12.149	9.528	3.986	0.581	0
27772535924	3.115	119.065	64.496	17.074	6.975	3.481	0.925	0
27772543718	2.31	122034	83.414	21.392	5.954	4.111	1.148	0
27772546298	3.11	135.937	73.307	19.538	6.331	4.385	1.787	0
27772546299	8	20900	3511	1030.603	3108.67	1252.11	420.112	1
27774435192	5.222	254.333	104.556	35.012	10.272	3.372	0.905	1
27776794646	1.767	50.163	28.605	3.683	1.62	0.68	0.148	0
27776794649	3.436	107.939	58.558	18.037	3.262	1.93	0.633	0
27776794650	10.458	322.516	68.632	27.551	11.908	3.786	1.07	0
27776794683	2.176	191.627	122.824	27.023	5.436	3.548	0.584	0
27776794684	7.466	377.733	123.116	37.08	12.577	3.839	1.174	0
27776794685	4.012	195	104.919	35.318	5.176	2.862	0.902	0
27776794693	4.784	244.275	134.127	40.475	4.804	2.88	0.585	0
27774435190	1	4	4	0	0	0	0	0
27779990001	1	3	3	0	0	0	0	1
27779990002	1	3	3	0	0	0	0	1
27779990003	1	3	3	0	0	0	0	1
27779990004	1	3	3	0	0	0	0	1
27779990005	1	3	3	0	0	0	0	1
27779990006	1	3	3	0	0	0	0	1
27779990007	1	3	3	0	0	0	0	1
27779990008	1	3	3	0	0	0	0	1
27779990009	1	3	3	0	0	0	0	1

Table 6.2: *Extract from daily subscriber statistics with call duration measured in seconds and call change measured in Rands.*

		$\hat{y}_i$		
		0	1	Errors
y_i	0	1 000	0	0
	1	1	84	1
	Errors	1	0	

Table 6.3: *Classification tree confusion matrix indicating* 1 *false negative.*

6.3, indicates one false negative, where the observed response indicates fraud, but the classification tree indicates the observation as legitimate subscriber behaviour. This false negative is caused by MSISDN 27774435192 shown in Table 6.2, describing a subscriber with average daily call duration 254.333s and call charge R10.272. This behaviour is classified as legitimate behaviour by the rules derived from the classification tree.

The ability of this method to detect fraud may be demonstrated by applying the classification tree to the unseen set of daily statistics on a fraudulent subscriber's account, given in Table 6.4. Rule 3 derived from the classification tree classifies this subscriber's behaviour as potential fraud.

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
7	21900	3591	1130.673	3158.77	1292.21	520.582

Table 6.4: *Daily statistics on the account of fraudulent subscriber 27895500021 with call count (x_1) 7, total call duration (x_2) $21\,900s$, maximum call duration (x_3) $3\,591s$, call duration standard deviation (x_4) $1\,130.673s$, total call charge (x_5) $R3\,158.77$, maximum call charge (x_6) $R1\,292.21$ and call charge standard deviation (x_7) $R520.582$.*

6.2 Artificial Neural Networks

A feed-forward neural network may be used to represent an arbitrary non-linear relationship, provided that observations exist exemplifying relationships as input-output pairs. In this section a three-layer feed-forward neural network was used, based on the paradigm of supervised learning, to learn a discriminative function able to classify subscribers, using summary statistics.

The same set of data (an excerpt of which is given in Table 6.2) was used here as the set used for training the classification tree in §6.1. The data set contains the average daily call count per subscriber, as well as the maximum daily call duration and call charge, the total daily call duration and call charge, and the daily call duration and call charge standard deviation, as listed in Table 6.1.

The feed-forward neural network used here to predict fraudulent and legitimate behaviour (see Appendix A.2 for a description of the computer programs), consists of $M = 7$ input units, one input unit for each explanatory input variable, $H = 3$ hidden units, and $O = 1$ binary output unit. The artificial neural network was trained using the daily subscriber statistics, as given in Table 6.2. Optimised prediction was achieved during training with a neural network of 3 hidden units, and 5 000 repeated trials, using the standard backpropagation algorithm as defined in §2.4.2. A sigmoidal activation function was used, as given in (2.16). The resultant neural network is shown in Figure 6.1. Applying the set of 1 085 observations to the neural network resulted in 3 misclassifications. These 3 mis-

		$\hat{y}_i$		
		0	1	Errors
y_i	0	999	1	1
	1	2	83	2
	Errors	2	1	

Table 6.5: *Artificial neural network confusion matrix indicating* 2 *false negatives and* 1 *false positive.*

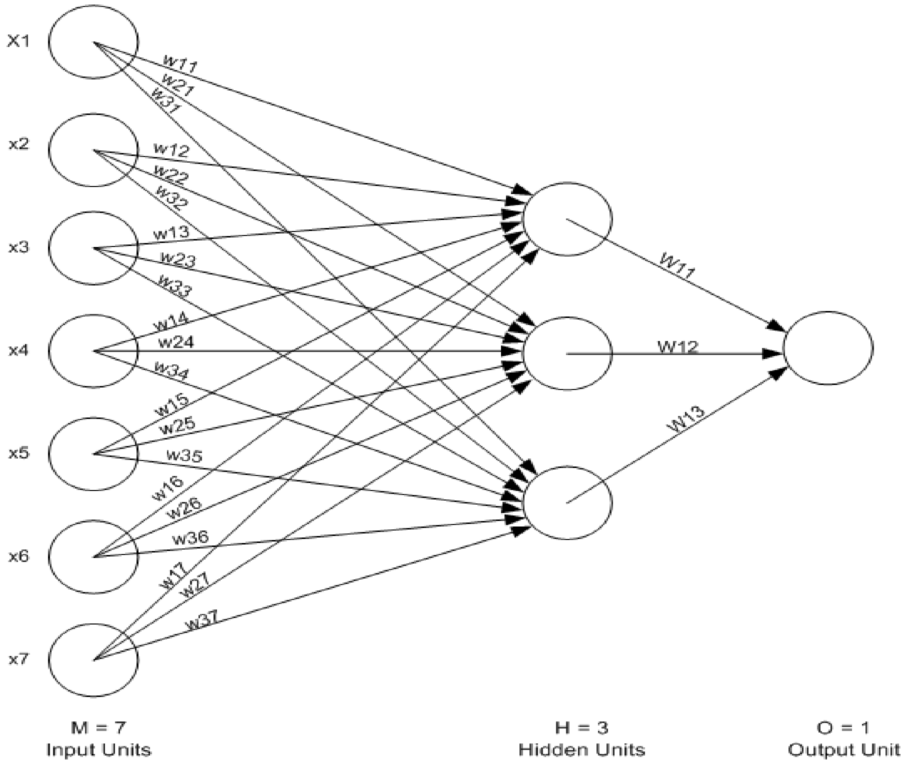

Figure 6.1: *Feed-forward neural network with $M = 7$ input units, $H = 3$ hidden units, and $O = 1$ binary output unit. The weights from input node j to hidden node k, w_{kj}, were found to be:* $w_{11} = 7.415\,59,\ w_{12} = -0.588\,319,\ w_{13} = 14.141\,9,\ w_{14} = -3.549\,39,\ w_{15} = -0.270\,243,\ w_{16} = 0.564\,485,\ w_{17} = -0.489\,127,\ w_{21} = -11.346\,9,\ w_{22} = 0.139\,34,\ w_{23} = -24.132\,7,\ w_{24} = 7.050\,1,\ w_{25} = 1.315\,41,\ w_{26} = 1.052\,06,\ w_{27} = 1.181\,82\ w_{31} = 12.616\,5,\ w_{32} = 0.429\,472,\ w_{33} = 27.266\,2,\ w_{34} = -7.677\,85,\ w_{35} = -1.721\,47,\ w_{36} = -1.286\,99,$ *and,* $w_{37} = -0.714\,687$. *The weights from hidden node k to output node q, W_{qk} are as follows:* $W_{11} = -13.527\,8,\ W_{12} = 24.287\,2,$ *and,* $W_{13} = -25.673\,9$.

classifications may be categorised into 2 false negatives and 1 false positive, as indicated by the confusion matrix in Table 6.5. The false positive caused by MSISDN 27774435190 in Table 6.2 is an observation with similar characteristics as those signifying dealer fraud (see Table 6.2, MSISDN 27779990001 to 27779990009). The opposite may be said of the two false negatives, shown in Table 6.2 as MSISDN 27774435191 and 27774435192. They show characteristics similar to legitimate observations. This again confirms that fraudulent behaviour on one account may be quite legitimate for another.

The ability of this method to detect fraud may be demonstrated by applying the artificial neural network to the unseen set of daily statistics on a subscriber's account exibiting characteristics similar to those found in dealer fraud cases, given in Table 6.6. The artificial neural network classified this subscriber's behaviour as potential fraud.

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
1	4	4	0	0.24	0.24	0

Table 6.6: *Daily statistics on the account of fraudulent subscriber 27895500053 with call count (x_1) 1, total call duration (x_2) 4s, maximum call duration (x_3) 4s, call duration standard deviation (x_4) 0s, total call charge (x_5) R0.24, maximum call charge (x_6) R0.24 and call charge standard deviation (x_7) R0.*

6.3 Bayesian Decision Making

Another classification method, applying supervised learning, is the naive Bayesian classifier. This method uses a probabilistic approach when classifying observations into fraudulent and legitimate behaviour, as described in §2.5.

The same set of data (an excerpt of which is given in Table 6.2) than was used for training the artificial neural network in §6.2, and the decision tree in §6.1 was used again for the derivation of the naive Bayesian classifier. The data set consists of the avarage daily call count per subscriber, as well as the maximum daily call duration and call charge, the total daily call duration and call charge, and the daily call duration and call charge standard deviation, as listed in Table 6.1. The Bayesian decision making method used in this section is a naive Bayesian classification, as described in §2.5.1, with measurement vector $\mathbf{X}_i$, and response variables y_i. The response variables y_i of the observations fall into one of two classes, C_1 or C_2, belonging to $\mathcal{C}$, indicating call behaviour as fraudulent or legitimate. The naive Bayesian classifier assigns class membership in $\mathcal{C}$ to each measurement vector $\mathbf{X}_i$ in $\mathcal{X}$. The learning sample $\mathcal{L}$, consisting of $N = 1\,085$ observations, was used to form the classifier (see Appendix A.3 for a description of the computer programs). The class prior probabilities, $P[C_i]$, and the conditional probabilities of the explanatory variables, $P[x_{1i}|C_i], P[x_{2i}|C_i], \ldots, P[x_{7i}|C_i]$, were estimated from the learning sample $\mathcal{L}$, and the results may be found in Table 6.7. The prior probabilities $P[C_i]$ were estimated as $P[C_i] = \frac{N_i}{N}$, where N_i is the number of observations in $\mathcal{L}$ belonging to class C_i, and N is the total number of observations in $\mathcal{L}$. The conditional probabilities of the explanatory variables are given as normal probability distributions, $N(\mu, \sigma^2)$, where μ is the expected value and σ^2 is the variance.

Applying the naive Bayesian classifier to the learning sample $\mathcal{L}$ resulted in 96 misclassifications, consisting of 15 false positives and 81 false negatives. The confusion matrix provided in Table 6.8, comparing the observed response variables y_i with the response variables generated by the naive Bayesian classifier $\hat{y}_i$, may be used as an indication of the quality of the naive Bayesian classifier. From the confusion matrix it is clear that the naive Bayesian classifier does not perform well with the particular set of observations

Classifier Probabilities	C_1	C_2
$P[C_i]$	$0.921\,7$	$0.078\,3$
$P[\mathbf{x}_1 \vert C_i]$	$N(7.005\,5, 108.173)$	$N(1.567\,3, 9.134\,9)$
$P[\mathbf{x}_2 \vert C_i]$	$N(449.858, 306\,372)$	$N(1\,367.49, 0.000\,000\,6)$
$P[\mathbf{x}_3 \vert C_i]$	$N(168.062, 18\,109.9)$	$N(131.124, 403\,741)$
$P[\mathbf{x}_4 \vert C_i]$	$N(46.920\,3, 1\,709.01)$	$N(25.8\,966, 25\,113.8)$
$P[\mathbf{x}_5 \vert C_i]$	$N(11.631\,9, 136.553)$	$N(255.637, 0.000\,003)$
$P[\mathbf{x}_6 \vert C_i]$	$N(4.356\,6, 8.307\,9)$	$N(36.726, 38\,972.2)$
$P[\mathbf{x}_7 \vert C_i]$	$N(1.139\,3, 0.734\,4)$	$N(8.8\,708, 3\,452.77)$

Table 6.7: *Probabilities obtained by the naive Bayesian classifier.*

		$\hat{y}_i$		
		0	1	Errors
y_i	0	985	15	15
	1	81	4	81
	Errors	81	15	

Table 6.8: *Naive Bayesian classifier confusion matrix indicating* 15 *false positives and* 81 *false negatives.*

used here.

The probability distributions (see Table 6.7) derived from the learning sample $\mathcal{L}$ shows the difference in fraudulent and legitimate subscriber behaviour, especially when looking more closely at explanatory variabels $\mathbf{x}_2$ and $\mathbf{x}_5$, indicating the subscriber's average daily call duration and charge, respectively. It may be seen that the mean values of both $\mathbf{x}_2$ and $\mathbf{x}_5$ are much larger for fraudulent subscribers, indicating that fraudulent subscribers spend more time making calls than legitimate subscribers, and as a result of this, are charged more.

The ability of this method to detect fraud may be demonstrated by applying the naive Bayesian classifier to the unseen set of daily statistics on a fraudulent subscriber's account, given in Table 6.9. The naive Bayesian classifier classified this subscriber's behaviour as potential fraud.

6.4 Cluster Analysis

Cluster analysis is conceptually simple, but computationally very expensive to perform. Therefore it was decided to reduce the number of observations described in §5.2. Observations, consisting of mobile originating call data records, were reduced by taking a

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
26	55282	3590	953.23	3847.53	403.92	67.823

Table 6.9: *Daily statistics on the account of fraudulent subscriber 27895500921 with call count (x_1) 26, total call duration (x_2) 55 282s, maximum call duration (x_3) 3 590s, call duration standard deviation (x_4) 953.23s, total call charge (x_5) R3 847.53, maximum call charge (x_6) R403 and call charge standard deviation (x_7) R67.823.*

random 10% sample on the 1 000 available subscribers, resulting in a data set containing 90 distinct subscribers, with a total of 82 072 observations. Cluster analysis was performed on a training sample of this sample, consisting of a random 10% sample of the 82 072 observations, in turn resulting in 8 024 observations.

Agglomerative hierarchical clustering was implemented by coding the methods described in §2.6.2) in *C* (see Appendix A.4 for a listing of the computer code). Gower's general similarity coefficient was used to compute the initial similarity matrix, $\mathbf{D} = \{d_{ik}\}$. McQuitty's similarity analysis (see §2.6.1) was then employed to merge the two most similar observations, and to compute the similarity between the newly formed cluster and the remaining clusters, updating the similarity matrix with this value. Cluster analysis was finally performed on the unnormalised explanatory variables $\mathbf{X}_i = (x_{1i}, x_{2i}, \ldots, x_{7i})$, with Gower's general similarity coefficient normalising the variables to within the range $[0.0, 1.0]$. The clustering process started with 8 024 initial clusters, each one containing a single observation. During each iteration of the clustering process the two most similar clusters were merged into one cluster, continuing with this process until the distance between the most similar clusters was found to be less than 0.8. The clustering process terminated after 8 013 iterations, with 11 clusters remaining, namely $C = (C_1, \ldots, C_{11})$. The observations were updated with the identifier of the cluster to which they belong. Table 6.10 is a representation of the resulting clusters, indicating the number of observations contained in each final cluster, and the similarity measure between the clusters merged to form that final cluster.

Each subscriber, m_k, $(k = 1, \ldots, 90)$, was subsequently assigned a profile, expressed as a series of 11 probabilities, $\mathbf{P}_{m_k} = (P[C_1|m_k], \ldots, P[C_{11}|m_k])$, indicating the probability of each cluster containing observations belonging to subscriber m_k. The conditional probability $P[C_j|m_k]$ was estimated by

$$P[C_j|m_k] = \frac{N_{C_j}}{N_{m_k}},$$

where N_{m_k} is the number of observations made on subscriber m_k, and N_{C_j} is the number of observations clustered into cluster C_j, belonging to m_k. The subscriber profiles calculated as part of the clustering process may be found in Table 6.12. Classification was performed,

	Number of	**Maximum**
Cluster	**observations**	**Similarity**
01	2	0.920 766
02	2	0.969 116
03	1 740	0.900 460
04	592	0.898 096
05	883	0.897 611
06	858	0.888 704
07	874	0.888 395
08	2	0.856 904
09	1	1
10	1	1
11	3 069	0.945 159

Table 6.10: *Results of the clustering procedure, indicating the number of observations grouped into each cluster and the similarity measure between the final two clusters merged to form this cluster.*

using the cluster identifier as response variable, and $\mathbf{X}_i = (x_{1i}, x_{2i}, \ldots, x_{7i})$ as explanatory variables, to derive eight rules, $d(\mathbf{X}_i)$, $i = 1, 2, \ldots 8$, predicting the cluster identifier. The classification tree algorithm in the computer program *Statistica* was employed to perform this part of the analysis, using the chi-square measure to determine goodness of fit, and estimating the prior probabilities. The resuling classification tree is shown in Figure 6.2. The following eight rules were extracted from the classification tree:

- IF $x_5 > 5\,711.5$ THEN *cluster08*,
- IF $x_5 \leq 5\,771.5$ AND $x_7 > 1\,710.5$ THEN *cluster02*,
- IF $x_5 \leq 5\,771.5$ AND $x_7 \leq 1\,710.5$ AND $x_3 = 1$ AND $x_2 \leq 68$ THEN *cluster07*,
- IF $x_5 \leq 5\,771.5$ AND $x_7 \leq 1\,710.5$ AND $x_3 = 1$ AND $x_2 > 68$ AND $x_4 > 623$ THEN *cluster05*,
- IF $x_5 \leq 5\,771.5$ AND $x_7 \leq 1\,710.5$ AND $x_3 = 1$ AND $x_2 > 68$ AND $x_4 \leq 623$ THEN *cluster11*,
- IF $x_5 \leq 5\,771.5$ AND $x_7 \leq 1\,710.5$ AND $x_3 = 0$ AND $x_2 \leq 68$ THEN *cluster06*,
- IF $x_5 \leq 5\,771.5$ AND $x_7 \leq 1\,710.5$ AND $x_3 = 0$ AND $x_2 > 68$ AND $x_4 > 623$ THEN *cluster04*, AND
- IF $x_5 \leq 5\,771.5$ AND $x_7 \leq 1\,710.5$ AND $x_3 = 0$ AND $x_2 > 68$ AND $x_4 \leq 623$ THEN *cluster03*.

To estimate the the accuracy of $d(\mathbf{X}_i)$, $i = 1, 2, \ldots 8$, a second independent random 10% test sample was taken, consisting of 8 161 observations from the same population as the one from which the training sample was extracted. The test sample was clustered using the rules $d(\mathbf{X}_i)$, $i = 1, 2, \ldots 8$, and the results compared to the clustered training sample, resulting in 18, out of a total of 8 161 observations, clustered into clusters not featuring in the particular subscriber's profile.

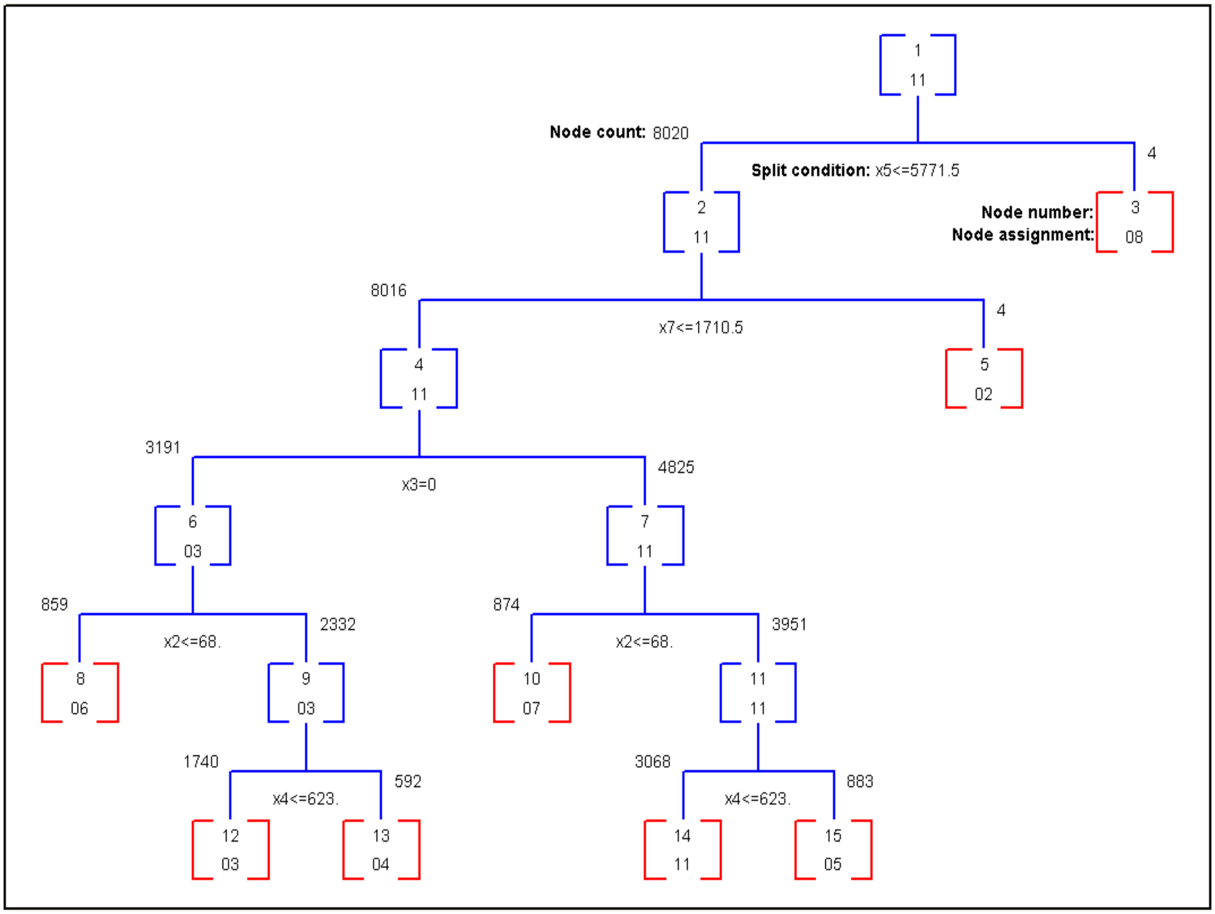

Figure 6.2: *The classification tree applied to the clustered observations, deriving rules to predict cluster membership. This classification tree terminated with 8 terminal nodes, each terminal node assiged to one of the observed clusters.*

The classification tree given in Figure 6.2 divides subscribers into a number of behaviour groups. The most significant variable in the classification tree is x_5, call distance, dividing subscribers into those making long distance international calls and those making national calls and international calls to other African countries. *Cluster08* contains international calls with a distance of more than 5 771.5 km, with 4 observations fitting that profile. The next variable dividing subscribers is x_7, call duration. *Cluster02* contains 4 observations of calls made to destinations less than or equal to 5 771.5 km from South Africa (Johannesburg), lasting more than 1 710.5 seconds. Variable x_3 divides subscribers further into those making calls during peak and off-peak rating periods. *Cluster06*, containing 859 observations, and *cluster07*, containing 874 observations, describe similar behaviour, except that observations in *cluster06* represent calls during off-peak rating periods and *cluster07* during peak rating periods. Variable x_2, namely subscriber tariff, divides subscribers into prepaid and postpaid customers, where the subscription tariff paid by prepaid subscriber is less than or equal to R68. Observations contained in *cluster06* may be described as calls being made by prepaid subscribers during off-peak rating periods, with duration less than or equal to 1 710.5 seconds to destinations not further than 5 771.5 km from South Africa. *Cluster03* contains 1 740 observations of calls made by postpaid subscribers in the proximity of cell sites less than or equal to 623 km from Johannesburg during off-peak rating periods, with duration less than or equal to 1 710.5 seconds and to a destination less than or equal to 5 771.5 km from South Africa. *Cluster04* describes similar behaviour as *cluster03* with the exception that observations in this cluster represent calls in the proximity of cell sites more than 623 km from Johannesburg. The only difference between *cluster11* and *cluster03* is that observations in *cluster11* represent calls during peak rating periods, while observations contained in *cluster03* represent calls during off-peak rating periods. This is also the behaviour difference between observations contained in *cluster04* and *cluster05*.

It is unlikely that any subscriber will belong to only one of these clusters, as may be seen in Table 6.12, which contains each sampled subscribers' probabilities of belonging to specific clusters. For example, the behaviour of subscriber $m_k = 27896704693$ may be described by the probability distribution

$$\mathbf{P}_{m_k} = (0.00, 0.00, 0.00, 0.00, 0.00, 0.56, 0.44, 0.00, 0.00, 0.00, 0.00),$$

indicating that the probability of this subscriber belonging to *cluster06* is 56% and belonging to *cluster07* is 44%. On further investigation of the probability distributions in Table 6.12 it was found that variable x_3 is (with most observations) the reason why subscribers are divided between *cluster06* and *cluster07* ($P[C_6|C_7] = 0.95, P[C_7|C_6] = 1.0$), *cluster03* and *cluster11* ($P[C_3|C_{11}] = 0.92, P[C_{11}|C_3] = 0.97$), and *cluster04* and *cluster05* ($P[C_4|C_5] = 0.9, P[C_5|C_4] = 1.0$). Variable x_3 may therefore be removed to simplify the set of rules classifying subscribers into different behaviour groups.

The clusters to which each subscriber belongs and the subscribers' probability distributions may be of aid in the fraud detection process in more than one way. Dividing subscribers into different behaviour groups may be useful in defining different fraud detection rules for different groups, customised to the types of subscriber contained in them. The rules used to classify subscribers into different behaviour groups may further be used to define threshold values for each group, identifying uncharacteristic behaviour for members of that group, indicating possible fraud. Most fraud management systems group subscribers according to behaviour, but based on the product the subscriber is subscribed to, assuming that the subscriber's product is an indication of his/her behaviour. The rules derived from the classification tree in Figure 6.2 may also be used to classify new observations into one of the 11 clusters. Observations clustered into clusters not typical for the subscriber, may be identified as possible fraudulent activity, and marked for further investigation.

The ability of this method to detect fraud on a subscriber's account may be demonstrated by artificially creating a fraudulent call not fitting the subscriber's behaviour profile. The behaviour of subscriber $m_k = 27893200574$ is described by the probability distribution

$$\mathbf{P}_{m_k} = (0.00, 0.00, 0.65, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.35)$$

in Table 6.12, indicating that 65% of the observations for this subscriber belong to *cluster03* and 35% to *cluster11*. The classification tree in Figure 6.2 indicates that *cluster03* and *cluster11* describe the behaviour of subscribers making calls to destinations (x_5) less than or equal to 5 771.5km from South Africa (Johannesburg), with call duration (x_7) less than or equal to 1 710.5s, paying a subscription tariff (x_2) greater than R68 and that calls are made in the proximity of cells less than or equal to 623km from Johannesburg, during both peak and off-peak (x_3) rating periods. A typical call classified as potential fraud for this subscriber's account is given in Table 6.11.

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
0	135	1	0	11541	24.26	60

Table 6.11: *Typical fraudulent call on the account of subscriber 27893200574 with prepaid indicator (x_1) 0, subscriber tariff (x_2) R135, peak indicator (x_3) 1, cell location (x_4) 0km, call distance (x_5) 11 541km, call charge (x_6) R24.26 and call duration (x_7) 60s.*

The classification tree in Figure 6.2 classified this call into *cluster08*, which is not in the behaviour profile of subscriber 27893200574.

m_k	$\mathbf{P}_{m_k} = (P[C_1\|m_k], \ldots, P[C_{11}\|m_k])$	N_{m_k}
27892546298	(0.00, 0.00, 0.00, 0.00, 0.00, 0.49, 0.51, 0.00, 0.00, 0.00, 0.00)	41
27892572436	(0.00, 0.00, 0.00, 0.00, 0.00, 0.84, 0.16, 0.00, 0.00, 0.00, 0.00)	37
27892585425	(0.00, 0.00, 0.00, 0.00, 0.00, 0.50, 0.50, 0.00, 0.00, 0.00, 0.00)	4
27892585436	(0.00, 0.00, 0.00, 0.00, 0.00, 0.33, 0.67, 0.00, 0.00, 0.00, 0.00)	6
27892585531	(0.00, 0.00, 0.00, 0.00, 0.00, 0.51, 0.49, 0.00, 0.00, 0.00, 0.00)	57
27892585584	(0.00, 0.00, 0.00, 0.00, 0.00, 0.27, 0.73, 0.00, 0.00, 0.00, 0.00)	102
27892585621	(0.00, 0.00, 0.00, 0.00, 0.00, 0.70, 0.30, 0.00, 0.00, 0.00, 0.00)	30
27892585774	(0.00, 0.00, 0.00, 0.00, 0.00, 0.62, 0.38, 0.00, 0.00, 0.00, 0.00)	26
27892585886	(0.00, 0.00, 0.00, 0.00, 0.00, 0.40, 0.60, 0.00, 0.00, 0.00, 0.00)	10
27892585944	(0.00, 0.00, 0.00, 0.00, 0.00, 0.49, 0.51, 0.00, 0.00, 0.00, 0.00)	72

Continued on next page

m_k	$\mathbf{P}_{m_k} = (P[C_1\|m_k], \ldots, P[C_{11}\|m_k])$	N_{m_k}
27892585961	(0.00, 0.00, 0.00, 0.00, 0.00, 0.18, 0.82, 0.00, 0.00, 0.00, 0.00)	17
27892586344	(0.00, 0.00, 0.00, 0.00, 0.00, 0.58, 0.42, 0.00, 0.00, 0.00, 0.00)	19
27892586351	(0.00, 0.00, 0.00, 0.00, 0.00, 0.17, 0.83, 0.00, 0.00, 0.00, 0.00)	35
27892586373	(0.00, 0.00, 0.00, 0.00, 0.00, 0.13, 0.63, 0.25, 0.00, 0.00, 0.00)	8
27896704693	(0.00, 0.00, 0.00, 0.00, 0.00, 0.56, 0.44, 0.00, 0.00, 0.00, 0.00)	160
27896714116	(0.00, 0.00, 0.00, 0.00, 0.00, 0.60, 0.40, 0.00, 0.00, 0.00, 0.00)	5
27896714643	(0.00, 0.00, 0.00, 0.00, 0.00, 0.47, 0.53, 0.00, 0.00, 0.00, 0.00)	19
27896718832	(0.00, 0.00, 0.00, 0.00, 0.00, 0.36, 0.64, 0.00, 0.00, 0.00, 0.00)	70
27896718871	(0.00, 0.00, 0.00, 0.00, 0.00, 0.80, 0.20, 0.00, 0.00, 0.00, 0.00)	54
27896724028	(0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00)	1
27896724638	(0.00, 0.00, 0.00, 0.00, 0.00, 0.31, 0.69, 0.00, 0.00, 0.00, 0.00)	26
27896733463	(0.00, 0.00, 0.00, 0.00, 0.00, 0.22, 0.78, 0.00, 0.00, 0.00, 0.00)	23
27896738904	(0.00, 0.00, 0.00, 0.00, 0.00, 0.72, 0.28, 0.00, 0.00, 0.00, 0.00)	79
27896748848	(0.00, 0.00, 0.00, 0.00, 0.00, 0.38, 0.63, 0.00, 0.00, 0.00, 0.00)	32
27896754119	(0.00, 0.00, 0.00, 0.00, 0.00, 0.14, 0.86, 0.00, 0.00, 0.00, 0.00)	36
27896758844	(0.00, 0.00, 0.00, 0.00, 0.00, 0.67, 0.33, 0.00, 0.00, 0.00, 0.00)	146
27896759059	(0.00, 0.00, 0.00, 0.00, 0.00, 0.20, 0.80, 0.00, 0.00, 0.00, 0.00)	40
27896763474	(0.00, 0.00, 0.00, 0.00, 0.00, 0.51, 0.49, 0.00, 0.00, 0.00, 0.00)	118
27896764568	(0.00, 0.00, 0.00, 0.00, 0.00, 0.41, 0.59, 0.00, 0.00, 0.00, 0.00)	17
27896764599	(0.00, 0.00, 0.00, 0.00, 0.00, 0.20, 0.80, 0.00, 0.00, 0.00, 0.00)	5
27896764616	(0.00, 0.00, 0.00, 0.00, 0.00, 0.53, 0.42, 0.00, 0.00, 0.00, 0.05)	19
27896764646	(0.00, 0.00, 0.00, 0.00, 0.00, 0.14, 0.86, 0.00, 0.00, 0.00, 0.00)	36
27896774610	(0.00, 0.00, 0.00, 0.00, 0.00, 0.33, 0.67, 0.00, 0.00, 0.00, 0.00)	6
27896774626	(0.00, 0.00, 0.00, 0.00, 0.00, 0.53, 0.47, 0.00, 0.00, 0.00, 0.00)	109
27896774647	(0.00, 0.00, 0.00, 0.00, 0.00, 0.35, 0.65, 0.00, 0.00, 0.00, 0.00)	17
27896779228	(0.00, 0.00, 0.00, 0.00, 0.00, 0.10, 0.90, 0.00, 0.00, 0.00, 0.00)	50
27896784113	(0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00)	7
27896788838	(0.00, 0.00, 0.00, 0.00, 0.00, 0.91, 0.09, 0.00, 0.00, 0.00, 0.00)	82
27896788883	(0.00, 0.00, 0.00, 0.00, 0.00, 0.60, 0.40, 0.00, 0.00, 0.00, 0.00)	58
27896788904	(0.00, 0.00, 0.00, 0.00, 0.00, 0.39, 0.61, 0.00, 0.00, 0.00, 0.00)	51
27896789050	(0.00, 0.00, 0.00, 0.00, 0.00, 0.40, 0.60, 0.00, 0.00, 0.00, 0.00)	5
27893200009	(0.00, 0.00, 0.01, 0.25, 0.72, 0.00, 0.00, 0.00, 0.00, 0.00, 0.02)	97
27893200076	(0.00, 0.00, 0.00, 0.63, 0.35, 0.00, 0.00, 0.00, 0.00, 0.00, 0.01)	68
27893200110	(0.00, 0.00, 0.56, 0.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.43)	70
27893200114	(0.00, 0.00, 0.16, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.84)	139
27893200117	(0.00, 0.00, 0.02, 0.53, 0.44, 0.00, 0.00, 0.00, 0.00, 0.00, 0.02)	55
27893200143	(0.00, 0.00, 0.39, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.61)	57
27893200148	(0.00, 0.00, 0.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.40)	20
27893200216	(0.00, 0.00, 0.35, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.65)	533
27893200227	(0.00, 0.00, 0.25, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.75)	129
27893200244	(0.00, 0.00, 0.38, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.62)	259
27893200254	(0.00, 0.00, 0.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.40)	20
27893200270	(0.00, 0.00, 0.00, 0.28, 0.71, 0.00, 0.00, 0.00, 0.00, 0.00, 0.01)	138

Continued on next page

m_k	$\mathbf{P}_{m_k} = (P[C_1\|m_k], \ldots, P[C_{11}\|m_k])$	N_{m_k}
27893200280	(0.00, 0.00, 0.41, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.59)	502
27893200327	(0.00, 0.00, 0.65, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.35)	72
27893200331	(0.00, 0.00, 0.85, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.15)	34
27893200338	(0.00, 0.00, 0.33, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.67)	288
27893200364	(0.00, 0.00, 0.26, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.74)	74
27893200391	(0.00, 0.00, 0.00, 0.30, 0.70, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	326
27893200395	(0.00, 0.00, 0.00, 0.40, 0.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	391
27893200495	(0.00, 0.00, 0.38, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.63)	544
27893200506	(0.00, 0.00, 0.46, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.54)	50
27893200519	(0.00, 0.00, 0.94, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.06)	52
27893200526	(0.00, 0.00, 0.00, 0.10, 0.86, 0.00, 0.00, 0.00, 0.05, 0.00, 0.00)	21
27893200549	(0.00, 0.00, 0.00, 0.73, 0.27, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	26
27893200562	(0.02, 0.00, 0.17, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.81)	101
27893200574	(0.00, 0.00, 0.65, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.35)	110
27893200576	(0.00, 0.00, 0.25, 0.00, 0.03, 0.00, 0.00, 0.00, 0.00, 0.00, 0.72)	60
27893200599	(0.00, 0.00, 0.65, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.35)	46
27893200621	(0.00, 0.00, 0.41, 0.56, 0.03, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	32
27893200660	(0.00, 0.00, 0.32, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.68)	198
27893200665	(0.00, 0.00, 0.37, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.63)	126
27893200671	(0.00, 0.00, 0.00, 0.52, 0.48, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	33
27893200674	(0.00, 0.00, 0.01, 0.52, 0.43, 0.00, 0.00, 0.00, 0.00, 0.00, 0.04)	83
27893200680	(0.00, 0.00, 0.00, 0.88, 0.13, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	8
27893200706	(0.00, 0.00, 0.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.79)	28
27893200711	(0.00, 0.00, 0.00, 0.88, 0.13, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	16
27893200767	(0.00, 0.00, 0.48, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.52)	50
27893200811	(0.00, 0.00, 0.28, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.72)	90
27893200907	(0.00, 0.00, 0.52, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.48)	33
27893200929	(0.00, 0.00, 0.00, 0.94, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	34
27893200993	(0.00, 0.00, 0.38, 0.08, 0.10, 0.00, 0.00, 0.00, 0.00, 0.00, 0.44)	39
27893201006	(0.00, 0.05, 0.00, 0.60, 0.35, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	43
27893201010	(0.00, 0.00, 0.44, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.56)	52
27893201054	(0.00, 0.00, 0.83, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.17)	41
27893201068	(0.00, 0.00, 0.32, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.68)	254
27893201082	(0.00, 0.00, 0.35, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.65)	46
27893201098	(0.00, 0.00, 0.19, 0.01, 0.03, 0.00, 0.00, 0.00, 0.00, 0.00, 0.77)	648
27893201162	(0.00, 0.00, 0.33, 0.03, 0.07, 0.00, 0.00, 0.00, 0.00, 0.00, 0.57)	67
27893201164	(0.00, 0.00, 0.00, 0.16, 0.84, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)	86

Table 6.12: Probability profiles indicating the probability distribution of each subscriber's call data observations between the different behaviour groups or clusters.

6.5 Outlier Analysis

In this section outlier analysis is applied (see Appendix A.5 for a listing of the computer code) to the population of legitimate call data records defined in §5.2. The observations in the population are normalised, using min–max normalisation, as described in (5.1), scaling the variables to within the range $[0.0, 1.0]$.

The Mahalanobis distance, denoted by D^2, is a measure of distance between two points in the space defined by two or more correlated variables, and is used to identify outliers amongst the multivariate observations, as described in §2.7. This distance measure is computed as

$$D^2(\mathbf{X}, \overline{\mathbf{X}}) = (\mathbf{X} - \overline{\mathbf{X}})'\Sigma^{-1}(\mathbf{X} - \overline{\mathbf{X}}), \tag{6.1}$$

where Σ is the covariance matrix for the explanatory variables, $\mathbf{X}$ is the vector of explanatory variables for all observations in the population, and $\overline{\mathbf{X}}$ is the vector of corresponding means, taken over all observations in the population. The covariance matrix for explanatory variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_7$ is given by

$$\begin{bmatrix}
Var(\mathbf{x}_1) & Cov(\mathbf{x}_1,\mathbf{x}_2) & Cov(\mathbf{x}_1,\mathbf{x}_3) & Cov(\mathbf{x}_1,\mathbf{x}_4) & Cov(\mathbf{x}_1,\mathbf{x}_5) & Cov(\mathbf{x}_1,\mathbf{x}_6) & Cov(\mathbf{x}_1,\mathbf{x}_7) \\
Cov(\mathbf{x}_2,\mathbf{x}_1) & Var(\mathbf{x}_2) & Cov(\mathbf{x}_2,\mathbf{x}_3) & Cov(\mathbf{x}_2,\mathbf{x}_4) & Cov(\mathbf{x}_2,\mathbf{x}_5) & Cov(\mathbf{x}_2,\mathbf{x}_6) & Cov(\mathbf{x}_2,\mathbf{x}_7) \\
Cov(\mathbf{x}_3,\mathbf{x}_1) & Cov(\mathbf{x}_3,\mathbf{x}_2) & Var(\mathbf{x}_3) & Cov(\mathbf{x}_3,\mathbf{x}_4) & Cov(\mathbf{x}_3,\mathbf{x}_5) & Cov(\mathbf{x}_3,\mathbf{x}_6) & Cov(\mathbf{x}_3,\mathbf{x}_7) \\
Cov(\mathbf{x}_4,\mathbf{x}_1) & Cov(\mathbf{x}_4,\mathbf{x}_2) & Cov(\mathbf{x}_4,\mathbf{x}_3) & Var(\mathbf{x}_4) & Cov(\mathbf{x}_4,\mathbf{x}_5) & Cov(\mathbf{x}_4,\mathbf{x}_6) & Cov(\mathbf{x}_4,\mathbf{x}_7) \\
Cov(\mathbf{x}_5,\mathbf{x}_1) & Cov(\mathbf{x}_5,\mathbf{x}_2) & Cov(\mathbf{x}_5,\mathbf{x}_3) & Cov(\mathbf{x}_5,\mathbf{x}_4) & Var(\mathbf{x}_5) & Cov(\mathbf{x}_5,\mathbf{x}_6) & Cov(\mathbf{x}_5,\mathbf{x}_7) \\
Cov(\mathbf{x}_6,\mathbf{x}_1) & Cov(\mathbf{x}_6,\mathbf{x}_2) & Cov(\mathbf{x}_6,\mathbf{x}_3) & Cov(\mathbf{x}_6,\mathbf{x}_4) & Cov(\mathbf{x}_6,\mathbf{x}_5) & Var(\mathbf{x}_6) & Cov(\mathbf{x}_6,\mathbf{x}_7) \\
Cov(\mathbf{x}_7,\mathbf{x}_1) & Cov(\mathbf{x}_7,\mathbf{x}_2) & Cov(\mathbf{x}_7,\mathbf{x}_3) & Cov(\mathbf{x}_7,\mathbf{x}_4) & Cov(\mathbf{x}_7,\mathbf{x}_5) & Cov(\mathbf{x}_7,\mathbf{x}_6) & Var(\mathbf{x}_7)
\end{bmatrix},$$

where $Var(x_i) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$ and $Cov(x_i, y_i) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})$. The Mahalanobis distance, applied to the population of call data observations, ranges between 1.37252 and 8.53905. For this data set Σ is given by

$$\Sigma = \begin{bmatrix}
0.1877000 & -110.69643 & -0.0283600 & -34.448540 & 0.1307730 & 0.0321130 & -3.3819600 \\
-110.69643 & 134209.36 & 32.944430 & 6393.8992 & -442.68969 & -166.72483 & -2174.2793 \\
-0.0283600 & 32.944443 & 0.2391700 & 11.059560 & -0.3183500 & 0.1246200 & -3.4062900 \\
-34.448540 & 6393.8992 & 11.059560 & 164345.64 & -242.54293 & 56.346180 & 2612.8225 \\
0.1307700 & -442.68969 & -0.3183600 & -242.54293 & 41858.556 & 80.932910 & 582.86200 \\
0.0321100 & -166.72483 & 0.1246300 & 56.346180 & 80.932910 & 9.0715300 & 298.06640 \\
-3.3819600 & -2174.2793 & -3.406290 & 2612.8224 & 582.86200 & 298.06640 & 15128.952
\end{bmatrix}$$

and hence Σ^{-1} may be computed as

$$\Sigma^{-1} = \begin{bmatrix}
11.142434 & 0.009157 & 0.015224 & 0.001923 & 0.000014 & 0.006562 & 0.003348 \\
0.009157 & 0.000015 & -0.001431 & 0.000001 & -0.000001 & 0.000424 & -0.000006 \\
0.015224 & -0.001431 & 4.685169 & -0.000259 & 0.000586 & -0.349019 & 0.007751 \\
0.001923 & 0.000001 & -0.000259 & 0.000006 & 0.000000 & -0.000003 & -0.000000 \\
0.000014 & -0.000001 & 0.000586 & 0.000000 & 0.000025 & -0.000600 & 0.000011 \\
0.006562 & 0.000424 & -0.349019 & -0.000003 & -0.000600 & 0.362768 & -0.007140 \\
0.003348 & -0.000005 & 0.007751 & -0.000000 & 0.000011 & -0.007140 & 0.000208
\end{bmatrix}.$$

It is clear, from the 11 observations with the largest Mahalanobis distance measures listed in Table 6.13 (identified in Table 6.13 by column heading D^2), that the explanatory variables $\mathbf{x}_5$ (Call Distance), $\mathbf{x}_6$ (Call Charge) and $\mathbf{x}_7$ (Call Duration) contribute significantly to the outlier status in the Mahalanobis sense. Explanatory variable $\mathbf{x}_4$ (Cell Location) also contributed to the Mahalanobis distance, but not as significantly as the others. Figure 6.3 places the 11 observations with the largest Mahalanobis distance measures (identified by the symbol ■) in context with the remaining observations (identified by the ●), and confirms that explanatory variables $\mathbf{x}_5$ (Call Distance), $\mathbf{x}_6$ (Call Charge) and $\mathbf{x}_7$ (Call Duration) contribute significantly to the Mahalanobis distance measure.

$\mathbf{D}^2$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
8.53905122	1	1	0	0	8669	332.28	923
8.18852204	1	1	0	0	8045	169.2	470
8.07769403	1	1	0	0	9079	102.24	426
8.05717768	1	1	0	1265	0	88.35	3365
8.05532172	1	1	0	159	0	86.34	1439
8.04577755	1	1	0	159	0	79.44	1324
8.04259353	1	1	0	1265	6308	75	290
8.04204735	1	1	0	1265	8363	75	292
8.04157582	1	1	0	300	11526	77.04	321
8.02879972	1	1	0	442	2464	65.52	182
8.02631010	1	1	0	0	980	65	780

Table 6.13: *The explanatory variables $\boldsymbol{x}_1$ (Prepaid Indicator), $\boldsymbol{x}_2$ (Subscriber Tariff in Rands), $\boldsymbol{x}_3$ (Peak Indicator), $\boldsymbol{x}_4$ (Cell Location in kilometers), $\boldsymbol{x}_5$ (Call Distance in kilometers), $\boldsymbol{x}_6$ (Call Charge in Rands) and $\boldsymbol{x}_7$ (Call Duration in seconds), of the* 11 *legitimate observations with the largest Mahalanobis distance measures.*

Since the population of observations corresponds to normal behaviour, the maximum Mahalanobis distance may be used to set a threshold value. The Mahalanobis distance between new observations and the population mean may be compared to the threshold value, and identified as an outlier when exceeding this value. Outliers may indicate fraudulent behaviour, as seen in Figure 6.4, where the symbols ○ and ■ indicate legitimate observations with Mahalanobis distance less than 8.026 310 1 and greater than 8.026 31, respectively. The symbol ▲ is used in Figure 6.4 to indicate fraudulent observations.

Implementing the Mahalanobis distance 8.53905 (the largest Mahalanobis distance measure on the legitimate observations) as threshold value resulted in 10 observations being identified as outliers. All of these outliers happen to be fraudulent observations, but the remaining 165 fraudulent observations were not identified as outliers. This is a well known phenomenon, indicating that fraudulent behaviour on one account may be legitimate for another.

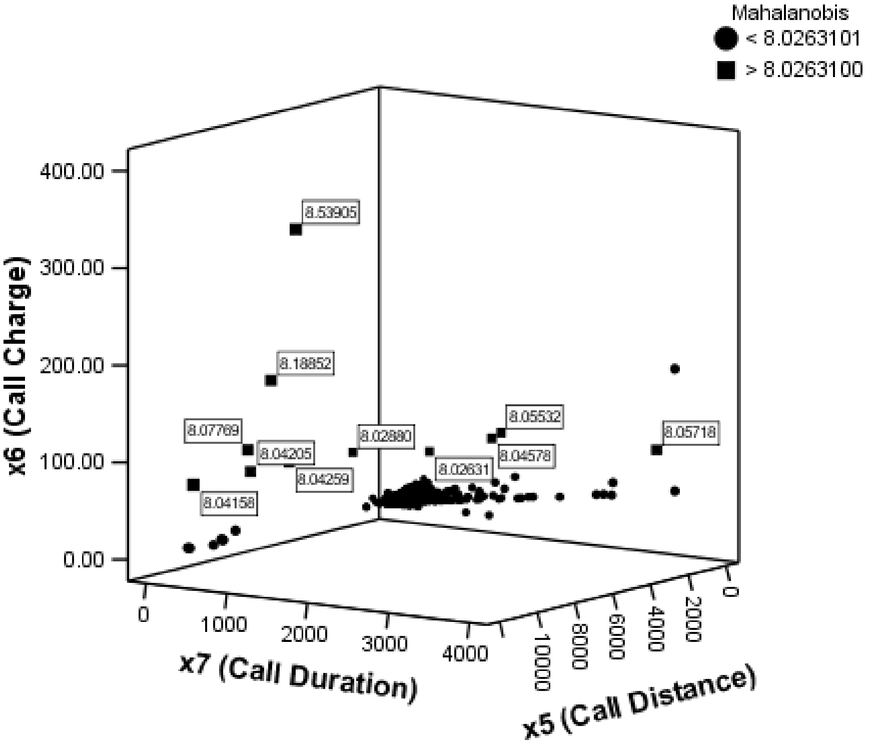

Figure 6.3: *Legitimate observations with the largest Mahalanobis distance measures in context with the remaining observations. Variable x_5 is measured in kilometers, x_6 in Rands and x_7 in seconds.*

The ability of this method to detect fraud may be demonstrated by calculating the Mahalanobis distance between the unseen fraudulent call data record, given in Table 6.14, and the population mean. The calculated Mahalanobis distance of 8.69021 exceeds the outlier threshold value of 8.53905, identifying this call data record as potential fraud.

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
1	1	0	0	8045	1251	3120

Table 6.14: *Typical fraudulent call on the account of subscriber 27555588856 with prepaid indicator (x_1) 1, subscriber tariff (x_2) R1, peak indicator (x_3) 0, cell location (x_4) 0km, call distance (x_5) 8 045km, call charge (x_6) R1 251 and call duration (x_7) 3 120s.*

6.6 Association Rule Mining

Association rule mining is a powerful method for so-called market basket analysis, which aims at detecting regularities in the behaviour of customers of supermarkets, telecommu-

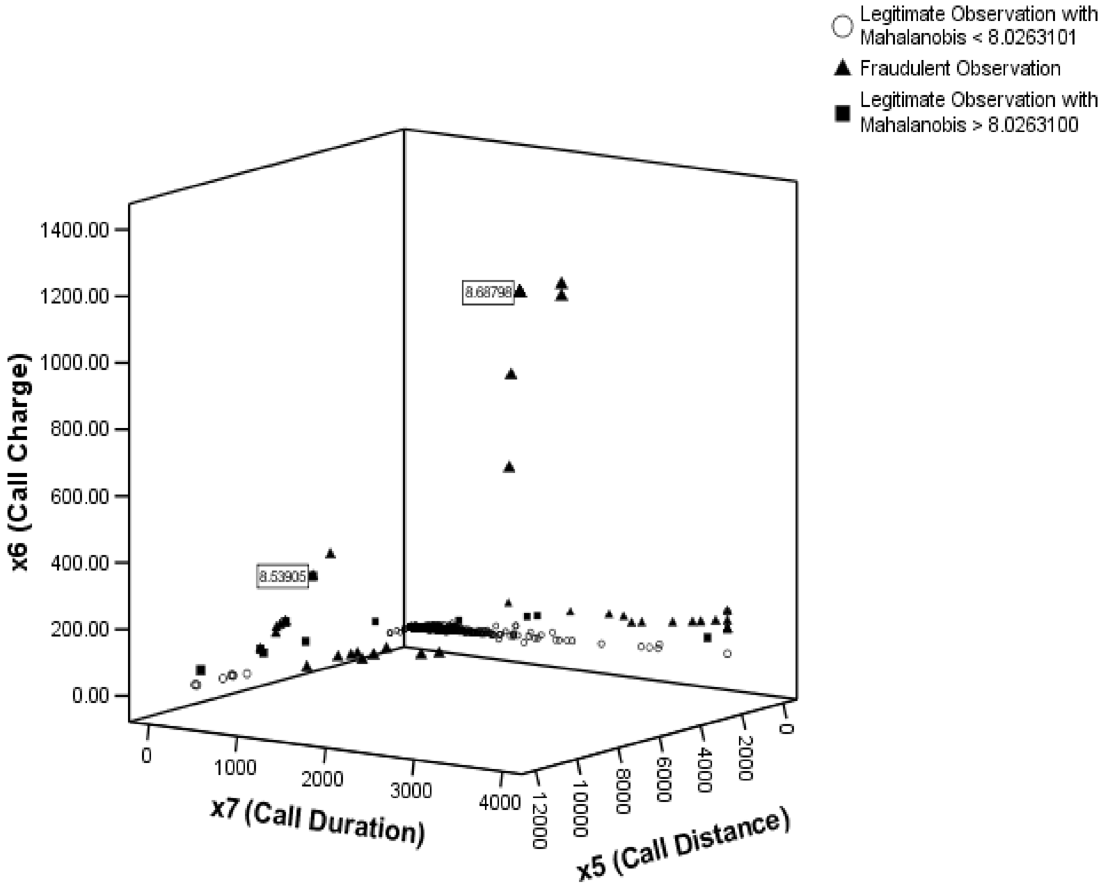

Figure 6.4: *Fraudulent observations in context with the legitimate observations given in Figure 6.3. Variable x_5 is measured in kilometers, x_6 in Rands and x_7 in seconds.*

nication companies and the like, as explained in §2.8. With the induction of association rules one tries to find sets of products or services that are frequently bought together, so that, from the presence of certain products, one may infer (with a high probability) that certain other products are also present [6]. In this section the Apriori algorithm (see §2.8.2) is used for mining frequent item sets (see Appendix A.6 for a description of the computer programs) in each subscriber's set of call data records, in an attempt to derive a unique fingerprint (signature) for each subscriber, describing the subscriber's behaviour on the network. Comparing a fingerprint of a suspected subscriber with the fingerprints of proved fraudsters may help identify fraudsters who are using a new identity.

The original form of the data set provided by the network operator was used for association rule mining. To reduce the complexity of the data and the number of observations, only mobile originating call data records were used in this process. Continuous attributes were categorised to aid in the process of finding frequent item sets. The attributes *call_charge* and *call_duration* were binned, reducing the number of distinct values that these attributes may attain, whilst retaining their original distribution. The attribute *call_charge* was binned into 15 bins and each bin's boundaries were selected to retain the attribute's original distribution. Histograms were used and to test different bin boundaries and numbers of bins against the attribute's original distribution. The bin boundaries chosen for this attribute are shown in Table 6.15, and the corresponding histograms are shown in Figure 6.5. The same method was used to bin the attribute *call_duration* into 10 bins. The bin boundaries chosen for this attribute are shown in Table 6.16 and the corresponding histograms are shown in Figure 6.6. The attribute *call_date*, indicating the date and time at which each call was made, was replaced by two attributes derived from it, named *call_day* and *call_hour*, indicating the day of the week and the hour of the day during which the call was made. An attribute *other_party_number*, indicating the number dialled, was also included in the set of data to be used for association rule mining, as well as an attribute derived from *other_party_number*, named *call_destination*, indicating the country of call destination. The attributes *location_area_code* and *cell_id*, indicating the location of the subscriber when a call was made, was also included in the data set.

	Bin Boundary	**Observation**
Bin	**(Rands)**	**Count**
01	$[0; 0.3]$	247 347
02	$(0.3; 1]$	239 278
03	$(1; 2]$	196 154
04	$(2; 3]$	108 550
05	$(3; 4]$	39 606
06	$(4; 5]$	19 879
07	$(5; 6]$	17 014
08	$(6; 8]$	14 778
09	$(8; 10]$	9 174
10	$(10; 12]$	4 926
11	$(12; 15]$	4 036
12	$(15; 20]$	3 277
13	$(20; 50]$	3 261
14	$(50; 100]$	246
15	$(100; \infty)$	26

Table 6.15: *Call_charge bin boundaries, measured in Rands.*

Each subscriber's set of call data records were separated from the remaining observations in the population. The Apriori algorithm was applied to the subscriber's set of call data records, with minimum support threshold set to 10%, and the minimum number of items allowed per item set to 1. Association rules were not generated from the frequent item sets, but each frequent item set was instead associated with the subscriber to which

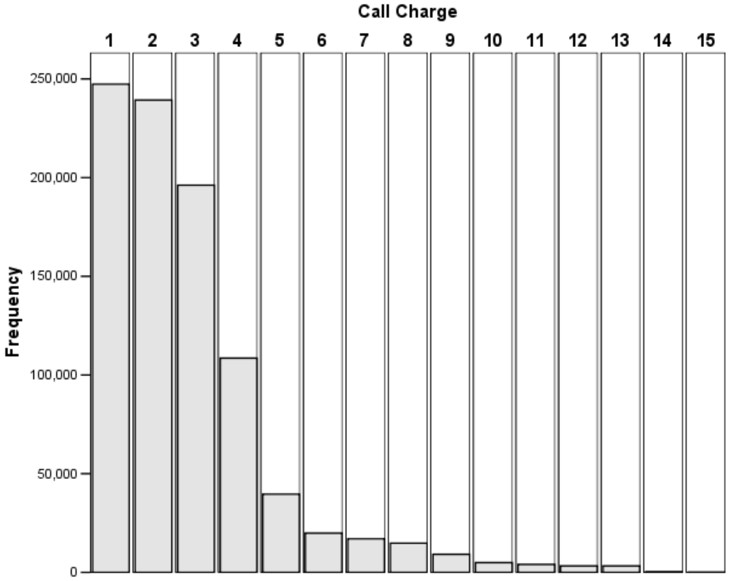

Figure 6.5: *Variable call_charge binned.*

the call data records (used to build the frequent item sets) belong. The subscriber's frequent item sets were used to construct a fingerprint, which may be used to identify uniquely the subscriber on the network, based on the subscriber's call behaviour.

The population of observations contains 504 call data records for subscriber 279899155, which constitude calls originating from the subscriber's handset. Table 6.17 shows the frequent item sets, I, contained in the fingerprint of subscriber 279899155, the probability of the item set occurring among the subscriber's call data records, denoted by $P[I|m_k]$, and the probability of the item set occurring in the entire population of call data records, $P[I]$. Focusing on the attribute *call_charge*, the fingerprint indicates that 10% of the subscriber's calls cost between $R3.00$ and $R4.00$, 11.6% between $R2.00$ and $R3.00$, 31.0% between $R1.00$ and $R2.00$, and 22.9% between $R0.30$ and $R1.00$. The attribute values for *call_day* shows that 20.5% of the subscriber's calls were made on Tuesdays, 11.2% on Wednesdays or Thursdays, 13.3% on Sundays, 20.3% on Fridays, and 24.8% on Saturdays. The subscriber's fingerprint also indicates that 11.2% of the subscriber's calls were made to number 631726426, and 99.2% were made to numbers located within the borders of South Africa. The values of the attribute *location_area* indicate that 99.5% of the subscriber's calls were made while the subscriber was in area 401 (Port Elizabeth and the surrounding area). Attribute *cell_id*, identifying the cell closest to the subscriber when making a call, indicates that 11.9% of the calls made by the subscriber, were made in close proximity to cell 20402 (UITENHAGE_2), and 30.9% were made in close proximity to cell 26252 (TOREGO_2). Focusing on the values attained by the attribute *call_duration*, the fingerprint indicates that 14.0% of the subscriber's calls last between 120s and 300s,

	Bin Boundary	**Observation**
Bin	**(Seconds)**	**Count**
01	$[0; 30]$	422 325
02	$(30; 60]$	220 048
03	$(60; 120]$	153 047
04	$(120; 300]$	85 934
05	$(300; 600]$	18 424
06	$(600; 1\,200]$	6 124
07	$(1\,200; 1\,800]$	1 112
08	$(1\,800; 2\,400]$	324
09	$(2\,400; 3\,000]$	126
10	$(3\,000; \infty)$	88

Table 6.16: *Call_duration bin boundaries, measured in seconds.*

21.7% between 60s and 120s, 19.7% between 30s and 60s, and 37.6% between 0s and 30s. Comparing $P[I|m_k]$ to $P[I]$ in Table 6.17 provides important insight into the subscriber's behaviour in relation to that of other subscribers on the network. The subscriber in question behaves similar to other subscribers in the network, when comparing attributes *call_charge*, *call_duration*, *call_day* and *call_destination*. However, the behaviour of the subscriber with respect to the attributes *other_party*, *location_area* and *cell_id* differs significantly from the behaviour of other subscribers, and may be used to distinguish this subscriber from others.

In the telecommunications industry, behaviour profiling on subscriber/account level is widely used by modern fraud detection and management tools. When a fraudulent subscriber is identified, the fraud analyst typically stores the subscriber's behaviour profile, taken from the subscriber's call data records. The fraud detection system then compares the fingerprint of the fraudulent subscriber to the behaviour profiles of all other subscribers, and generates a list of possible fraudulent subscribers, with a measure indicating the difference in behaviour between the fraudulent and suspected subscriber. The fraud analyst may then typically choose a list of suspected fraudulent subscribers to be investigated.

The fraud analyst may choose the fingerprint of subscriber $m_1 = 279899155$, given in Table 6.17, as a fraud indicator. Many measures exist, and more may be developed, with the ability of calculating the difference between subscriber behaviour profiles. For illustrative purposes the Euclidean distance measure was choosen. Table 6.18 compares the fingerprint of fraudster $m_1 = 279899155$ with the fingerprints of two potential fraudulent subscribers, $m_2 = 27982145$ and $m_3 = 27985684$. The Euclidean distance between fraudster m_1 and potential fraudster m_2 is calculated as $1.730\,977$, between fraudster m_1 and potential fraudster m_3 as $0.374\,476$, and between potential fraudsters m_2 and m_3 as

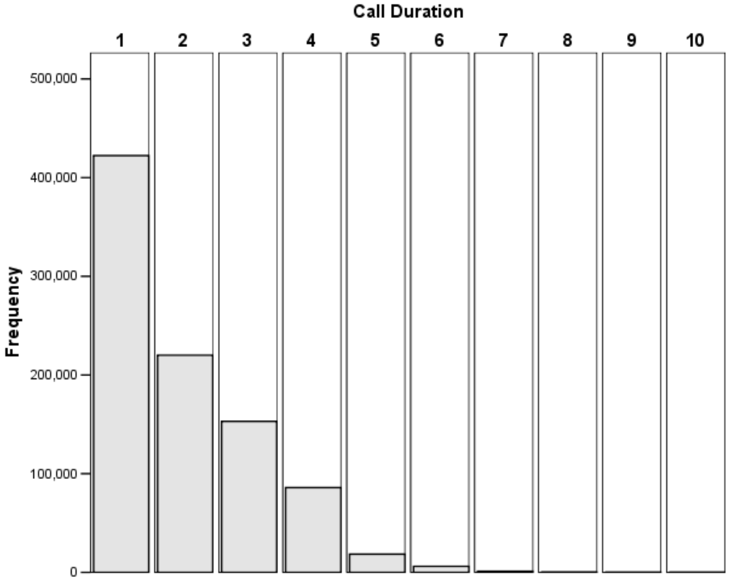

Figure 6.6: *Variable call_duration binned.*

1.765 743. Subscriber m_3 therefore exhibits similar behaviour to that of the known fraudster under the Euclidean distance metric, and may be marked for further investigation.

6.7 Chapter Summary

Well-known data mining methods (described in Chapter 2) were applied in this chapter to the set of call data records provided by a South African cellular network operator. Classification techniques making use of supervised learning were applied, building models with the ability to classify unseen call data observations as fraudulent or legitimate. Applying a sample of call data observations to the classification tree described in §6.1 resulted in 1 misclassification. Applying the same sample of call data observations to the artificial neural network described in §6.2 resulted in 3 misclassifications. The Bayesian classifier described in §6.3 was tested making use of the same method used to test the classification tree and artificial neural network, and resulted in 96 misclassifications. Cluster analysis and classification trees were combined in §6.4 to construct a method with the ability to cluster subscribers into behaviour groups and predict group membership on unseen call data observations. Observations classified into groups not fitting the subscriber's behaviour were identified as potential fraud. Outlier analysis using the Mahalanobis distance measure was applied in §6.5 on a set of legitimate call data observations. The maximum Mahalanobis distance was taken as a threshold value and new observations exceeding this value were identified as possible fraudulent activity. Association rule mining was applied

Frequent Item Set (I)	$P[I\|m_k]$	$P[I]$
$call\_charge = 5$	10.0%	4.36%
$call\_day = 3$	10.5%	14.27%
$other\_party = 631726426$	11.2%	0.01%
$call\_day = 4$	11.2%	13.53%
$call\_day = 5$	11.2%	13.80%
$call\_charge = 4$	11.6%	11.96%
$cell\_id = 20402$	11.9%	0.06%
$call\_day = 1$	13.3%	12.72%
$call\_duration = 4$	14.0%	9.47%
$call\_duration = 2$	19.7%	24.25%
$call\_day = 6$	20.3%	15.79%
$call\_duration = 3$	21.7%	16.86%
$call\_charge = 2$	22.9%	26.37%
$call\_day = 7$	24.8%	16.33%
$cell\_id = 26252$	30.9%	0.08%
$call\_charge = 3$	31.0%	21.61%
$call\_duration = 1$	37.6%	46.53%
$call\_destination = SouthAfrica$	99.2%	99.88%
$location\_area = 401$	99.5%	4.97%

Table 6.17: *Fingerprint for subscriber* 279899155 *given as the probabilities* ($P[I|m_k]$) *of the frequent item sets* (I) *occurring among the subscriber's* (m_k) *call data records.*

to the data in §6.6, building subscriber fingerprints from the frequent item sets identified by the Apriori algorithm. Fraud detection was achieved by comparing the fingerprint of a known fraudster to the behaviour profiles of all other subscribers, identifying subscribers displaying similar behaviour as potential fraudsters.

Frequent Item Set (I)	$P[I\|m_2]$	$P[I\|m_3]$	$P[I\|m_1]$
$call\_day = 1$	0.0%	16.9%	13.3%
$call\_day = 2$	13.4%	0.0%	0.0%
$call\_day = 3$	26.4%	10.4%	10.5%
$call\_day = 4$	18.2%	13.4%	11.2%
$call\_day = 5$	16.1%	14.9%	11.2%
$call\_day = 6$	0.0%	20.1%	20.3%
$call\_day = 7$	10.6%	21.4%	24.8%
$call\_charge = 1$	0.0%	20.4%	0.0%
$call\_charge = 2$	0.0%	23.9%	22.9%
$call\_charge = 3$	18.8%	22.4%	31.0%
$call\_charge = 4$	14.4%	0.0%	11.6%
$call\_charge = 5$	11.6%	0.0%	10.0%
$call\_charge = 6$	20.9%	0.0%	0.0%
$call\_duration = 1$	27.4%	55.7%	37.6%
$call\_duration = 2$	25.3%	17.4%	19.7%
$call\_duration = 3$	24.0%	18.9%	21.7%
$call\_duration = 4$	16.8%	0.0%	14.0%
$location\_area = 133$	100.0%	0.0%	0.0%
$location\_area = 401$	0.0%	99.5%	99.5%
$cell\_id = 20402$	0.0%	10.2%	11.9%
$cell\_id = 26252$	0.0%	22.9%	30.9%
$cell\_id = 53250$	42.8%	0.0%	0.0%
$cell\_id = 55210$	43.2%	0.0%	0.0%
$other\_party = 631726426$	0.0%	16.4%	11.2%
$other\_party = 9933156689555$	17.5%	0.0%	0.0%
$call\_destination = France$	33.6%	0.0%	0.0%
$call\_destination = SouthAfrica$	65.4%	97.5%	99.2%

Table 6.18: *Fingerprints for subscribers* $m_2 = 27982145$, $m_3 = 27985684$ *and* $m_1 = 29899155$ *given as the probabilities* ($P[I|m_k]$) *of frequent item sets* (I) *occurring among the subscriber's* (m_k) *call data records.*

Chapter 7

Conclusion

This chapter consists of three sections. In the first (§7.1) a brief summary of the work contained in this thesis is given, in the second (§7.2) an appraisal of the fraud detection methods employed in Chapter 6 is given, while in the third (§7.3) possible improvements to the work presented in this study as well as some ideas with respect to further work are outlined. Modern fraud management systems make use of proprietary data mining techniques and proclaim the capability to detect most types of telecommunications fraud. This chapter closes with a design for a comprehensive fraud detection model suggested by the author, implementing a combination of data mining techniques and with the ability of outperforming models based on a single data mining methodology.

7.1 Thesis Summary

Apart from the introductory chapter, in which fraud detection in telecommunication networks was introduced and the problem description and thesis objectives were given, and the current chapter (the conclusion), this thesis comprises a further five chapters.

Well-known data mining methodologies used in this thesis for data exploration, the building of user behaviour profiles and the detection of fraud were described in Chapter 2. Linear and logistic regression were discussed in the context of data exploration and variable selection, and a comparison was drawn between these two methods. *Decision trees*, *artificial neural networks* and *Bayesian classification* techniques were studied as examples of data mining techniques able to learn from classified observations and with the ability to predict the response value of unseen observations. Data mining methodologies with the capacity to learn from observations, but without a known response, were also studied, including the methods of *cluster analysis*, *outlier analysis* and *association rule mining.*

The focus of Chapter 3 was on the cellular telecommunications industry. The architecture of a cellular telecommunications network was described, providing the reader with an understanding of the different components in such a network. Background on the operation of a cellular telecommunications network was provided, and the flow of data through such a network was described, starting at the point in time when a call is placed until the time that it is charged. A separate section was devoted to a discussion on methods typically used by fraudsters to defraud cellular network operators. The chapter closed with three elements required in a successful fraud management strategy: fraud prevention, fraud detection and fraud deterrence.

Chapter 4 contains a brief overview of fraud detection literature available, paying particular attention to fraud detection methods employed in cellular telecommunications networks. The fraud detection literature items described in Chapter 4 were categorised according to their ability to detect fraud in real-time or at fixed points in time. The concept of customer behaviour profiles, also called account signatures, was introduced and methods traditionally used to build and maintain such profiles were described.

Chapter 5 opened with a description of the data collection process and definitions of the different attributes comprising each call data record. The introduction into this data set of artificially created call data records indicating fraudulent behaviour was motivated. The necessary preparation of the data for use in data mining techniques was described. New attributes were derived from the collected call data and added to the data set. Insight into the most important characteristics of the data set was provided by applying regression techniques as part of the forward variable selection method.

The data set described in Chapter 5 was further transformed in Chapter 6, as dictated by the data mining method applied to it, in the sense that the number of records were reduced into sets of statistics and certain attributes were binned. The techniques of *classification trees*, *artificial neural networks*, *Bayesian classifiers*, *cluster analysis*, *outlier analysis* and *association rule mining* were applied to the transformed data. The results emanating from these applications were described, and the suitability of these methods to be used in the fraud detection process was discussed.

Three classification methods applying supervised learning were presented in Chapter 6. The first classification method applied to the set of observations was *classification trees*. The number of records in the data set were reduced to one set of daily statistics for each subscriber. The classification tree provided a set of rules that may be used to detect fraud at the end of each daily period. The performance of the classification tree was tested by using the derived rules to classify a set of observations. The confusion matrix, indicating the inherent ability of the classification tree to distinguish between fraudulent and legitimate behaviour, indicated that the classification tree fits the set of observations well with only 1 observation out of a total of 1 085 observations being misclassified.

The second classification method applied was *artificial neural networks*, learning from the same set of data used to train the classification tree. The result of classifying observations using the artificial neural network was that 3 observations out of a total 1 085 observations were misclassified.

The final classification method applied was *Bayesian classifiers*, learning from the same set of data used to train the classification tree and artificial neural network. The confusion matrix, indicating the ability of the Bayesian classifier to distinguish between fraudulent and legitimate behaviour, indicated that the Bayesian classifier did not fit the particular set of observations well, with 96 observations out of a total 1 085 observations misclassified.

Three data mining methods applying unsupervised learning were presented in Chapter 2 and applied to call data in Chapter 6. *Cluster analysis* was performed on a learning sample, and rules predicting cluster membership derived by constructing a classification tree on the clustered observations. The classification tree indicated that it was well-suited for predicting cluster membership with only 18 misclassifications when applied to test sample of 8 161 observations.

The ability of the derived classification tree to assign unseen legitimate call data records to a cluster fitting the subscriber's profile was tested and the results analysed. The identification of fraudulent activity when new observations are clustered into clusters not typical for the subscriber was discussed. Cluster analysis was identified as a good method for grouping subscribers into behaviour profiles.

Outlier analysis using the Mahalanobis distance measure was applied to the population of normalised legitimate call data records. The observations with largest Mahalanobis distance are provided in Table 6.13. The maximum Mahalanobis distance was taken as a threshold value and new observations exceeding this value were identified as possible fraudulent activity.

Finally, the *Apriori algorithm*, used for mining frequent item sets in *association rule mining*, was applied to each subscriber's set of call data records in an attempt to derive a unique fingerprint for each. The use of fingerprinting to identify fraudulent behaviour, comparing the fingerprint of the fraudulent subscriber to the behaviour profiles of all other subscribers in an attempt to identify possible fraudulent subscribers, was discussed, and found to be a feasible technique for creating subscriber behaviour profiles.

7.2 Appraisal of Fraud Detection Methods

The *classification tree* constructed in §6.1 was described by a set of three rules, which resulted in one misclassification when applied to the same data set used as the learning sample, indicating a good fit on the training sample. However, applying these rules to a larger sample (of 109 592 observations) of daily subscriber statistics results in a very large number of misclassifications, with the confusion matrix given in Table 7.1. The

		$\hat{y}_i$		
		0	1	Errors
y_i	0	107 576	1 923	1 923
	1	9	84	9
	Errors	9	1 923	

Table 7.1: *Confusion matrix of the classification tree constructed in Chapter 6 applied to a more representative test sample of* 109 592 *observations. The classification tree misclassified* 1 932 *observations, including* 1 923 *false positives and* 9 *false negatives.*

majority of these misclassifications (1 919 false positives) are as a result of the second rule describing the classification tree in §6.1, indicating that fraudulent subscribers make calls of total daily duration less than or equal to 3.5 seconds. This rule may fit the behaviour of a small number of subscribers, but for the majority of subscribers, making calls with a daily duration of less than or equal to 3.5 seconds does not signify fraud, as indicated by the confusion matrix in Table 7.1. The *artificial neural network* and *Bayes classifier* derived from the learning sample in Chapter 6 were also applied to the larger test sample with similar results given by the confusion matrices in Tables 7.2 and 7.3 respectively.

		$\hat{y}_i$		
		0	1	Errors
y_i	0	106 955	2 544	2 544
	1	13	80	13
	Errors	13	2 544	

Table 7.2: *Confusion matrix of the artificial neural network constructed in Chapter 6 applied to a more representative test sample of* 109 592 *observations. The artificial neural network misclassified* 2 557 *observations, including* 2 544 *false positives and* 13 *false negatives.*

The models created by the classification techniques applied in Chapter 6 are not able to distinguish between different subscribers, which may be the reason for the large number of misclassifications. This confirms the statement made in the introduction of this thesis; that callers are dissimilar, so that calls appearing to be fraud for one account, may seem to be expected behaviour for other accounts. Fraud detection must therefore be tailored to each account's own activity. Classification methods may play an important role in the fraud detection process, but not as stand-alone fraud detection techniques, given the large number of misclassifications when applied to an unseen data set. In §6.4 the ability of classification trees to describe training data with a set of rules was instrumental in

		$\hat{y}_i$		
		0	1	Errors
y_i	0	102 220	7 279	7 279
	1	89	4	89
Errors		89	7 279	

Table 7.3: *Confusion matrix of the Bayes classifier constructed in Chapter 6 applied to a more representative test sample of* 109 592 *observations. The Bayes classifier misclassified* 7 368 *observations, including* 7 279 *false positives and* 89 *false negatives.*

dividing new observations into different behaviour groups (clusters). Classification trees may also play an important role in guiding fraud analysts when defining fraud detection rules in rule-based fraud detection systems. The ability of artificial neural networks and Bayesian classifiers to learn the behaviour of known fraudulent subscribers in order to detect similar behaviour may be used to confirm fraud and assign fraud probability to subscribers suspected of being defrauded.

The ability of *cluster analysis* to discover natural groupings in observations, based on the similarities between them, makes it ideal for use in behaviour profiling. Based on the method of cluster analysis (and combined with other data mining methods) a fraud detection technique may be developed with the ability of real-time fraud detection. In §6.4 a classification tree was constructed on the clustered observations and the rules describing the classification tree were used to assign new call data records to behaviour-fitting clusters. Call data records classified into clusters not typical for the subscriber may be identified as possible fraudulent activity.

The *outlier analysis* in §6.5 applied the Mahalanobis distance measure to all call data observations without distinguishing between the subscribers generating these observations. This technique was unable to detect the majority of fraudulent call data observations with only 10 fraudulent observations detected out of a total of 175. However, outlier analysis may be tailored to each subscriber's own activity, thereby defining a threshold value fitting the subscriber's behaviour. Outlier analysis is another technique with the ability to detect fraud in real-time, because it evaluates the outlier-status of individual call data observations.

In §6.6 the Apriori algorithm's frequent item sets were used in an attempt to derive a fingerprint unique to each subscriber. The subscriber's fingerprint is a summary of behaviour during a certain time period. This technique may be used to create fingerprints of known fraudsters and search for similar fingerprints as an indication of fraud. Another application of this technique may be to detect changes in user behaviour by comparing each subscriber's fingerprint based on long-term historical behaviour with the subscriber's fingerprint based on recent short-term behaviour. Statistical scoring methods may also be developed for computing the difference between two fingerprints: the higher such a score the greater the behaviour change and the more suspicious the subscriber's account.

7.3 Possible Further Work

In view of the revenue loss experienced by cellular network operators due to telecommunications fraud, it would be valuable to research real-time fraud detection techniques with the ability to detect fraud accurately from the first call made by a subscriber.

An area of potential study is to evaluate the use of hybrid techniques in fraud detection. Such a technique was discussed in §6.4, combining cluster analysis and classification trees to group subscribers based on their past behaviour and predict group membership of unseen observations. Another option would be to construct group-specific classification trees, learning from the classified observations in that behaviour group. The derived group-specific rules may then be used to classify unseen call data observations for the subscribers belonging to that behaviour group as fraudulent or legitimate. Classification trees may be replaced with a different classification method, like Bayesian classifiers or artificial neural networks.

Another area of potential study is the creation of meaningful initial behaviour profiles for new subscribers so that fraud detection techniques may be applied from the first call data record. One way to achieve this is to segment the behaviour profiles for existing subscribers based on information available in the first one or few calls for a subscriber.

Subscriber behaviour may change over time demanding a technique to keep the subscriber's behaviour profile current, which is another area of potential study. A technique capable of detecting temporal changes was mentioned in §4.2.1, using exponentially weighted moving averaging for updating the subscriber behaviour profile with each new call data record.

Suggestion as to a Comprehensive Fraud Detection Model

It is advisable that fraud management systems employ a large number of different fraud detection techniques, each one contributing towards estimating a subscriber's overall fraud probability. Modern fraud management systems make use of proprietary data mining techniques and proclaim the capability to detect most types of telecommunications fraud. The author suggests, in Figure 7.1, such a combination of data mining techniques that may be used to build a comprehensive fraud detection model that is capable of outperforming models based on a single data mining methodology. The fraud detection model suggested in Figure 7.1 comprises four logical areas: data input (A), fraud detection (B), alarm analysis and case creation (C), and detection model adaptations (D).

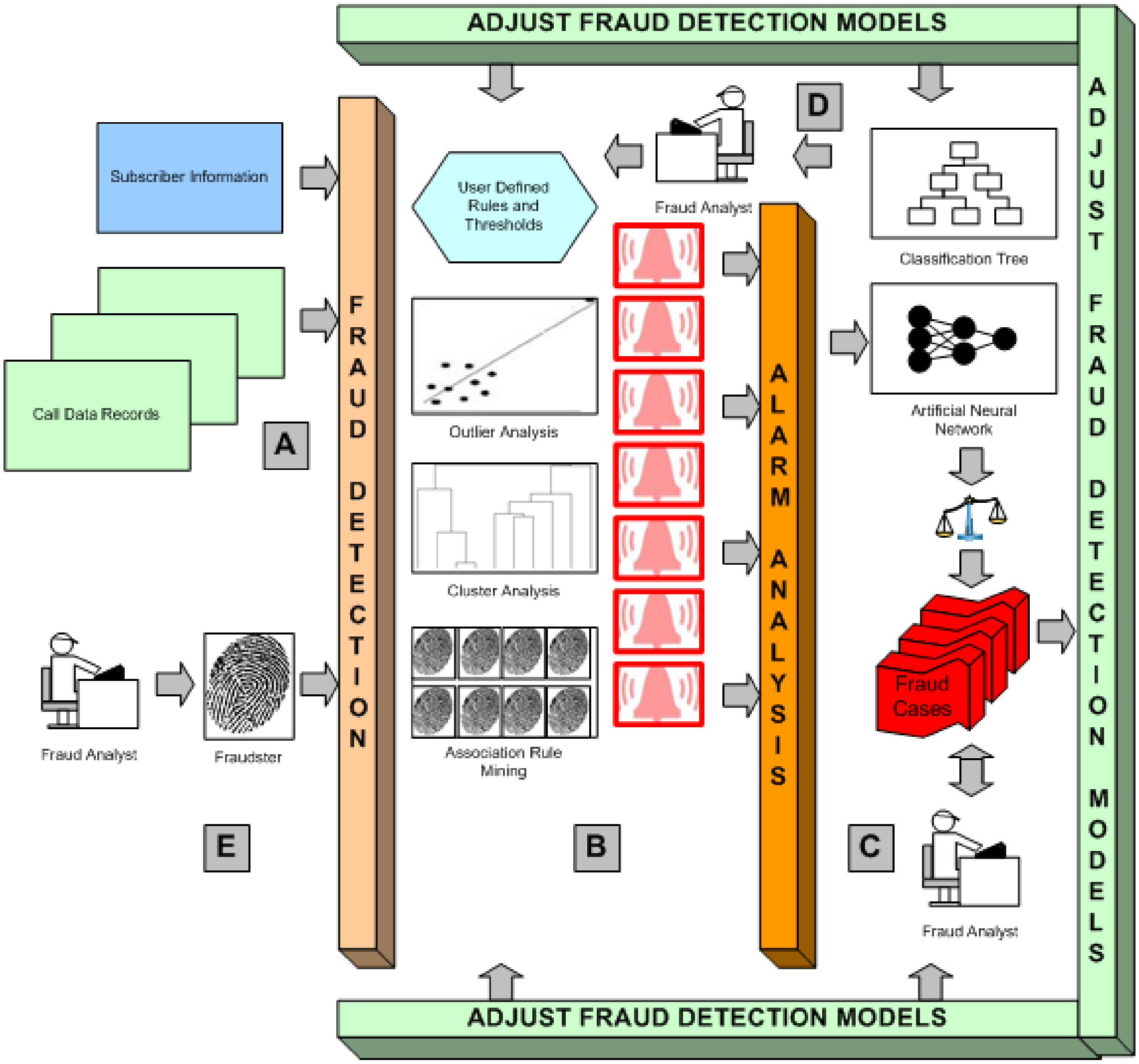

Figure 7.1: *Suggested architechture for a comprehensive fraud detection model implementing a combination of data mining techniques, detection rules and threshold values.*

The network profile of each subscriber, indicating the subscriber's status and provisioned services in the cellular network, as well as call data records, serve as input to the suggested fraud detection model (Figure 7.1 A).

A combination of fraud detection methods (Figure 7.1 B) may be applied to the input data, each one with the ability to generate alarms and contribute to the subscriber's fraud probability. The fraud detection techniques employed to detect fraud include techniques based on classification trees (described in §2.1 and applied in §6.1), artificial neural networks (described in §2.4 and applied in §6.2), cluster analysis (described in §2.6 and applied in §6.4), outlier analysis (described in §2.7 and applied in §6.5) and association rule mining (described in §2.8 and applied in §6.6). The ability to construct fraud detection models using rules and threshold values may be included as an additional fraud detection technique.

Hybrid fraud detection techniques, employing cluster analysis and classification trees, may be used to cluster subscribers into behaviour groups, using subscribers' available call data records to define behaviour clusters. New subscribers may be assigned to initial clusters describing the behaviour expected by these subscribers. A classification tree may be used to assign new call data records to behaviour-fitting clusters. Call data records classified into clusters not typical for subscribers may then raise an alarm.

The method of outlier analysis may be applied to subscribers' available call data records, using maximum Mahalanobis distances to determine threshold values. The Mahalanobis distances between new call data records and subscribers' call data population means may be calculated and alarms may be raised when the Mahalanobis distances are greater than the threshold values set for these subscribers.

The frequent item set procedure, implemented as part of the Apriori algorithm in association rule mining, may be used to create fingerprints for subscribers, based on available call data records. At the end of each profiling period new fingerprints may be created, describing recent calling behaviour, and may be compared with the subscribers' saved fingerprints. A significant change in behaviour should result in an alarm being raised. Functionality to label fingerprints of known fraudsters may be provided to fraud analysts (Figure 7.1 E). Fingerprints similar to the fingerprints of labelled fraudsters may be used to raise additional alarms.

The ability to construct fraud detection rules and define threshold values may be included by means of a classification tree, constructed on call data records belonging to cases resolved as fraudulent or legitimate, as the source of potential rules. Call data records may then be validated against the rules and threshold values and alarms may be raised when rules are matched or threshold values exceeded.

Alarms generated by the fraud detection techniques (Figure 7.1 B) may serve as input to the alarm analysis area of the fraud detection model (Figure 7.1 C). The generated alarms may then be analysed by the alarm analysis engine and fraud cases may be created when the number and severity of alarms on subscribers exceed defined threshold values. An artificial neural network, constructed from fraud cases resolved as fraudulent or legitimate, may be used to assign fraud probabilities to cases. Fraud analysts should be able to investigate cases further (manually) and resolve them as fraudulent or legitimate, based on experience.

The data mining models implemented may be kept up to date with changes in subscriber behaviour by means of updating behaviour profiles, adjusting threshold values and revising classification models (Figure 7.1 D). Only legitimate call data records should be employed in updating subscriber behaviour profiles used in cluster analysis and association rule mining, and threshold values defined as part of outlier analysis. Call data records raising alarms on cases resolved as legitimate behaviour should also be employed in adjusting behaviour profiles on legitimate subscribers. Classification techniques, including classification trees and artificial neural networks, may be revised based on call data records on resolved cases.

Bibliography

[1] A.A. Afifi & V. Clark, *Computer–Aided Multivariate Analysis* (3rd ed), Chapman and Hall, Los Angeles (CA), 1996.

[2] R. Agrawal & R. Srikant, *Fast Algorithms for Mining Association Rules*, [Online], [cited 2003, Aug 16], Available from: `http://citeseer.nj.nec.com/392628.html`

[3] S.D. Bay & M. Schwabacher, *Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule*, [Online], [cited 2003, Aug 20], Available from: `http://citeseer.nj.nec.com/569806.html`

[4] M.J.A. Berry & G. Linoff, *Data Mining Techniques (For Marketing, Sales, and Customer Support)*, John Wiley and Sons, Inc., New York (NY), 1996.

[5] C. Borgelt, *Christian Borgelt's Webpages*, [Online], [cited 2003, Jul 15], Available from: `http://fuzzy.cs.uni-magdeburg.de/∼borgelt/software.html`

[6] C. Borgelt & R. Kruse, *Introduction of Association Rules: Apriori Implementation*, [Online], [cited 2004, Jul 25], Available from: `http://fuzzy.cs.uni-magdeburg.de/∼borgelt/papers/cstat_02.pdf`

[7] E. Brand & R. Gerritsen, *Decision Trees*, [Online], [cited 2003, Apr 30], Available from: `http://www.dbmsmag.com/9807m05.html`

[8] L. Breiman, J.H. Friedman, R.A. Olshen & C.J. Stone, *Classification and Regression Trees*, Wadsworth Inc., Montery (CA), 1984.

[9] The Business Day, *Check that phone bill before you pay*, [Online], [cited 2004, Jan 10], Available from: `http://www.bday.co.za/bday/content/direct/1,3523,1309612-6078-0,00.html`

[10] M.H. Cahill, D. Lambert, J.C. Pinheiro & D.X. Sun, *Detecting Fraud in the Real World*, [Online], [cited 2003, Apr 30], Available from: `http://stat.bell-labs.com/cm/ms/departments/sia/doc/HMDS.pdf`

[11] F. Chen, D. Lambert & J.C. Pinheiro, *Incremental Quantile Estimation for Massive Tracking*, [Online], [cited 2003, Apr 30], Available from: http://stat.bell-labs.com/cm/ms/departments/sia/doc/KDD2000.pdf

[12] F. Chen, D. Lambert, J.C. Pinheiro & D.X. Sun, *Reducing Transaction Databases, Without Lagging Behind the Data or Losing Information*, [Online], [cited 2003, Apr 30], Available from: http://stat.bell-labs.com/cm/ms/departments/sia/doc/vldb.pdf

[13] T. Fawcett & F. Provost, *Activity Monitoring: Noticing Interesting Changes in Behavior*, [Online], [cited 2003, Apr 30], Available from: http://cs.oregonstate.edu/∼tgd/ml2001-workshop/provost.pdf

[14] T. Fawcett & F. Provost, *Adaptive Fraud Detection*, [Online], [cited 2003, Apr 30], Available from: http://www.hpl.hp.com/personal/Tom_Fawcett/papers/DMKD_97.ps.qz

[15] U.M. Fayyad, G. Piatetsky–Shapiro, P. Smyth & R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, 1996.

[16] *Geobytes: City Distance Tool*, [Online], [cited 2004, May 27], Available from: http://www.geobytes.com/CityDistanceTool.htm?loadpage

[17] E.I. George, *The Variable Selection Problem*, [Online], [cited 2003, Apr 30], Available from: http://www-stat.wharton.upenn.edu/∼edgeorge/Research_papers/vignette.pdf

[18] N. Gough & C. Grezo, *Africa: The Impact of Mobile Phones*, [Online], [cited 2005, Mar 20], Available from: http://www.gsmworld.com/documents/external/vodafone_africa_report05.pdf

[19] *Gower's Similarity Coefficient*, [Online], [cited 2004, May 23], Available from: http://www.clustan.com/gower_similarity.html

[20] J.F. Hair, R.E. Anderson, R.L. Tatham & W.C. Black, *Multivariate Data Analysis* (5th ed), Prentice Hall, Upper Saddle River (NJ), 1998.

[21] J. Han & M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco (CA), 2001.

[22] J. Hollmén, *User Profiling and Classification for Fraud Detection in Mobile Communications Networks*, [Online], [cited 2003, Apr 30], Available from: http://lib.hut.fi/Diss/2000/isbn9512252392/

[23] J. Hollmén & V. Tresp, *Call–based Fraud Detection in Mobile Communication Networks using a Hierarchical Regime–Switching Model*, [Online], [cited 2003, Apr 30], Available from: `http://lib.hut.fi/Diss/2000/isbn9512252392/article3.pdf`

[24] D.W. Hosmer & S. Lemeshow, *Applied Logistic Regression* (2nd ed), John Wiley & Sons Inc., New York (NY), 2000.

[25] *International Numbering Plans*, [Online], [cited 2004, May 31], Available from: `http://www.numberingplans.com/index.php?goto=areacodes&s=ZA&len=5`

[26] R. Jacobs, *Telecommunications Fraud*, [Online], [cited 2005, Mar 28], Available from: `http://www.didata.com/DocumentLibrary/WhitePapers/TelecommunicationsFraudWhitePaper.htm`

[27] M.D. Jankowitz, *Hiërargiese en Nie-Hiërargiese Trosontledingstegnieke: 'n Empiriese Vergelyking*, Masters Thesis, Stellenbosch University, Stellenbosch, 1990.

[28] R.A. Johnson & D.W. Wichern, *Applied Multivariate Statistical Analysis* (5th ed), Prentice-Hall Inc., New York (NY), 2002.

[29] D. Lambert, J.C. Pinheiro & D.X. Sun, *Estimating Millions of Dynamic Timing Patterns in Real–Time*, [Online], [cited 2003, Apr 30], Available from: `http://stat.bell-labs.com/cm/ms/departments/sia/doc/chron.pdf`

[30] Y. Moreau, B. Preneel & K.U. Leuven, *Definition of Fraud Detection Concepts*, [Online], [cited 2003, Oct 23], Available from: `ftp://ftp.cs.rhbnc.ac.uk/pub/aspect/asp_d06.pdf`

[31] B.W. Morgan, *An Introduction to Bayesian Statistical Decision Processes*, Prentice-Hall Inc., Englewood Cliffs (NJ), 1968.

[32] *MTN Supports crime prevention*, [Online], [cited 2003, Apr 24], Available from: `http://www.mtn.co.za/home/news/doNews.asp?item=33&arc=y`

[33] S.M. Ross, *Introduction to Probability Models* (7th ed), Academic Press, San Diego (CA), 2000.

[34] W.S. Sarle, *Neural Networks and Statistical Models*, [Online], [cited 2003, Apr 30], Available from: `http://www.foretrade.com/Documents/NeuralNetwork.pdf`

[35] S.L. Scott, *A Bayesian Paradigm for Designing Intrusion Detection Systems*, [Online], [cited 2003, Apr 30], Available from: `www-rcf.usc.edu∼sls/bayes_nid.pdf`.

[36] J. Scourias, *Overview of the Global System for Mobile Communications*, [Online], [cited 2005, Mar 26], Available from: `http://ccnga.uwaterloo.ca/∼jscouria/GSM/gsmreport.html`

[37] *Statistics of Cellular in South Africa*, [Online], [cited 2005, Mar 20], Available from: `http://www.cellular.co.za/stats/statistics_south_africa.htm`

[38] M. Taniguchi, M. Haft, J. Hollmén & V. Tresp, *Fraud Detection in Communications Networks using Neural and Probabilistic Methods*, [Online], [cited 2003, Apr 30], Available from: `http://www.cis.hut.fi/jhollmen/Publications/icassp98.pdf`

[39] Vodaworld Magazine, *Cellular Fraud in South Africa*, [Online], [cited 2005, Mar 28], Available from: `http://www.vodaworld.co.za/showarticle.asp?id=1154`

[40] B. Warber & M. Misra, *Understanding Neural Networks as Statistical Tools*, The American Statistician, 50(4) (1996), 284–293.

[41] A.R. Webb, *Statistical Pattern Recognition* (2nd ed), John Wiley & Sons Inc., Chichester, 2003.

[42] W.L. Winston, *Operations Research: Applications and Algorithms* (3rd ed), Duxbury Press, Belmont (CA), 1994.

[43] J. Yu, *Clustering Methods: Applications of Multivariate Statistical Analysis*, [Online], [cited 2004, Jul 17], Available from: `icl.pku.edu.cn/yujs/papers/pdf/cluster.pdf`

Appendix A

Computer Programs

This appendix is devoted to providing the reader access to most of the computer programs that were used to prepare the data and apply the data mining techniques. Most of the computer programs used in this thesis were downloaded from Christian Borgelt's webpages [5] and are available on the attached compact disk. These programs were executed on a personal computer with one 2.4GHz central processing unit, 1Gb random access memory and a Microsoft Windows 2000 operating system.

Instructions on how to construct classification trees, artification neural networks, naive Bayesian classifiers and association rules using the software available on the attached compact disk are provided in §§A.1, A.2, A.3 and A.6, respectively. A C++ routine implementing cluster analysis is provided in §A.4. An Oracle PL/SQL routine calculating the Mahalanobis distance between observations is provided in §A.5. SPSS 13.0 for Windows and Statistica 6.0 were used for statistical analysis of the data and variable selection. The data set used in this thesis is available in the directory `\Data\` on the compact disc.

A.1 Classification Tree

The C++ implementation of the decision tree algorithm [5] (described in §2.1) was used to construct and execute a classification tree on the available classified observations. The decision tree program executables, source code and helpfile may be found on the attached compact disk, in the directory `\DTrees\`.

```
The domain file construction program determining the domains of observation attributes and is invoked
as follows:

dom.exe [options] [-d|-h hdrfile] tabfile domfile

The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
domfile  file to write domain descriptions to
```

```
The possible options are:

-s       sort domains alphabetically (default: order of appearance)
-S       sort domains numerically/alphabetically
-a       automatic type determination (default: all symbolic)
-i       do not print intervals for numeric attributes
-l#      output line length (default: no limit)
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

The decision tree construction program is invoked as follows:

```
dti.exe [options] domfile [-d|-h hdrfile] tabfile dtfile

The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
dtfile   file to write induced decision/regression tree to
domfile  file containing domain descriptions

The possible options are:

-c#      target attribute name (default: last attribute)
-q#      balance class frequencies (weight tuples)
         l: lower, b: boost, s: shift tuple weights
-e#      attribute selection measure (default: infgr/rmse)
-!       print a list of available attribute selection measures
-z#      sensitivity parameter (default: 0)
         (for measures wdiff, bdm, bdmod, rdlen1, rdlen2)
-p#      prior (positive value) or equivalent sample size (negative value)
         (for measures bdm, bdmod)
-i#      minimal value of the selection measure (default: no limit)
-w       do not weight measure with fraction of known values
-t#      maximal height of the tree (default: no limit)
-m#      minimal number of tuples in two branches (default: 2)
-s       try to form subsets on symbolic attributes
-l#      output line length (default: no limit)
-a       align values of test attributes (default: do not align)
-v       print relative frequencies (in percent)
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

```
The decision tree execution program is invoked as follows:

dtx.exe [options] dtfile [-d|-h hdrfile] tabfile [outfile]

The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
outfile  file to write output table to (optional)
dtfile   file containing decision/regression tree description

The possible options are:

-p#      prediction field name (default: dt)
-s#      support    field name (default: no support    field)
-c#      confidence field name (default: no confidence field)
-a       align fields (default: do not align)
-w       do not write field names to output file
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

```
The decision tree rule extraction program is invoked as follows:

dtr.exe [options] dtfile rsfile

The normal arguments are:

dtfile   file containing decision/regression tree description
rsfile   file to write rule set description to

The possible options are:

-s       print support    of a rule
-c       print confidence of a rule
-d       print only one condition per line
-l#      output line length (default: no limit)

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

A.2 Artificial Neural Network

The C++ implementation of the multilayer perceptron algorithm [5] (described in §2.4) was used to construct and execute a neural network on the available classified observations. The neural network program executables, source code and helpfile may be found on the attached compact disk, in the directory `\NeuralNetwork\`.

```
The domain file construction program determining the domains of observation attributes and is invoked
```

as follows:

```
dom.exe [options] [-d|-h hdrfile] tabfile domfile
```

The normal arguments are:

```
hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
domfile  file to write domain descriptions to
```

The possible options are:

```
-s       sort domains alphabetically (default: order of appearance)
-S       sort domains numerically/alphabetically
-a       automatic type determination (default: all symbolic)
-i       do not print intervals for numeric attributes
-l#      output line length (default: no limit)
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile
```

(# always means a number, a letter, or a string that specifies the parameter of the option.)

The neural network training program is invoked as follows:

```
mlpt.exe [options] domfile [-d|-h hdrfile] tabfile mlpfile
```

The normal arguments are:

```
hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
mlpfile  file to write multilayer perceptron to
domfile  file containing domain descriptions
```

The possible options are:

```
-o#      output/target attribute name (default: last attribute)
-c#:#..  number of units per hidden layer (default: no hidden layer)
-x#      expansion factor for output ranges (default: 1)
-w#      initial weight range     (default: 1)
-a#      weight update method     (default: 0)
-!       print a list of available weight update methods
-t#      learning rate            (default: 0.2)
-z#:#    minimal and maximal change/learning rate (default: 1e-006:16)
-g#:#    growth and shrink factor (default: 1.2:0.5)
-i#      flat spot elimination    (default: 0)
-m#      momentum coefficient     (default: 0)
-y#      weight decay factor      (default: 0)
-j#      range for weight jogging (default: 0)
-T#      error for termination    (default: 0)
-e#      maximum number of training epochs (default: 1000)
-k#      patterns between two weight updates (default: 1)
         (0: update at the end of each epoch)
-v#      verbose output (print sse every # epochs)
```

```
-q       do not normalize input value ranges
-s       do not shuffle patterns between epochs
-S#      seed value for random number generation (default: time)
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-l#      output line length (default: no limit)
         (and maybe a pretrained network)
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

```
The neural network execution program is invoked as follows:

mlpx.exe [options] mlpfile [-d|-h hdrfile] tabfile [outfile]

The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
outfile  file to write output table to (optional)
mlpfile  file to read multilayer perceptron from

The possible options are:

-p#      prediction field name (default: nn)
-c#      confidence field name (default: no confidence field)
-x       print extended confidence information
-a       align fields (default: do not align)
-w       do not write field names to output file
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

A.3 Naive Bayesian Classification

The C++ implementation of the naive Bayesian classification algorithm [5] (described in §2.5) was used to construct and execute a naive Bayesian classifier on the available classified observations. The Bayesian classifier program executables and source code may be found on the attached compact disk, in the directory `\Bayes\`.

```
The domain file construction program determining the domains of observation attributes and is invoked
as follows:

dom.exe [options] [-d|-h hdrfile] tabfile domfile

The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
```

```
domfile  file to write domain descriptions to

The possible options are:

-s       sort domains alphabetically (default: order of appearance)
-S       sort domains numerically/alphabetically
-a       automatic type determination (default: all symbolic)
-i       do not print intervals for numeric attributes
-l#      output line length (default: no limit)
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

```
The Bayesian classifier construction program is invoked as follows:

bci.exe [options] domfile [-d|-h hdrfile] tabfile bcfile

The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
bcfile   file to write Bayes classifier to
domfile  file containing domain descriptions

The possible options are:

-F       induce a full Bayes classifier (default: naive Bayes)
-c#      class field name (default: last field)
-w#      balance class frequencies (weight tuples)
         l: lower, b: boost, s: shift weights
-s#      simplify classifier (naive Bayes only)
         a: by adding, r: by removing attributes
-L#      Laplace correction (default: 0)
-t       distribute tuple weight for unknown values
-m       use maximum likelihood estimate for the variance
-p       print relative frequencies (in percent)
-l#      output line length (default: no limit)
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default table header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

```
The Bayesian classifier execution program is invoked as follows:

bcx.exe [options] bcfile [-d|-h hdrfile] tabfile [outfile]
```

```
The normal arguments are:

hdrfile  file containing table header (field names)
tabfile  table file to read (field names in first record)
outfile  file to write output table to (optional)
bcfile   file containing classifier description

The possible options are:

-c#      classification field name (default: bc)
-p#      confidence/probability field name (default: no confidence output)
-L#      Laplace correction (default: as specified in classifier)
-v/V     (do not) distribute tuple weight for unknown values
-m/M     (do not) use maximum likelihood estimate for the variance
-a       align fields (default: do not align)
-w       do not write field names to the output file
-b/f/r#  blank characters, field and record separators
         (default: " \t\r", " \t", "\n")
-u#      unknown value characters (default: "?")
-n       number of tuple occurrences in last field
-d       use default table header (field names = field numbers)
-h       read table header (field names) from hdrfile

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```

A.4 Cluster Analalysis

The section of code provided below uses Gower's general similarity coefficient given in (2.22) to populate an initial similarity matrix $\mathbf{D} = \{d_{ik}\}$. McQuitty's similarity analysis (see §2.6.1) was then employed to merge the two most similar observations, and to compute the similarity between the newly formed cluster and the remaining clusters, updating the similarity matrix with this value. The source code may be found on the attached compact disk, in the directory `\Cluster\`.

```
//Loop through the available observations in the data set

for (k=0;k<i;k++)
{
//Calculate similarity matrix only for entries below the diagonal to avoid performing the same similarity
//calculation twice.
  for (l=0;l<=k;l++)
  {
    //Initialise the similarity measure
    dist = 0.0;
    //No need to calculate the similarity between observation k and k.
    if (l<k)
    {
      //Calculate the similarity between categorical explanatory variables
      //in observations k and l where l <> k
      if (strcmp(prepaid_ind[k],prepaid_ind[l])==0)
        dist = dist+1.0;
      if (strcmp(peak_ind[k],peak_ind[l])==0)
        dist = dist+1.0;
      //Calculate the similarity between ordinal and continuous explanatory variables in observations
      //k and l where l <> k
      dist = dist + (1.0 - (fabs(atof(call_duration[k]) - atof(call_duration[l])))/v_call_duration_range) +
                    (1.0 - (fabs(atof(call_charge[k]) - atof(call_charge[l])))/v_call_charge_range) +
                    (1.0 - (fabs(atof(tariff_ind[k]) - atof(tariff_ind[l])))/v_tariff_range) +
                    (1.0 - (fabs(atof(location_ind[k]) - atof(location_ind[l])))/v_location_range) +
                    (1.0 - (fabs(atof(terminating_ind[k]) - atof(terminating_ind[l])))/v_terminating_range);

      //Calculate the average similarity across 7 explanatory variables
      dist = dist / 7.0;
    }
    //update entry [k][l] of the similarity matrix with the calculated similarity between
    //observations k and l
    sprintf(buffer,"%.4f",dist);
    strcpy(matrix[k][l],buffer);
  }
}

//Use the similarity matrix calculated above and start merging most similar observations into clusters
//Initialise the matrix of clusters
for (m=0;m<i;m++)
{
  clusters[m].distance = 0;
  clusters[m].deleted = 0;
  sprintf(clusters[m].clustid,"%d",m);
  clusters[m].clustcount = 1;
  sprintf(clusters[m].clustid_concat,"%d",m);
}

//Loop
for(;;)
{
  //Find the two most similar observations or cluster of observations
  for (m=0;m<i;m++)
  {
    for (n=0;n<=m;n++)
    {
      if (atof(matrix[m][n])>max)
      {
        max = atof(matrix[m][n]);
        c1 = m;
        c2 = n;
      }
    }
  }

  //Terminate clustering process once the terminating condition has been reached
  if (max < v_stop_cluster)
    break;
  //Number of observations in cluster 1
  n_c1 = clusters[c1].clustcount;
  //Number of observations in cluster 2
  n_c2 = clusters[c2].clustcount;
  //Similarity coefficient between cluster 1 and 2
  d_c12 = max;
  max = 0;
  //Calculate the similarity between newly formed cluster 12 and all other clusters
  for (r=0;r<c1;r++)
  {
    if(clusters[r].deleted == 0)
    {
      n_cr = clusters[r].clustcount;
      d_c1r = atof(matrix[c1][r]);
      if (c2>=r)
        d_c2r = atof(matrix[c2][r]);
      else
        d_c2r = atof(matrix[r][c2]);

      d_c12r = (((double)n_cr+(double)n_c1)/((double)n_cr+(double)n_c1+(double)n_c2))*d_c1r +
               (((double)n_cr+(double)n_c2)/((double)n_cr+(double)n_c1+(double)n_c2))*d_c2r -
               (((double)n_cr)/((double)n_cr+(double)n_c1+(double)n_c2))*d_c12;

      sprintf(buffer,"%.4f",d_c12r);
      strcpy(matrix[c1][r],buffer);
    }
  }
  //Calculate the similarity between newly formed cluster 12 and all other clusters
  for (r=c1+1;r<i;r++)
  {
    if(clusters[r].deleted == 0)
    {
      n_cr = clusters[r].clustcount;
      d_c1r = atof(matrix[r][c1]);
      d_c2r = atof(matrix[r][c2]);

      d_c12r = (((double)n_cr+(double)n_c1)/((double)n_cr+(double)n_c1+(double)n_c2))*d_c1r +
               (((double)n_cr+(double)n_c2)/((double)n_cr+(double)n_c1+(double)n_c2))*d_c2r -
               (((double)n_cr)/((double)n_cr+(double)n_c1+(double)n_c2))*d_c12;

      sprintf(buffer,"%.4f",d_c12r);
      strcpy(matrix[r][c1],buffer);
    }
  }

  //Update the similarity matrix by removing similarity measures between the
  //merged and other clusters
  for (r=c2;r<i;r++)
  {
    strcpy(matrix[r][c2],"0.0000");
  }
  for (r=0;r<=c2;r++)
  {
    strcpy(matrix[c2][r],"0.0000");
  }
  //Delete cluster 2 and add newly formed cluster and cluster 1
  clusters[c1].distance = d_c12;
  clusters[c2].deleted = 1;
  clusters[c1].clustcount = clusters[c1].clustcount + clusters[c2].clustcount;
  strcat(clusters[c1].clustid_concat,",");
  strcat(clusters[c1].clustid_concat,clusters[c2].clustid_concat);
}
```

A.5 Outlier Analysis

The following Oracle PL/SQL procedure calculates the Mahalanobis distance (described in §2.7) given as

$$D^2(\mathbf{X}, \overline{\mathbf{X}}) = (\mathbf{X} - \overline{\mathbf{X}})' \Sigma^{-1} (\mathbf{X} - \overline{\mathbf{X}}),$$

where Σ is the covariance matrix for the explanatory variables, $\mathbf{X}$ is the vector of explanatory variables for all observations in the population, and $\overline{\mathbf{X}}$ is the vector of corresponding means, taken over all observations in the population. The source code may be found on the attached compact disk, in the directory `\Outlier\`.

```
/*Declare variables*/
v_prepaid_ind NUMBER;
v_subscriber_tariff NUMBER;
v_peak_ind NUMBER;
v_cell_location NUMBER;
v_other_party NUMBER;
v_charge NUMBER;
v_duration NUMBER;
x_1 NUMBER;
x_2 NUMBER;
x_3 NUMBER;
x_4 NUMBER;
x_5 NUMBER;
x_6 NUMBER;
x_7 NUMBER;
v_mahalanobis NUMBER;
BEGIN
  /*Loop though the normalised legitimate observations and calculate the Mahalanobis distance
    between the observation and population mean*/
  FOR i IN 1..v_max_observation_id LOOP

    /*Calculate the difference between the explanatory variables values and the population mean*/
    SELECT A.prepaid_ind - v_prepaid_ind_mean,
           A.subscriber_numerated_tariff - v_numerated_tariff_mean,
           A.peak_ind - v_peak_ind_mean,
           A.cell_location_numerated - v_location_numerated_mean,
           A.other_party_country_numerated - v_country_numerated_mean,
           A.call_charge - v_call_charge_mean,
           A.call_duration - v_call_duratio_mean
      INTO v_prepaid_ind,
           v_subscriber_tariff,
           v_peak_ind,
           v_cell_location,
           v_other_party,
           v_charge,
           v_duration
      FROM normalised_data_set A
     WHERE A.observation_id = i;

  BEGIN
    /*Multiply the vector of differences between the explanatory variables values and the population mean
      with the inverse covariance matrix*/
    x_1 := (v_prepaid_ind*v_inv_covar_11)+(v_subscriber_tariff*v_inv_covar_12)+(v_peak_ind*v_inv_covar_13)+
           (v_cell_location*v_inv_covar_14)+(v_other_party*v_inv_covar_15)+(v_charge*v_inv_covar_16)+
           (v_duration*v_inv_covar_17);
    x_2 := (v_prepaid_ind*v_inv_covar_21)+(v_subscriber_tariff*v_inv_covar_22)+(v_peak_ind*v_inv_covar_23)+
           (v_cell_location*v_inv_covar_24)+(v_other_party*v_inv_covar_25)+(v_charge*v_inv_covar_26)+
           (v_duration*v_inv_covar_27);
    x_3 := (v_prepaid_ind*v_inv_covar_31)+(v_subscriber_tariff*v_inv_covar_32)+(v_peak_ind*v_inv_covar_33)+
           (v_cell_location*v_inv_covar_34)+(v_other_party*v_inv_covar_35)+(v_charge*-v_inv_covar_36)+
           (v_duration*v_inv_covar_37);
    x_4 := (v_prepaid_ind*v_inv_covar_41)+(v_subscriber_tariff*v_inv_covar_42)+(v_peak_ind*v_inv_covar_43)+
           (v_cell_location*v_inv_covar_44)+(v_other_party*v_inv_covar_45)+(v_charge*v_inv_covar_46)+
           (v_duration*v_inv_covar_47);
    x_5 := (v_prepaid_ind*v_inv_covar_51)+(v_subscriber_tariff*v_inv_covar_52)+(v_peak_ind*v_inv_covar_53)+
           (v_cell_location*v_inv_covar_54)+(v_other_party*v_inv_covar_55)+(v_charge*v_inv_covar_56)+
           (v_duration*v_inv_covar_57);
    x_6 := (v_prepaid_ind*v_inv_covar_61)+(v_subscriber_tariff*v_inv_covar_62)+(v_peak_ind*v_inv_covar_63)+
           (v_cell_location*v_inv_covar_64)+(v_other_party*v_inv_covar_65)+(v_charge*v_inv_covar_66)+
           (v_duration*v_inv_covar_67);
    x_7 := (v_prepaid_ind*v_inv_covar_71)+(v_subscriber_tariff*v_inv_covar_72)+(v_peak_ind*v_inv_covar_73)+
           (v_cell_location*v_inv_covar_74)+(v_other_party*v_inv_covar_75)+(v_charge*v_inv_covar_76)+
           (v_duration*v_inv_covar_77);

    /*Multiply the previous result with the transposed vector of differences between the
      explanatory variables values and the population mean to obtain the Mahalanobis distance*/
    v_mahalanobis := v_prepaid_ind*x_1 + v_subscriber_tariff*x_2 + v_peak_ind*x_3 + v_cell_location*x_4 +
                     v_other_party*x_5 + v_charge*x_6 + v_duration*x_7;

    /*Insert the Mahalanobis distance of each observation into the database table*/
    INSERT
      INTO data_set_mahalanobis (
           observation_id,
           mahalanobis)
    VALUES(i,
           v_mahalanobis);
  END;
END LOOP;
COMMIT;
```

A.6 Association Rule Mining

The C++ implementation of the Apriori algorithm [5] (described in §2.8) was used to mine for frequent item sets among the explanatory variables of the observations. The Apriori program executable, source code and a helpfile may be found on the attached compact disk, in the directory \Apriori\.

```
The Apriori program is invoked as follows:

apriori.exe [options] infile outfile [appfile]

The normal arguments are:

infile   file to read transactions from
outfile  file to write association rules / frequent item sets to
appfile  file stating item appearances (optional)

The possible options are:
```

```
-t#    target type (default: association rules)
       (s: itemsets, c: closed itemsets, m: maximal itemsets,
        r: association rules, h: association hyperedges)
-m#    minimal number of items per set/rule/hyperedge (default: 1)
-n#    maximal number of items per set/rule/hyperedge (default: 5)
-s#    minimal support of a set/rule/hyperedge (default: 10%)
-S#    minimal support of a set/rule/hyperedge (default: 100%)
-c#    minimal confidence of a rule/hyperedge (default: 80%)
-o     use original definition of the support of a rule (body & head)
-k#    item separator for output (default: " ")
-p#    output format for support/confidence (default: "%.1f%%")
-x     extended support output (print both rule support types)
-a     print absolute support (number of transactions)
-y     print lift value (confidence divided by prior)
-e#    additional rule evaluation measure (default: none)
-!     print a list of additional rule evaluation measures
-d#    minimal value of additional evaluation measure (default: 10%)
-v     print value of additional rule evaluation measure
-g     write output in scanable form (quote certain characters)
-l     do not load transactions into memory (work on input file)
-q#    sort items w.r.t. their frequency (default: 1)
       (1: ascending, -1: descending, 0: do not sort,
        2: ascending, -2: descending w.r.t. transaction size sum)
-u#    filter unused items from transactions (default: 0.5)
       (0: do not filter items w.r.t. usage in item sets,
       <0: fraction of removed items for filtering,
       >0: take execution times ratio into account)
-h     do not organize transactions as a prefix tree
-j     use quicksort to sort the transactions (default: heapsort)
-z     minimize memory usage (default: maximize speed)
-i#    ignore records starting with characters in the given string
-b/f/r#  blank characters, field and record separators (default: " \t\r", " \t", "\n")

(# always means a number, a letter, or a string that specifies the parameter of the option.)
```