GLOBAL OPTIMIZATION APPLIED TO CHURN MODELS

by

Kim Talbot

179587 (M)

A Dissertation Submitted in Partial Fulfillment of the Requirements

For the Degree of Bachelor of Science (Honours)

Statistics & Operations Research as main area

DEPARTMENT OF STATISTICS & OPERATIONS RESEARCH

FACULTY OF SCIENCE

UNIVERSITY OF MALTA

7th MAY 2010

To my family and fiancé

for your love and support throughout my years of study

I would like to thank Dr. J. Sklenar for all his help and support all through this dissertation. His encouragement, dedication and contributions were of utmost importance. I would also like to thank Mr. K. Galea and Ms. G. Xuereb from Vodafone (Malta) Ltd. for providing the data for this dissertation.

Abstract

Kim Talbot, B.Sc. (Hons.)

Department of Statistics & Operations Research

May 2010

University of Malta

This dissertation examines the probability that a subscriber churns from the current tariff he is subcribed to. These probabilities differ from one churn model to another and the optimal churn probabilities will be found by a global optimization algorithm and a standard optimization algorithm. When the optimal probabilities are obtained, a prediction of five or eight weeks is calculated, depending on the churn model. These predictions will then show which of the churn models implemented is the most accurate. In fact, the shifted-beta geometric (sBG) model is the most accurate and moreover, the global optimization algorithm performs better than the standard optimization algorithm. Modelling is done by use of Microsoft Excel and Matlab.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the telecommunications industry has reached a level of saturation. Even though the number of mobile service subscriptions has been increasing, the rate of this increment is not as fast as it used to be. Companies investing in this sector are now faced with the problem of churning due to the increase in competition. They tend to forget to attend to the needs and expectations of their present subscribers, thus increasing the risk of churn considerably. So nowadays, companies are trying to offer the best tariffs to retain their current subscribers and offer new attracting tariffs to acquire new ones.

To retain the current subscribers within a company, it would be useful to have an idea when a subscriber is most likely to churn. Conventional statistical methods, such as decision trees and neural networks, are very successful in predicting which customers might churn. However, these methods could hardly predict when customers will churn, or how long the customers will stay active [13].

1.1 Purpose of Dissertation

This dissertation examines the probability that a subscriber churns from the current tariff he is subcribed to. These probabilities differ from one churn model to another and the optimal churn probabilities will be found by a global optimization algorithm, namely the simulated annealing algorithm. Its background theory and performance are examined to demonstrate the benefits when compared with standard optimization algorithms. When the optimal probabilities are obtained, a prediction of five or eight weeks, depending on the churn model, is calculated. These predictions will then show which of the churn models implemented is the most accurate. Modelling is done by use of Microsoft Excel and Matlab.

1.2 Structure of Dissertation

Chapter 2 gives a brief overview of research in optimization, simulated annealing and churn models. Chapter 3 gives a deeper mathematical explanation of the global optimization algorithm, Simulated Annealing, showing how this algorithm converges to the global minimum/maximum, and gives necessary and sufficient conditions for convergence. Different churn models used in various areas are discussed in Chapter 4. In particular, the Cox model, variations of this model and the Shifted-Beta Geometric model. The data for this dissertation is provided by Vodafone (Malta) Ltd., and consists of the number of pre-paid subscribers in two different tariffs, the It's Good to Talk (IGTT) tariff and the Friends tariff. Several churn models will be applied to this data to optimize the model parameters. The resultant optimal model parameters will then be used for predictions, and a comparison of the actual data with the predictions will be made to test the accuracy of each model, as discussed in Chapter 5.

Chapter 2

Review of the Literature

2.1 Global Optimization Algorithms

In real-life problems, functions of many variables have a large number of *local* minima and/or maxima. By using local optimization algorithms, it is relatively easy to find an arbitrary local optimum. A local optimum is a solution which is optimal within a neighbouring set of solutions. Finding the global maximum/minimum of a function is more complex. A global optimum is the optimal solution among all possible solutions.

The objective of global optimization is to find the globally best solution of possibly nonlinear models, in the presence of multiple local optima. Nonlinear models are present in many applications, such as in advanced engineering design, biotechnology, data analysis, environmental management, financial planning, process control, risk management, scientific modelling, and others. Their solution often requires a global search approach. [20] describes six different heuristic strategies for convergence to global optima:

1. Globalized extensions of local search methods: The idea of these methods is to apply a preliminary grid search or random search based global phase, followed by

applying a local convex programming method.

2. Evolution strategies: These methods adapt a search procedure based on a population of candidate solution points. Iterations involve a competitive selection that drops the poorer solutions. The remaining pool of candidates are then recombined with other solutions, for example, by swapping components with another.

3. Simulated Annealing: These techniques are based upon the physical analogy of cooling crystal structures that spontaneously attempt to arrive at some stable (global) equilibrium.

4. Tabu search: The idea of this search is to forbid search moves to points already visited in the search space, at least for the upcoming few steps. That is, one can temporarily accept new inferior solutions, in order to avoid paths already investigated. This approach can lead to exploring new regions of the feasible set, with the goal of finding a solution by globalized search. Tabu search has traditionally been applied to combinatorial optimization problems.

5. Approximate convex global underestimation: This strategy attempts to estimate the large scale, overall convexity characteristics of the objective function based on directed sampling in the feasible set.

6. Sequential improvement of local optima: These methods usually operate on adaptively constructed auxiliary functions, to assist the search for gradually better optima.

In this dissertation Simulated Annealing is the strategy that will be implemented.

2.2 Tariff Optimization

In the telecommunications industry, each and every subscriber is associated to one particular tariff at a time. The two simplest forms of tariffs are the *prepaid tariff* and the *postpaid/contract tariff*. The difference between these two tariffs is that for the prepaid tariff, subscribers buy a 'top-up card' before making use of the service, while for contract subscribers, subscribers make use of the service before making use of the service. Usually, at the end of each month a bill is sent to these subscribers to pay for the service consumed. A subscriber can switch from one tariff to another, as long as the subscriber is making use of only one tariff.

[22] presents a practical problem of determining an optimal tariff for a subscriber of a mobile telecommunications company. It explains how a company tries to offer the best possible combination of services in different contracts to satisfy as much as possible the subscribers. Services are distributed into several contracts with a fixed monthly payment. However, a customer is charged accordingly if he makes use of more services than the proposed services in the contract.

2.3 Simulated Annealing

The simplest form of optimization problems usually deal with one single objective function having a linear objective function and linear constraints. In general, this

form of optimization problem can be written as:

$$\begin{aligned}
\min / \max\ Z &= c_1x_1 + c_2x_2 + \ldots + c_Nx_N \\
\text{s.t. } a_{11}x_1 &+ a_{12}x_2 + \ldots + a_{1N}x_N \leq b_1 \\
a_{21}x_1 &+ a_{22}x_2 + \ldots + a_{2N}x_N \leq b_2 \\
&\vdots \\
a_{M1}x_1 &+ a_{M2}x_2 + \ldots + a_{MN}x_N \leq b_M
\end{aligned}$$

By using adequate techniques or software, it is not difficult to find a direct solution which maximizes or minimizes the objective function. Moreover, the solution obtained guarantees a global optimum, that is, the highest or lowest value from the objective function. However, in many real life cases, optimization problems have more than one objective function. These problems are known as *multiple-objective/multi-objective* simulation optimization problems. A multiple-objective simulation optimization problem is of the form:

$$\begin{aligned}
\min / \max\ Z_1 &= c_{11}x_1 + c_{12}x_2 + \ldots + c_{1N}x_N \\
\min / \max\ Z_2 &= c_{21}x_1 + c_{22}x_2 + \ldots + c_{2N}x_N \\
&\vdots \\
\min / \max\ Z_S &= c_{S1}x_1 + c_{S2}x_2 + \ldots + c_{SN}x_N
\end{aligned}$$

Unlike optimization problems with a single objective function, multi-objective optimization problems do not converge to a unique solution. Apart from that, improvement in the convergence made with respect to one objective function, may lead to a deviation from one or more other objective functions. In this case an adjustment must be made

in order to obtain an acceptable global optimal solution. The criteria to define this adjustment varies from one problem to another and so it cannot be determined as a general case.

In many multi-objective simulation optimization problems, the objective function is obtained from a simulation model with more than one output variable as an optimization objective. A problem which arises with these optimization problems is that the simulation model cannot be expressed as an exact and deterministic mathematical expression, and therefore they cannot be solved using direct methods [3].

Simulated Annealing (SA) is a meta-heuristic technique that has proved to be effective as a solving solution for simulation optimization problems. This algorithm tries to find an optimum solution that satisfies all objective functions simultaneously according to a specific criteria which must be determined beforehand. The simulated annealing method involves searching and evaluating a set of feasible solutions [3]. It tries to avoid convergence to local optimum solutions in the early stages of the algorithm. In fact, this is obtained by allowing solutions in a neighbourhood which have a lower optimal value than the previously evaluated result. The probability of accepting such solutions is calculated from a mathematical function called the *acceptance function.*

For example, if a lower quality solution X' is compared with another solution X from its neighbourhood, with a variation in the objective function $C'-C$, the simulated annealing algorithm still explores the neighbourhood of the lower quality solution X' if the acceptance function is satisfied.
Assuming minimization, the acceptance criterion can be written as:

$$exp\left[\frac{(C'-C)}{T}\right] < R$$

where T is a control parameter and R is an independent, identically distributed random

number in the range $[0, 1]$.

In order for the algorithm to choose less frequently neighbourhoods of lower quality solutions as the number of iterations increase, the parameter T is chosen such that it decreases with time, so that the chance of converging to a local optimal solution in the first few iterations is eliminated. The relation between the control parameter T and time is called the *cooling curve.*

At every step of the algorithm, the evaluation of the objective functions can result in either that all objectives improve, or that all objectives get worse, or that some improve while others get worse. In the first case it is clear that the last solution obtained is better than the previous one, and so this solution is retained for the next iteration. Similarly, if all objectives get worse, the last solution must be evaluated by some other acceptance function. The more complex case is when some objectives improve and others get worse. In this case, a decision whether the solution is to be retained or not must be made, or else, whether to evaluate the objectives using another acceptance function.

A modified simulated annealing algorithm is proposed in [3] which is designed in such a way that it guides the search in order to satisfy all objectives simultaneously. This includes more than one cooling curves, in particular, one global cooling curve and one particular cooling curve for each objective function. This method decides which of the multiple objectives should become a reference objective by introducing a *selection function.* This function indicates which objective will be treated as reference objective whenever the third case of the evaluation step is obtained.

Another approach to global stochastic simulation optimization, combines stochastic approximation with simulated annealing. Stochastic approximation directs a search of the response surface efficiently, using a conservative number of simulation replications to

approximate the local gradient of a probabilistic loss function. Simulated annealing adds a random component, a Monte Carlo randomness term, to the stochastic approximation search, which is needed to avoid local optima [12].

Another variant of the simulated annealing algorithm for solving discrete stochastic optimization problems, where the objective function is stochastic, can be evaluated through Monte Carlo simulations. In particular, the Metropolis criterion depends on whether the objective function values indicate statistically significant difference at each iteration based on confidence intervals associated with these values. To the contrary of the original simulated annealing algorithm, this method uses a constant control parameter T, and the first m iterations converge almost surely to a global optimizer [2].

2.4 Churn Modelling

2.4.1 Churn

Churn is the term used to represent the action that a subscriber abandons the service from his current service provider. In many research papers, churn is divided into two categories: involuntary churn and voluntary churn. Involuntary churn is when churn is initiated by the company itself and it is the least common of the two. Involuntary churn occurs when the subscriber is disconnected from the service. The grace period is the term used to refer to the time when the subscriber is allowed to receive the service even though the credit amount has expired. Failure to recharge credit within the grace period or right after will result in disconnection from the service provider. Another reason for involuntary churn is if the subscriber is found to be making some sort of fraudulent usage. On the other hand, voluntary churn is when churn is initiated by the

subscriber himself and voluntary churn is considered to be more complex with much more various reasons why a subscriber decides to churn. Some reasons for voluntary churn are dissatisfaction from the current service provider (for example lack of service from customer care), changes in geographic locations (for example when migrating to a different country it makes more sense to switch to a service provider in the other country), and finally, switching to another competitor when competitor's promotion attracts the subscribers. More reasons may arise however the ones mentioned are the most frequent reasons for churning.

In [24] churn is further divided into *financial/non-financial churn*, where financial churn is defined as bad-debt subscribers who churn, while non-financial churn refers to paying subscribers who churn.

2.4.2 Churn Rate

Churn rate is the number of subscribers who disconnect their use of a service, divided by the average number of total subscribers within a particular company. The average number of total subscribers is just an estimate since it is difficult to calculate the exact total number of subscribers when considering a large company. Churn rate helps the service providers gain knowledge of the growth or decline of the subscriber base and gives a hint of the average length of participation in the service. In the telecommunications industry, a level of saturation has been reached in practically all over the world. Even though in the past years there was an increase in subscriptions, churn rates have been increasing, mostly due to an increase in competition. Companies are now investing in efficient churn models to help them predict churn and keep a stable, low churn rate as much as possible.

2.4.3 Churn Management

[8] divides the possible causes of churn into four different components namely:

1. Static component: is the behaviour of the subscribers within a particular company and what type of tariff or contract a subscriber is subscribed to.

2. Dynamic component: is the contract made between a subscriber and the customer care service provided by the company.

3. Seasonal component: more related to contract bound subscribers where the deactivation date of a contract corresponds to the time at which a subscriber churns.

4. External component: referring to the influence from other competitors' advertisments which might attract subscribers.

Retaining subscribers is one of the most critical challenges in the maturing mobile telecommunications service industry. Telecom operators stand to lose a great deal in price premium, decreasing profit levels and a possible loss of referrals from continuing service subscribers. Figuring how to deal with churn is turning out to be the key to the survival of telecom organizations [15]. Companies are now interested in predicting those subscribers who are most likely to churn, and if possible *when* will these subscribers churn. This is important so that they plan strategies to either retain their subscribers, since often they find out too late that a subscriber is going to churn, or else try to acquire new subscribers. For postpaid subscriptions it is slightly more easy to predict churn since usually the time at which a subscriber churns is equal to the deactivation time of the contract. However, for prepaid subscriptions the time at which a subscriber churns varies considerably. Despite the best efforts of these companies to prevent churn, the company will lose some of its subscribers to the competition sooner or later and

try to win them back by running reacquisition strategies since subscriber acquisition is relatively more easy. Three fundamental strategic approaches discussed in [15] are:

1. Ignoring loss of subscribers and trying harder to acquire new subscribers as replacements
2. Trying to steal subscribers from competitors to make up for the losses
3. Building customer churn management capabilities

2.4.4 Traditional Churn Models

Since large companies have been investing in predicting churn, a number of models have been discussed over the years. Statistical models typically used to predict churn are based on logistic regression or classification trees (CART) [8] and survival analysis models. Most of the models classify data according to predictive accuracy (being able to identify correctly those individuals that will become churners during the evaluation phase) [8]. An important difference between survival prediction models and other prediction models is the fact that survival analysis models, model time-dependent data. Thus logistic regression and classification trees may help to model which subscribers are at a high risk to churn however they lack information about *when* will these subscribers churn. Survival analysis models are hence considered to be more efficient models which help estimate churn especially when the knowledge of *when* subscribers will churn is of utmost importance. Apart from that, survival analysis is most commonly used when dealing with censored data (discussed later).

2.4.5 Life Time Value

Life Time Value (LTV) is mainly used when trying to model long-term customer satisfaction. LTV is the present value of the future cash flows attributed to the customer relationship. A definition of terms associated with LTV should be mentioned before stating the formula used to calculate the LTV.

- **Churn rate** is the number of subscribers who disconnect their use of service divided by the average number of total subscribers.
- **Retention Rate** is the complement of Churn rate and is given by

$$(1 - \text{Churn Rate}) = \textbf{Retention Rate}$$

- **Profit Margin** is the percentage of the net profit.
- **Discount Rate** is the cost of capital used to discount future revenue from a subscriber.
- **Period** is the unit of time into which a subscriber relationship is divided for analysis.

Consider a subscriber who generates a margin m_t for each period t with discount rate i and probability of retention rate r [8]. The Life Time Value is given by:

$$LTV = \sum_{t=0}^{\infty} \frac{m_t r_t}{(1+i)^t}$$

A firm acquires n_0 subscribers at time 0 at an acquisition cost of c_0 per subscriber. Over time, subscribers defect (churn) so that the firm is left with $n_0 \times r$ subscribers at

the end of period 2, and so on. So in general, the LTV for the k^{th} cohort at time 0 is given [8] by

$$LTV_k = \frac{n_k}{(1+i)^k} \sum_{t=k}^{\infty} \frac{m_{t-k} r^{t-k}}{(1+i)^{t-k}} - \frac{n_k c_k}{(1+i)^k}$$

In general, an LTV model is made up of three components:

1. Customer value over time - $v(t)$ for time $t > 0$.

2. Customer length of service - is usually given by a survival function which gives the probability that a subscriber will be active at time t, where a subscriber is said to be *active* if this subscriber is currently making use of the service.

3. Discounting factor - $D(t)$ which describes the profit made in some future time t.

Let $f(t)$ be the subscribers instantaneous probability of churn at time t such that $f(t) = -\frac{dS}{dt}$, where $S(t)$ is the survival function. Given the three components, $v(t)$, $S(t)$ and $D(t)$, the explicit formula for a subscriber's LTV is given by,

$$LTV = \int_0^{\infty} S(t)\, v(t)\, D(t)\, dt$$

So in other words, LTV is used to find the total profit gained while the customer is still active.

2.4.6 Survival Analysis Models

As mentioned previously, survival analysis is convenient when dealing with censored time-dependent data. In a survival analysis model, an *origin of time* must be established to indicate the starting time at which the data will be observed. When the origin of time and end of the origin of time is established, there is a possibility that an event

(churn), is not registered since it does not occur during the period of observation. This is referred to as censoring. There can be three different types of censoring:

- Right censoring
- Left censoring
- Interval censoring

In real life situations, right censoring is the most common. This takes place when the time that an event (churn) occurs, happens after the end time of the observation period. Similarly, left censoring takes place when the time that an event occurs, happens before the start time of the observation period. In interval censoring, the event occurs in an interval during the observation period, however, the exact time that the event happens cannot be determined. Clearly, an event that occurs during the period of observation with the exact time known has no censoring. These types of censoring are illustrated in Figure 2.1.

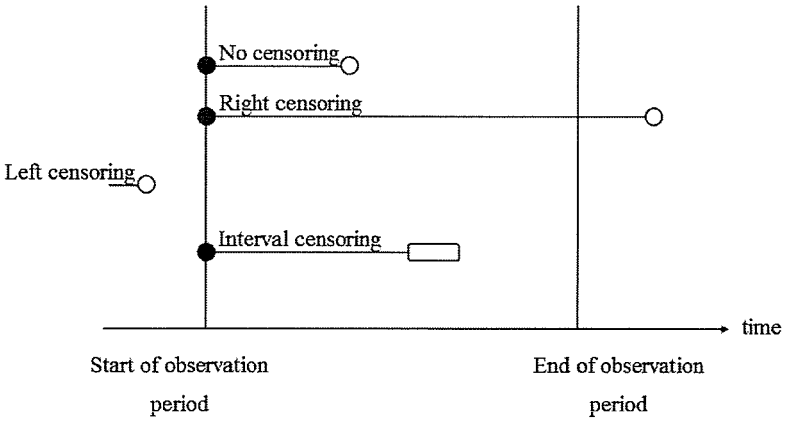

Figure 2.1: Different types of censoring

If $T \geq 0$ is the random variable denoting the time at which an event occurs, the density function is given by $f(t)$ and the distribution function is given by $F(t)$, then the survival function $S(t)$, is given by

$$S(t) = P[T > t] = 1 - F(t) = \int_t^{\infty} f(u)\, du$$

where $S(t)$ is a monotone decreasing function from 1 to 0 with $S(0) = 1$. This represents the probability that an observed customer will survive up to time t.

The hazard function (or the instantaneous failure rate) $\lambda(t)$, gives the rate at which a customer fails to survive up to time t. For the hazard function, the interval of time is taken to be smaller and smaller until the interval becomes infinitely small, Δt. The hazard function is defined as

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P[t < T < t + \Delta t | T > t]}{\Delta t} = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{S(t)}$$

(Note: Derivation of the hazard function given in Chapter 4.)

The Cox model became the most used procedure for modelling the relationship of covariates to a survival or other censored outcomes. Its form is flexible enough to allow time-dependent covariates, however it has some restrictions. One of the restrictions of using a Cox model with fixed time is its proportional hazards assumption, that is, that the hazard ratio between two covariate values has to be constant over time. This is due to the common baseline hazard function cancelling out in the ratio of the two hazards [8].

Chapter 3

Simulated Annealing

Optimization has been introduced in various areas such as, engineering, operations research, computer science and communication. Combinatorial optimization is one of the major subfields of optimization, which tries to find an optimal solution out of a set of feasible solutions. In general, a combinatorial optimization problem can be expressed as a pair $(\mathcal{R}, C)$, where $\mathcal{R}$ is the finite - or possibly countably infinite - set of configurations (configuration space) and C is a cost function, such that $C : \mathcal{R} \rightarrow \mathbb{R}$ assigns a real number to each configuration. The configurations and cost functions vary according to the particular optimization problem one is trying to optimize. Assuming a minimization problem, the aim is to find a configuration for which C takes a minimum value. In other words, an optimal configuration, i_0, must be found such that it satisfies

$$C_{opt} = C(i_0) = \min_{i \in \mathcal{R}} C(i)$$

where C_{opt} is the optimum minimum cost. When trying to solve an optimization problem, one can use either an optimization algorithm or an approximation algorithm. The

difference between the two is that an optimization algorithm finds a globally optimal solution, while an approximation algorithm finds an approximate solution. The simulated annealing algorithm is a mixture of both types of algorithms, since it is able to find approximate optimal solutions.

3.1 Introduction of the algorithm

Annealing of solids is widely used in physics and it is the process of heating solid metal to a maximum temperature such that it reaches thermal equilibrium and cooling it slowly so that its particles arrange themselves into a defined lattice (ordered set). When the solid reaches thermal equilibrium, the probability that a temperature with value T, is in a state (condition of an object in the system at a particular time) with energy E, is given by the Boltzmann distribution

$$\mathbb{P}\left[state = E\right] = \frac{1}{Z\left(T\right)} \exp\left(-\frac{E}{k_B T}\right) \tag{3.1}$$

where T is the temperature measured in Kelvin, $Z\left(T\right)$ is a normalization factor (partition function) depending on the temperature T and k_B is the Boltzmann constant such that $k_B = 1.380650524 \times 10^{-23} J/K$. The factor $\exp\left(-\frac{E}{k_B T}\right)$ is referred to as the Boltzmann factor.

The Boltzmann factor clearly shows that in the cooling phase, as the temperature T decreases, the Boltzmann distribution approaches the states with lowest energy. In particular, as $T \to 0$, only the minimum energy states have a non-zero probability of occurence. To simulate the evolution to thermal equilibrium of a solid for a fixed value of the temperature T, a Monte Carlo method was introduced which generates sequences of states of the solid [27]. Suppose a small randomly generated perturbation

(slight deviation) is applied to the current state of the solid. Then the difference in energy between the current state and the perturbed state is ΔE. If $\Delta E < 0$, then the process is continued using this new perturbed state. Otherwise, if $\Delta E \geq 0$, the probability of accepting the perturbed state is

$$\exp\left(-\frac{\Delta E}{k_B T}\right)$$

This acceptance rule for the new perturbed state is referred to as the *Metropolis criterion* [27]. After reaching thermal equilibrium, the probability distribution of the perturbed states approaches the Boltzmann distribution given by Equation 3.1. This Monte Carlo method is known as the Metropolis algorithm. When using the Metropolis algorithm to generate sequences of configurations of a combinatorial problem, the cost function C and the control parameter c are used instead of energy and temperature respectively. Given a configuration i, another configuration j can be obtained by choosing at random an element from the neighbourhood of i [27]. The configuration j corresponds to the slightly perturbed state. Let the difference between the cost functions of configuration i and configuration j be given by ΔC_{ij} such that $\Delta C_{ij} = C(j) - C(i)$. If $\Delta C_{ij} \leq 0$, then the probability for configuration j to be the next configuration in the sequence is 1. Otherwise, if $\Delta C_{ij} > 0$, the probability for configuration j to be the next configuration in the sequence is given by the Metropolis criterion, which in this case is,

$$\exp\left(-\frac{\Delta C_{ij}}{c}\right) \tag{3.2}$$

Thus, the Boltzmann distribution is given by

$$P\left[configuration = i\right] = \frac{1}{Q\left(c\right)} \exp\left(-\frac{C\left(i\right)}{c}\right) \tag{3.3}$$

where $Q\left(c\right)$ is a normalization factor depending on the control parameter c.

The algorithm lowers the value of the control parameter c until thermal equilibrium is reached and the algorithm is then terminated for some small value of c, for which it cannot be lowered further. The acceptance criterion is evaluated by comparing a random number from a uniform distribution on $[0, 1)$ to Equation 3.2.

A simplified pseudo-code for the algorithm is given below

repeat

perturb (configuration $i \rightarrow$ configuration j), compute ΔC_{ij};

if $\Delta C_{ij} \leq 0$ then *accept* else

if $\exp\left(-\frac{\Delta C_{ij}}{c}\right) >$ random $[0, 1)$ then *accept*;

if *accept* then **update** (configuration j);

until **equilibrium is approached sufficiently closely**;

Simulated Annealing works by means of searching and evaluating a set of feasible solutions, reducing the possibility of finding a solution that might turn out to be a local optimum. This means it avoids converging to a local optimum solution at early stages of the search [3].

3.2 Mathematical model of the algorithm

Simulated Annealing is an algorithm that continuously attempts to transform the current configuration into one of its neighbours. This is best described by means of a Markov chain [27].

Definition 1. *A **Markov chain** is a collection of random variables* $\{X_t\}, t = 0, 1, 2, \ldots$ *having the property that given present state, the future state is conditionally independent of the past state, such that*

$$P[X_t = y | X_0 = x_0, X_1 = x_1, \ldots, X_{t-1} = x_{t-1}] = P[X_t = y | X_{t-1} = x_{t-1}]$$

if $P[X_0 = x_0, X_1 = x_1, \ldots, X_{t-1} = x_{t-1}] \neq 0$.

Let $a_i(k)$ denote the probability outcome i at the k-trial. Then $a_i(k)$ is given by

$$a_i(k) = \sum_l a_l(k-1) P_{li}(k-1, k), k = 1, 2, \ldots,$$

where the sum is taken over all possible outcomes. Let $\mathbf{X}(k)$ denote the outcome of the k-th trial, such that

$$P_{ij}(k-1, k) = P[\mathbf{X}(k) = j | \mathbf{X}(k-1) = i]$$

and

$$a_i(k) = P[\mathbf{X}(k) = i] \tag{3.4}$$

The changes of state of the system are called transitions, and the probabilities associated with various state-changes are called transition probabilities. In the case of Simulated Annealing, the Markov chain is described by a set of conditional probabili-

ties $P_{ij}(k-1,k)$ for each pair of outcomes (i,j). Then $P_{ij}(k-1,k)$, which represents the transition probability, is the probability that the k^{th} transition is a transition from configuration i to configuration j, and $\mathbf{P}(k-1,k)$ is an $|\mathcal{R}| \times |\mathcal{R}|$ matrix called the transition matrix. The transition probabilites depend on the value of the control parameter c, such that, if c is constant, the corresponding Markov chain is homogeneous and its transition matrix $\mathbf{P} = \mathbf{P}(c)$ is defined as,

$$P_{ij}(c) = \begin{cases} G_{ij}(c) A_{ij}(c) & \text{when } j \neq i \\ 1 - \sum_{l=1, l\neq i}^{|\mathcal{R}|} G_{il}(c) A_{il}(c) & \text{when } j = i \end{cases} \tag{3.5}$$

$G_{ij}(c)$ and $A_{ij}(c)$ are two conditional probabilities, where $G_{ij}(c)$, is the generation probability of generating configuration j from configuration i, and $A_{ij}(c)$, is the acceptance probability of accepting configuration j, once it has been generated from i. The corresponding matrices $\mathbf{G}(c)$ and $\mathbf{A}(c)$ are called the generation and acceptance matrix respectively. Then by Equation 3.5, $\mathbf{P}(c)$ is a stochastic matrix such that, $\forall\, i$, $\sum_j P_{ij}(c) = 1$ [27].

Definition 2. *A **Stochastic matrix** is a square matrix with non-negative entries whose rows sum to 1.*

Since the algorithm lowers the value of the control parameter c, two formulations of the algorithm arise. These are the homogeneous algorithm and the inhomogeneous algorithm. The homogeneous algorithm is described by a sequence of homogeneous Markov chains where each Markov chain is generated at a fixed value of c and c is decreased in between subsequent Markov chains. The inhomogeneous algorithm is described by a single inhomogeneous Markov chain where the value of c is decreased in between subsequent transitions [27].

Definition 3. *A **Homogeneous Markov chain** is a process where*

$$P\left[X_{t+h} = y | X_t = x\right] = P\left[X_h = y | X_0 = x\right] \ \forall \ t, h > 0$$

*and otherwise called **Inhomogeneous Markov chain.***

The aim of the Simulated Annealing algorithm is to obtain a global minimum. So, after a large number of transitions K, the following probability must be satisfied,

$$P\left[\mathbf{X}\left(K\right) \in \mathcal{R}_{opt}\right] = 1$$

where $\mathcal{R}_{opt}$ is the set of globally minimal configurations and $\mathbf{X}\left(K\right)$ is the configuration obtained after k transitions.

3.3 Asymptotic Convergence Results

3.3.1 The Homogeneous Algorithm

The convergence to global optima for the homogeneous algorithm is based on certain conditions about the stationary distribution. The stationary distribution is the limiting distribution in a Markov chain, such that it gives the probability distribution of the configurations after an infinite number of transitions. Suppose that the stationary distribution is given by a vector $\mathbf{q}$, where the i^{th} component, q_i, is given by

$$q_i = \lim_{k \to \infty} P\left[\mathbf{X}\left(k\right) = i | \mathbf{X}\left(0\right) = j\right] \tag{3.6}$$

for an arbitrary j. Suppose that $\mathbf{X}\left(k\right) = i$ and $\mathbf{X}\left(0\right) = j$ are independent. Then by independence, $P\left[\mathbf{X}\left(k\right) = i | \mathbf{X}\left(0\right) = j\right] = P\left[\mathbf{X}\left(k\right) = i\right]$. Furthermore, by Equation 3.4

and Equation 3.6, q_i is given by

$$q_i = \lim_{k \to \infty} P\left[\mathbf{X}(k) = i\right] = \lim_{k \to \infty} \mathbf{a}(0)' \mathbf{P}^k \tag{3.7}$$

where $\mathbf{a}(0)$ is the initial probability distribution, such that it satisfies

$$\forall\, i \in \mathcal{R} : a_i(0) \geq 0,\ \sum_{i \in \mathcal{R}} a_i(0) = 1$$

The algorithm follows such that as c decreases, $\mathbf{q}(c)$ converges to a uniform distribution on the set of globally minimal configurations. So conditions on the matrices $\mathbf{A}(c)$ and $\mathbf{G}(c)$ are derived such that existence of $\mathbf{q}(c)$ is guaranteed. Suppose that

$$\lim_{c \to 0} \mathbf{q}(c) = \pi \tag{3.8}$$

where π is an $|\mathcal{R}|$-vector defined by

$$\pi_i = \begin{cases} |\mathcal{R}_{opt}|^{-1} & \text{if } i \in \mathcal{R}_{opt} \\ 0 & \text{if } i \notin \mathcal{R}_{opt} \end{cases} \tag{3.9}$$

By Equation 3.7 and Equation 3.8,

$$\lim_{c \to 0} \left(\lim_{k \to \infty} P\left[\mathbf{X}(k) \in \mathcal{R}_{opt}\right] \right) = 1$$

3.3.1.1 Existence of the Stationary Distribution

Definition 4. *A Markov chain is* ***irreducible*** *if and only if for all pairs of configurations (i, j), there is a positive probability of reaching j from i in a finite number of transitions, such that $\forall\, i, j\ \exists\, n : 1 \leq n < \infty \wedge (\mathbf{P}^n)_{ij} > 0$.*

Definition 5. *A Markov chain is* ***aperiodic*** *if and only if for all configurations $i \in \mathcal{R}$, the greatest common divisor of all integers $n \geq 1$, such that $(\mathbf{P}^n)_{ii} > 0$, is equal to 1.*

The existence of the stationary distribution is assured by the following theorem which gives the necessary conditions on the vector $\mathbf{q}$.

Theorem 1. *The stationary distribution $\mathbf{q}$ of a finite homogeneous Markov chain exists if the Markov chain is irreducible and aperiodic. Furthermore, the vector $\mathbf{q}$ is uniquely determined by,*

$$\forall\, i\, :\, q_i > 0,\ \sum_i q_i = 1 \tag{3.10}$$

$$\forall\, i\, :\, q_i = \sum_j q_j P_{ji} \tag{3.11}$$

where the matrix $\mathbf{P}$ is defined by Equation 3.5.

Assuming that $\forall\, i, j, c > 0 : A_{ij}(c) > 0$, it is sufficient for irreducibility to assume that the Markov chain induced by $\mathbf{G}(c)$ is irreducible itself, so that

$$\forall\, i, j \in \mathcal{R}\ \exists\, p \geq 1\ \exists\, l_0, l_1, \ldots, l_p \in \mathcal{R} : (l_0 = i \wedge l_p = j) :$$

$$G_{l_k l_{k+1}}(c) > 0, k = 0, 1, \ldots, p-1 \tag{3.12}$$

Moreover, an irreducible Markov chain is aperiodic if

$$\forall\, c > 0\ \exists\, i_c \in \mathcal{R} : P_{i_c i_c}(c) > 0 \tag{3.13}$$

Thus for aperiodicity, it is sufficient to assume that

$$\forall\, c > 0\ \exists\, i_c, j_c \in \mathcal{R} : A_{i_c j_c}(c) < 1 \wedge G_{i_c j_c} > 0 \tag{3.14}$$

By Equation 3.14 and by the fact that $\forall\, i,j : A_{ij} \leq 1$, then

$$\begin{aligned}
\sum_{l=1,l\neq i_c}^{|\mathcal{R}|} A_{i_c l}(c)\, G_{i_c l}(c) &= \sum_{l-1,l\neq i_c,j_c}^{|\mathcal{R}|} A_{i_c l}(c)\, G_{i_c l}(c) + A_{i_c j_c}(c)\, G_{i_c j_c}(c) \\
&< \sum_{l=1,l\neq i_c,j_c}^{|\mathcal{R}|} G_{i_c l}(c) + G_{i_c j_c}(c) \\
&= \sum_{l=1,l\neq i_c}^{|\mathcal{R}|} G_{i_c l}(c) \\
&\leq \sum_{l=1}^{|\mathcal{R}|} G_{i_c l}(c) \\
&= 1
\end{aligned}$$

Thus aperiodicity holds, since

$$P_{i_c i_c} = 1 - \sum_{l=1,l\neq i_c}^{|\mathcal{R}|} A_{i_c l}(c)\, G_{i_c l}(c) > 0$$

and Equation 3.13 is satisfied.

So, the homogeneous Markov chain with conditional probabilities that satisfy Equation 3.5, has a stationary distribution if the acceptance matrix $\mathbf{A}(c)$ and generation matrix $\mathbf{G}(c)$ satisfy Equation 3.12 and Equation 3.14, given that the acceptance probabilities are

$$A_{ij}(c) = \min\left\{1, \exp\left(-\frac{\Delta C_{ij}}{c}\right)\right\} \tag{3.15}$$

3.3.1.2 Convergence of the Stationary Distribution

Suppose that for an arbitrary configuration $i \in \mathcal{R}$, the corresponding component of the stationary distribution is

$$q_i(c) = \frac{\psi(C(i), c)}{\sum_j \psi(C(j), c)}$$

where $\psi(\gamma, c)$ is a two-argument function satisfying two conditions. In particular,

$$\forall\, i \in \mathcal{R},\, c > 0 : \psi(C(i), c) > 0$$

and the *global balance condition* such that $\forall\, j \in \mathcal{R}$:

$$\sum_{i=1, i \neq j}^{|\mathcal{R}|} \psi(C(i), c)\, G_{ij}(c)\, A_{ij}(c) = \psi(C(j), c) \sum_{i=1, i \neq j}^{|\mathcal{R}|} G_{ji}(c)\, A_{ji}(c) \tag{3.16}$$

In fact, $\mathbf{q}(c)$ is the unique stationary distribution because the $q_i(c)$'s satisfy the necessary conditions. Convergence of $\mathbf{q}(c)$ is guaranteed by the following conditions

$$\lim_{c \to 0} \psi(\gamma, c) = \begin{cases} 0 & \text{if } \gamma > 0 \\ 1 & \text{if } \gamma = 0 \\ \infty & \text{if } \gamma < 0 \end{cases}$$

$$\frac{\psi(\gamma_1, c)}{\psi(\gamma_2, c)} = \psi(\gamma_1 - \gamma_2, c)$$

$$\forall\, c > 0 : \psi(0, c) = 1$$

Equation 3.12, Equation 3.14 and Equation 3.16, give the conditions required for the acceptance matrix $\mathbf{A}(c)$ and generation matrix $\mathbf{G}(c)$ such that an asymptotic convergence to a global minimum is achieved. The conditions mentioned for convergence

of the stationary distribution are sufficient to define the stationary distribution of the Markov chain, however they are not necessary as the conditions given by Equation 3.10 and Equation 3.11. Moreover, the explicit form for the stationary distribution is not straightforward. So more explicit conditions for the $q_i(c)$'s at the cost of a more restrictive set of conditions on the matrices $\mathbf{A}(c)$ and $\mathbf{G}(c)$ should be considered by making a different choice for the two-argument function $\psi(\gamma, c)$. In particular, $\psi(C(i) - C_{opt}, c)$ is taken as $A_{i_0 i}(c)$, for an arbitrary configuration $i_0 \in \mathcal{R}_{opt}$, and let $G(c)$ be independent of c.

Theorem 2. *If the two-argument function $\psi(C(i) - C_{opt}, c)$ is taken as $A_{i_0 i}(c)$ for an arbitrary configuration $i_0 \in \mathcal{R}_{opt}$ and if $G(c)$ is not depending on c, then the stationary distribution $\mathbf{q}(c)$ is given by*

$$\forall\, i \in \mathcal{R} : q_i(c) = \frac{A_{i_0 i}(c)}{\sum_{j \in \mathcal{R}} A_{i_0 j}(c)} \tag{3.17}$$

provided the matrices $\mathbf{A}(c)$ and $\mathbf{G}$ satisfy the following conditions

$$\forall\, i, j \in \mathcal{R} : G_{ji} = G_{ij} \tag{3.18}$$

$$\forall\, i, j, k \in \mathcal{R} : C(i) \leq C(j) \leq C(k) \Rightarrow A_{ik}(c) = A_{ij}(c) A_{jk}(c)$$

$$\forall\, i, j \in \mathcal{R} : C(i) \geq C(j) \Rightarrow A_{ij}(c) = 1 \tag{3.19}$$

$$\forall\, i, j \in \mathcal{R},\, c > 0 : C(i) < C(j) \Rightarrow 0 < A_{ij}(c) < 1$$

Proof

$$\begin{aligned}
\sum_j q_j(c) P_{ji}(c) &= \sum_{j \neq i, C(j) \leq C(i)} \frac{1}{N} A_{i_0 j}(c) G_{ji} A_{ji}(c) + \sum_{j \neq i, C(j) > C(i)} \frac{1}{N} A_{i_0 j}(c) G_{ji} A_{ji}(c) \\
&+ q_i(c) P_{ii}(c) \\
&= \sum_{j \neq i, C(j) \leq C(i)} \frac{1}{N} A_{i_0 i}(c) G_{ji} + \sum_{j \neq i, C(j) > C(i)} \frac{1}{N} A_{i_0 j}(c) G_{ij} + q_i(c) P_{ii}(c) \\
&= q_i(c) \sum_{j \neq i, C(j) \leq C(i)} G_{ij} + \sum_{j \neq i, C(j) > C(i)} q_j(c) G_{ij} + q_i(c) P_{ii}(c)
\end{aligned}$$

where $N = \sum_{j \in \mathcal{R}} A_{i_0 j}(c)$ and

$$\begin{aligned}
q_i(c) P_{ii}(c) &= q_i(c) \left(1 - \sum_{j \neq i, C(j) \leq C(i)} G_{ij} A_{ij}(c) - \sum_{j \neq i, C(j) > C(i)} G_{ij} A_{ij}(c) \right) \\
&= q_i(c) - q_i(c) \sum_{j \neq i, C(j) \leq C(i)} G_{ij} - \sum_{j \neq i, C(j) > C(i)} \frac{1}{N} A_{i_0 i}(c) G_{ij} A_{ij}(c) \\
&= q_i(c) - q_i(c) \sum_{j \neq i, C(j) \leq C(i)} G_{ij} - \sum_{j \neq i, C(j) > C(i)} q_j(c) G_{ij}
\end{aligned}$$

Combining these two equations gives

$$\forall\, i \in \mathcal{R} : \sum_j q_j(c) P_{ji}(c) = q_i(c)$$

Thus Equation 3.17 satisfies the necessary conditions □

Equation 3.8 is satisfied if the following condition holds

$$\forall\, i, j \in \mathcal{R} : C(i) < C(j) \Rightarrow \lim_{c \to 0} A_{ij}(c) = 0$$

since this condition and Equation 3.19 ensure convergence of the stationary distribution.

So for the homogeneous algorithm, under certain conditions on the matrices $\mathbf{A}(c)$ and $\mathbf{G}(c)$ given by Theorem 2, the Simulated Annealing algorithm converges to a global minimum with probability 1, if for each value c_l of the control parameter, where $l = 0, 1, 2, \ldots$, the corresponding Markov chain is of infinite length and the c_l eventually converge to 0 for $l \to \infty$ [27]. Hence

$$\lim_{c \to 0} \left(\lim_{k \to \infty} P\left[\mathbf{X}(k) = i\right] \right) = \lim_{c \to 0} q_i(c) = \begin{cases} |\mathcal{R}_{opt}|^{-1} & \text{if } i \in \mathcal{R}_{opt} \\ 0 & \text{if } i \notin \mathcal{R}_{opt} \end{cases} \tag{3.20}$$

3.3.2 The Inhomogeneous Algorithm

The inhomogeneous algorithm occurs when the limits in the left-hand side of Equation 3.20 are taken along a path in the (c, k) plane, such that the value of the control parameter is changed after each transition and therefore for the inhomogeneous algorithm the control parameter is given by $c = c_k$. The inhomogeneous Markov chain with transition matrix $\mathbf{P}(k-1, k)$, for $k = 0, 1, 2, \ldots$, is given by

$$P_{ij}(k-1, k) = \begin{cases} G_{ij}(c_k) A_{ij}(c_k) & \forall\, j \neq i \\ 1 - \sum_{l=1, l \neq i}^{|\mathcal{R}|} G_{il}(c_k) A_{il}(c_k) & j = i \end{cases} \tag{3.21}$$

Assume that the sequence $\{c_k\}$, for $k = 0, 1, 2, \ldots$, satisfies the conditions

$$\lim_{k \to \infty} c_k = 0 \tag{3.22}$$

$$c_k \geq c_{k+1}, \text{ for } k = 0, 1, \ldots, \tag{3.23}$$

From Equation 3.23 it is clear that the sequence $\{c_k\}$ is a decreasing sequence and it is possible for c_k to be constant for some number of transitions, corresponding to a

homogeneous Markov chain of finite length.

3.3.2.1 Sufficient Conditions for Convergence

Definition 6. *An inhomogeneous Markov chain is* ***weakly ergodic*** *if*

$\forall\, m \geq 1,\ i, j, l \in \mathcal{R}$

$$\lim_{k \to \infty} \left(P_{il}\left(m, k\right) - P_{jl}\left(m, k\right) \right) = 0$$

where the transition matrix $\mathbf{P}\left(m, k\right)$ *is defined by*

$$P_{il}\left(m, k\right) = P\left[\mathbf{X}\left(k\right) = l | \mathbf{X}\left(m\right) = i\right]$$

Weak ergodicity shows that as $k \to \infty$ the dependence of $\mathbf{X}\left(k\right)$ with respect to $\mathbf{X}\left(0\right)$ vanishes. Theorem 3 gives conditions for weak ergodicity of the inhomogeneous Markov chain.

Theorem 3. *An inhomogeneous Markov chain is weakly ergodic if and only if there is a strictly increasing sequence of positive numbers* $\{k_l\}$, *where* $l = 0, 1, 2, \ldots$, *such that*

$$\sum_{l=0}^{\infty} \left(1 - \tau_1\left(\mathbf{P}\left(k_l, k_{l+1}\right)\right)\right) = \infty \tag{3.24}$$

where $\tau_1\left(\mathbf{P}\right)$ *is the coefficient of ergodicity of an* $n \times n$*-matrix* $\mathbf{P}$ *and is defined by*

$$\tau_1\left(\mathbf{P}\right) = 1 - \min_{i,j} \sum_{l=1}^{n} \min\left(P_{il}, P_{jl}\right)$$

Definition 7. *An inhomogeneous Markov chain is* ***strongly ergodic*** *if there exists a*

vector π *which satisfies*

$$\sum_{i=1}^{|\mathcal{R}|} \pi_i = 1 \quad \forall\, i\text{:}\ \pi_i \geq 0$$

such that $\forall\, m \geq 1,\ i, j \in \mathcal{R}$*:*

$$\lim_{k \to \infty} P_{ij}(m, k) = \pi_j \tag{3.25}$$

Strong ergodicity implies convergence in distribution of the $\mathbf{X}(k)$, such that if Equation 3.25 holds, then

$$\lim_{k \to \infty} P[\mathbf{X}(k) = j] = \pi_j$$

Theorem 4 gives conditions for strong ergodicity of the inhomogeneous Markov chain.

Theorem 4. *An inhomogeneous Markov chain is strongly ergodic if it is weakly ergodic and if* $\forall\, k$ *there exists a vector* $\pi(k)$ *such that* $\pi(k)$ *is an eigenvector with eigenvalue 1 of* $\mathbf{P}(k-1, k)$, $\sum_{i=1}^{|\mathcal{R}|} |\pi_i(k)| = 1$ *and*

$$\sum_{k=0}^{\infty} \sum_{i=1}^{|\mathcal{R}|} |\pi_i(k) - \pi_i(k+1)| < \infty \tag{3.26}$$

Moreover, if $\pi = \lim_{k \to \infty} \pi(k)$*, then* π *satisfies Equation 3.25.*

Under the assumptions of existence of the stationary distribution for the homogeneous algorithm on the matrices $\mathbf{A}(c)$ and $\mathbf{G}(c)$, there exists an eigenvector $\mathbf{q}(c_k)$ of $\mathbf{P}(k-1, k)$, for each $k \geq 0$. Under the assumptions of convergence of the stationary distribution for the homogeneous algorithm, $\lim_{k \to \infty} c_k = 0$. Strong ergodicity with $\pi(k) = \mathbf{q}(c_k)$ can be proved if the Markov chain is weakly ergodic and if $\mathbf{q}(c_k)$, for

$k = 0, 1, 2, \ldots$, satisfies Equation 3.26. Then

$$\lim_{k \to \infty} P\left[\mathbf{X}\left(k\right) \in \mathcal{R}_{opt}\right] = 1$$

By inserting $\mathbf{q}\left(c_k\right)$ in the original formulation of the Simulated Annealing algorithm, the i^{th} component of the stationary distribution is given by

$$q_i\left(c_k\right) = \frac{\exp\left(-\frac{C(i)-C_{opt}}{c_k}\right)}{\sum_{j=1}^{|\mathcal{R}|} \exp\left(-\frac{C(j)-C_{opt}}{c_k}\right)}$$

It can be shown that under certain conditions on the acceptance matrix, the rate of convergence of the sequence $\{c_k\}$ cannot be faster than $\frac{\Gamma}{\log k}$, for some constant Γ, giving a bound on the value of c_k for each k [27]. In fact, a sufficient condition on the sequence $\{c_k\}$ where $k = 0, 1, 2, \ldots$, using Theorem 3, can be derived such that if the bound on c_k is given by

$$\exists\, k_0 \geq 2 \;\forall\, k \geq k_0 : c_k \geq \frac{|\mathcal{R}|\, \Delta C_{max}}{\log k}$$

where $\Delta C_{max} = \max\left\{C\left(i\right) | i \in \mathcal{R}\right\} - \min\left\{C\left(i\right) | i \in \mathcal{R}\right\}$. Then, Equation 3.24 is satisfied for some sequence $\{c_k\}$ where $k = 0, 1, 2, \ldots$, and hence weak ergodicity is obtained. Other sharper bounds were also proved to satisfy weak ergodicity. So the sufficient condition ensures that the algorithm converges to the set of globally minimal configurations.

If $c\left(t\right)$ is given by $c\left(t\right) = \frac{\Gamma}{\log t}$, for some constant Γ, then it can be shown that the expected time to leave a cup $\mathcal{V}$ (set of configurations that can be reached from a local minimum in a finite number of transitions) is finite if $\Gamma > d(\mathcal{V})$, where $d(\mathcal{V})$ is the

depth of the cup $\mathcal{V}$ (see Definition 9). For $\Gamma < d(\mathcal{V})$, there is a positive probability that the cup will never be left. In fact, the condition $\Gamma \geq D$, where D is the largest depth of any cup, is both necessary and sufficient for convergence to global minima [27].

3.3.2.2 Necessary and Sufficient Conditions for Convergence

Definition 8. *A configuration j is called* ***reachable*** *at height L from a configuration i, if there is a sequence of configurations $i = l_0, l_1, \ldots, l_p = j$, such that*

$$G_{l_k l_{k+1}}(c) > 0 \text{ for } k = 0, 1, \ldots, p-1$$

and

$$C(l_k) \leq L \text{ for } k = 0, 1, \ldots, p$$

Definition 9. *A* ***cup*** *is a subset $\mathcal{V}$ of the set of configurations such that for some number E*

$$\forall\, i \in \mathcal{V} : \mathcal{V} = \{j \in \mathcal{R} | j \text{ is reachable from } i \text{ at height } E\}$$

For a cup $\mathcal{V}$, let $\underline{\mathcal{V}} = \min\{C(i) | i \in \mathcal{V}\}$ and $\bar{\mathcal{V}} = \min\{C(j) | j \notin \mathcal{V} \wedge \exists\, i \in \mathcal{V} : G_{ij} > 0\}$. The **depth** $d(\mathcal{V})$ is given by $d(\mathcal{V}) = \bar{\mathcal{V}} - \underline{\mathcal{V}}$. So, a local minimum can be seen as a configuration i such that no configuration j with $C(j) < C(i)$ is reachable at height $C(i)$ from i. The depth of a local minimum i is taken to be the smallest number $d(i)$ such that there is a configuration j with $C(j) < C(i)$ reachable at height $C(i) + d(i)$ from i. If i is a global minimum, then $d(i) = +\infty$ [27].

Theorem 5. *Suppose that the one-step transition matrix is given by Equation 3.21, where $\mathbf{A}(c_k)$ is given by Equation 3.15, and that the generation matrix is independent of c, such that the Markov chain associated with $\mathbf{G}$ given by Inequality 3.12 is irreducible,*

and for any real number L and any two configurations i and j, j is reachable at height L from i. Assume furthermore, that Equation 3.22 and Equation 3.23 hold. If D is the maximum of depths $d(i)$ of all configurations i that are local but not global minima, then

$$\lim_{k \to \infty} P\left[\mathbf{X}(k) \in \mathcal{R}_{opt}\right] = 1$$

if and only if

$$\sum_{k=1}^{\infty} \exp\left(-\frac{D}{c_k}\right) = \infty \tag{3.27}$$

If c_k is of the form $c_k = \frac{\Gamma}{\log k}$, then Equation 3.27 holds if and only if $\Gamma \geq D$. The constant D is given by

$$D = \max_{j \notin \mathcal{R}_{opt}} \left(\min_{i \in \mathcal{R}_{opt}} D_{ji} \right)$$

Under certain conditions related to the matrix Π, whose entries are defined by

$$\forall\, i, j \in \mathcal{R} : \Pi_{ij} = \lim_{k \to \infty} P_{ij}(k-1, k)$$

a necessary and sufficient condition for the annealing algorithm to converge with probability 1 to a global minimum is given by

$$\exists\, k_0 \geq 1\ \forall\, k \geq k_0 : c_k \geq \frac{\Gamma}{\log k}$$

Chapter 4

Churn Modelling

Churn is the term used to represent the action that a subscriber abandons the service from his current service provider. Churn modelling is of great interest for companies who offer telecommunication services, since they can model which subscribers are at high risk to churn and when will these subscribers churn. Then, they can plan strategies to either retain their subscribers or try to acquire new subscribers. There are various techniques which can be used to model churn, however the most efficient technique and the one that can give an idea of when a subscriber might churn, is to represent subscribers with a survival model.

Survival analysis is concerned with studying the time between entry and a subsequent churn event [8]. If $T \geq 0$ is the random variable denoting the time at which an event occurs, the density function is given by $f(t)$ and the distribution function is given by $F(t)$, such that

$$F(t) = P[T \leq t] = \int_0^t f(u)\, du$$

Then the survival function $S(t)$, is given by

$$S(t) = P[T > t] = 1 - F(t) = \int_t^{\infty} f(u)\, du$$

where $S(t)$ is a monotone decreasing function from 1 to 0 such that $S(0) = 1$ and $S(\infty) = 0$. This represents the probability that an observed subscriber will survive up to time t.

Since survival analysis is mostly used for censored data, the following list gives the different types of censoring in terms of the density function, $f(t)$, the distribution function, $F(t)$, and the survival function $S(t)$.

- Uncensored: $P[T = T_i] = f(T_i)$
- Right censoring: $P[T > T_i] = 1 - F(T_i) = S(T_i)$
- Left censoring: $P[T < T_i] = F(T_i) = 1 - S(T_i)$
- Interval censoring: $P[T_{i,l} < T < T_{i,r}] = S(T_{i,l}) - S(T_{i,r})$

where T is the start of the observation period, T_i is the time at which churn occurs, and $T_{i,l} < T < T_{i,r}$ is an interval between the start and end of the observation period. Reference to Figure 2.1 helps to understand this better.

The hazard function (or the instantaneous failure rate) $\lambda(t)$, gives the rate at which a subscriber fails to survive up to time t. For the hazard function, the interval of time is taken to be smaller and smaller until the interval becomes infinitely small, Δt. The

hazard function is defined as

$$\begin{aligned}
\lambda(t) &= \lim_{\Delta t \to 0} \frac{P[t \leq T \leq t + \Delta t | T > t]}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{P[t \leq T \leq t + \Delta t \cap T > t]}{\Delta t P[T > t]} \\
&= \lim_{\Delta t \to 0} \frac{\Delta F}{\Delta t} \frac{1}{(1 - F(t))} \\
&= \frac{dF}{dt} \frac{1}{(1 - F(t))} \\
&= \frac{f(t)}{1 - F(t)} \\
&= \frac{f(t)}{S(t)}
\end{aligned}$$

There are various churn models that are used in real life applications. The following sections will give a short description of some of these models, with particular interest on the shifted-beta geometric model.

4.1 Cox Model

In survival models, the hazard function for a given individual describes the instantaneous risk of experiencing an event of interest within an infinitesimal interval of time, given that the individual has not yet experienced that event [8]. In this case, the hazard function describes the risk that a subscriber will churn in the near future, given that the subscriber is still active. The Cox model is frequently used to model the relationship of covariates (predictors) to a survival or other censored outcome [8].

Let X_{ij} denote the j^{th} covariate of the i^{th} subject, for $i = 1, \ldots, n$ and $j = 1, \ldots, p$. Then $\mathbf{X}$ is an $n \times p$ matrix whose row X_i denotes the covariate vector of subject i. The

Cox hazard function for fixed-time covariates X_i, is given by

$$\lambda_i(t) = \lambda_0(t) \exp(X_i'\beta)$$

where $\lambda_0(t)$ is the baseline hazard and β is a p-vector of regression coefficients. $\lambda_0(t)$ is defined as a nonnegative function over time for that individual with zero on all covariates.

The survival function of the Cox model is given by

$$S(t) = \exp\left[-\exp(X_i'\beta)\int_0^t \lambda_0(u)\,du\right]$$

where the integral part of the survival function is called the baseline cumulative hazard function.

The Cox model is referred to as the proportional hazard model. This proportional hazard assumption is in fact one of the restrictions in using the Cox model with time-fixed covariates. This is because the hazard ratio between two covariate values is constant over time,

$$\frac{\lambda_i(t)}{\lambda_j(t)} = \frac{\lambda_0(t)\exp(X_i'\beta)}{\lambda_0(t)\exp(X_j'\beta)} = \frac{\exp(X_i'\beta)}{\exp(X_j'\beta)}$$

since the baseline hazard function cancels out in the ratio of the two hazards. This means that the covariates must have the same effect on the hazard at any point in time.

The estimation of the parameter β is based on the partial likelihood function and this is done without estimating the baseline hazard function since the baseline hazard is typically considered to be a nuisance parameter which is not of immediate interest but which must be accounted for in the analysis of those parameters which are of interest.

The likelihood formula can be written as a product of several likelihoods, one for each event time. The likelihood at time t_i denotes the likelihood of having an event at time t_i, given survival up to time t_i [1]. The partial likelihood is given by

$$L(\beta) = \prod_{i=1}^{D} \frac{\exp(X_i'\beta)}{\left[\sum_{j \in R(t_i)} \exp(X_j'\beta)\right]^{d_i}} \tag{4.1}$$

where D is the total number of events, d_i is the number of events at time k, and $R(k)$ is the set of individuals at risk, called the risk set, at time k.

When building a Cox model, it is important to identify the variables that are most associated with the churn event. For a given subscriber i, a hazard function indicates the probability $\lambda_i(t)$ of cancellation at a given time t in the future. A hazard curve can be converted to a survival curve or to a survival function which plots the probability $S_i(t)$ of non-cancellation at any time t, given that customer was active at time $t-1$ such that

$$S_i(t) = S_i(t-1) \times [1 - \lambda_i(t)]$$

with $S_i(0) = 1$ [8].

4.2 Extended Cox Model

A problem which arises in the Cox model is that it is not suitable as a predictive model for prepaid customers who churn. This is because the covariates are fixed over time. So, a variation of the Cox model is the Extended Cox model which includes the ability to accommodate censored data, time-varying covariates and multiple events. In this case, the proportionality assumption does not have to hold since the covariates are dependent

on time t. The Extended Cox model is given by

$$\lambda_i(t) = \lambda_0(t) \exp\left[\sum_{i=1}^{p_1} X_i'\beta + \sum_{j=1}^{p_2} X_j'\beta(t)\right]$$

where the covariates are split in p_1 time-independent covariates and p_2 time-dependent covariates.

In order to include time-varying covariates in the Cox model, a counting process formulation is required. A counting process is a stochastic process starting at 0 and whose sample paths are right continuous step functions with height 1. The counting process formulation makes it possible to include multiple event times and multiple at-risk intervals [1].

4.3 Aalen Model

Another alternative to the Cox model is the additive risk model of Aalen. Let $T = [0, \tau]$, for $0 < \tau < \infty$, be a fixed time interval and consider an n-variate counting process $N(t) = (N_i(t), i = 1, \ldots, n)$, together with a matrix of covariates $Y_{ij}(t)$, for $j = 1, \ldots, p$ given that $p \leq n$, observed for each component $N_i(t)$. The covariate $Y_{ij}(t)$ is set equal to 0 if the individual i is not at risk [10]. By assuming an intensity process $\gamma_i(t)$ of $N_i(t)$, the relationship between the covariates $Y_{ij}(t)$ and the counting process $N(t)$, is given by

$$\gamma_i(t) = \sum_{j=1}^{p} Y_{ij}(t)\,\alpha_j(t) \quad t \in T$$

where $\alpha_j(t)$ are deterministic baseline intensities that are specified under some regularity conditions.

An estimator for the integrated baseline intensity $\mathbf{B}(t) = \int_0^t \boldsymbol{\alpha}(s)\,ds$ is given by a

generalized Nelson-Aalen estimator

$$\hat{\mathbf{B}}(t) = \int_0^t \mathbf{Y}^{-}(s)\, d\mathbf{N}(s)$$

where $\mathbf{Y}^{-}(t)$ is a generalized inverse of $\mathbf{Y}(t)$ such that $\mathbf{Y}^{-}(t)\,\mathbf{Y}(t) = \mathbf{I}$.
Usually the estimation of $\boldsymbol{\alpha}(t)$ is more useful than estimating $\mathbf{B}(t)$. So let $b > 0$ and K a kernel, then an estimator for $\boldsymbol{\alpha}$ is given by

$$\hat{\boldsymbol{\alpha}}(t) = \frac{1}{b}\int_T K\left(\frac{t-s}{b}\right) d\hat{\mathbf{B}}(s) \;\; t \in [b, \tau - b]$$

This model is usually used in life insurance to estimate the cumulative number of expected events.

4.4 Stratified Cox Model

One of the restrictions of using a Cox model with fixed time is its proportional hazards (PH) assumption. Let $t_1, \ldots, t_d$ be d unique ordered event (churn) times, and let $X_i(s)$ be the $p \times 1$ covariate vector for the i^{th} individual at time s. Note that for time-fixed covariates, $X_i(s) = X_i$. The weighted mean of the $X_i(s)$ over those still at risk to churn at time s is given by

$$\overline{X}(s) = \frac{\sum Y_i(s) \exp\left(X_i(s)\,\hat{\beta}\right) X_i(s)}{\sum Y_i(s) \exp\left(X_i(s)\,\hat{\beta}\right)}$$

where $Y_i(s)$ is the predictable variation process indicating whether observation i is at risk at time s, so that $Y_i(s) = 1$ if observation i is still at risk at time s and is zero otherwise. The estimate $\hat{\beta}$ comes from fitting a Cox PH model. In particular, a

Schoenfeld residual is a $p \times 1$ vector that is defined at the k^{th} churn event time as

$$s_k = \int_{t_{k-1}}^{t_k} \sum_i \left[X_i(s) - \bar{X}(s) \right] dN_i(s)$$

where $N_i(s)$ is a counting process that counts the number of events for observation i at time s. Thus s_k sums the quantities $X_i(t_k) - \overline{X}(t_k)$ over observations that have experienced the event by time t_k.

An alternative to a PH model is to stratify the model across levels of one or more covariates, leading to a Stratified Cox model. A Stratified Cox model is useful when a factor does not affect the hazard multiplicatively. The strata divide the subjects into disjoint groups, each of which has a distinct arbitrary baseline hazard function, but have common values for the coefficients β. The hazard function for an individual i who belongs to stratum k is then given by

$$\lambda(t) = \lambda_k(t) \exp(X_i' \beta)$$

The stratified Cox model allows a deviation from proportional hazards and provides an alternative to the assumption of proportional hazards. The hazard functions for two different strata do not have to be proportional to one another, however, within a stratum, proportional hazards are assumed to hold. The partial likelihood for Stratified Cox models with K strata is a product of K terms, each of the form of Equation 4.1, but where i ranges over only the subjects in stratum k, for $k = 1, \ldots, K$. Stratification entails fitting separate baseline hazard functions across strata. A baseline hazard function represents the hazard rate over time for an individual with all modelled covariates set to zero. With a Stratified Cox model, a proportional hazards structure does not necessarily hold for the combined data, but it is assumed to hold within each stratum.

However, the coefficients on the included covariates are common across strata so that the relative effect of each predictor is the same across strata, unless there is a significant strata-by-covariate interaction, which means that the effect of the covariate differs within strata [8].

A Bayesian version for the Stratified Cox model is

$$\lambda_i(t) = \lambda_{0i}(t) \exp(\beta' X)$$

where $\lambda_{0i}(t)$ are the stratum-specific baseline hazards.

4.5 Shifted-Beta Geometric Model

An alternative to common curve fitting regression models is introduced by Fader and Hardie in [5] which is a probability model for the churn process. This basic model known as the *shifted-beta geometric* model which can be implemented in a simple Microsoft Excel spreadsheet and this model provides very accurate forecasts of customer retention.

For this model it is important to explain in slightly more detail the definition of retention rate and churn rate. The *retention rate* for time t, given by r_t, is defined as the proportion of customers active at the end of time $t-1$ who are still active at the end of period t. On the other hand, the *churn rate* for a given period is defined as the proportion of customers active at the end of time $t-1$ who are not active by time t.

In the beginning of Chapter 4, the survival function was given in terms of the distribution function. However, the probability that a customer is still active at time t

can also be given in terms of the retention rate by

$$S(t) = r_1 \times \cdots \times r_t = \prod_{i=1}^{t} r_i$$

such that

$$r_t = \frac{S(t)}{S(t-1)} \tag{4.2}$$

In statistics, the geometric distribution is either of two discrete probability distributions:

- The probability distribution of the number X of Bernoulli trials needed to get one success, supported on the set $1, 2, 3, \ldots$
- The probability distribution of the number $Y = X - 1$ of failures before the first success, supported on the set $0, 1, 2, 3, \ldots$

Often, the name *shifted* geometric distribution is adopted for the former one.

The shifted-beta geometric model for the duration of customer lifetimes is based on two assumptions. Suppose that an individual remains a customer of the company with constant retention probability $1 - \theta$. This is equivalent to assuming that the duration of the customer's relationship with the company, denoted by the random variable T, is characterized by the shifted-geometric distribution with probability mass function $P(T = t|\theta)$ and survival function $S(t|\theta)$, given by

$$\begin{aligned} P(T = t|\theta) &= \theta(1-\theta)^{t-1}, \; t = 1, 2, 3, \ldots \\ S(t|\theta) &= (1-\theta)^{t}, \; t = 1, 2, 3, \ldots \end{aligned}$$

The second assumption is about the heterogeneity in θ which follows a beta distribution

with probability density function

$$f(\theta|\alpha,\beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)}$$

where $B(\alpha,\beta)$ is the beta function defined by

$$B(\alpha,\beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}\,d\theta,\ \alpha,\beta > 0 \tag{4.3}$$

The beta function can be expressed in terms of gamma functions, such that

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

Figure 4.1 shows that if both parameters, α and β, of the beta distribution are less than 1, then the churn probability θ is U-shaped, as shown by the red curve. If both parameters are large, such that $\alpha,\beta > 1$, then the shape of the beta distribution is unimodal, that is, for some value m the curve of the beta distribution is monotonically increasing for $x \leq m$ and monotonically decreasing for $x \geq m$, as shown by the purple and black curve. If one parameter is large while the other is small, the beta distribution shape is either J-shaped or reverse-J-shaped, as shown by the green and blue curves. These various shapes can model the nature of heterogeneity in churn probabilities across the customer base.

Since the customer's value of θ is unobserved, the equations for the first assumption cannot be used. So the expectation of $P(T=t|\theta)$ and $S(t|\theta)$ over the beta distribution are used instead to arrive at the corresponding expressions for a randomly chosen subscriber. If θ is known, the probability of churning at time t would simply be the geometric probability $\theta(1-\theta)^{t-1}$. But since θ is unobserved, $P(T=t)$ for a randomly

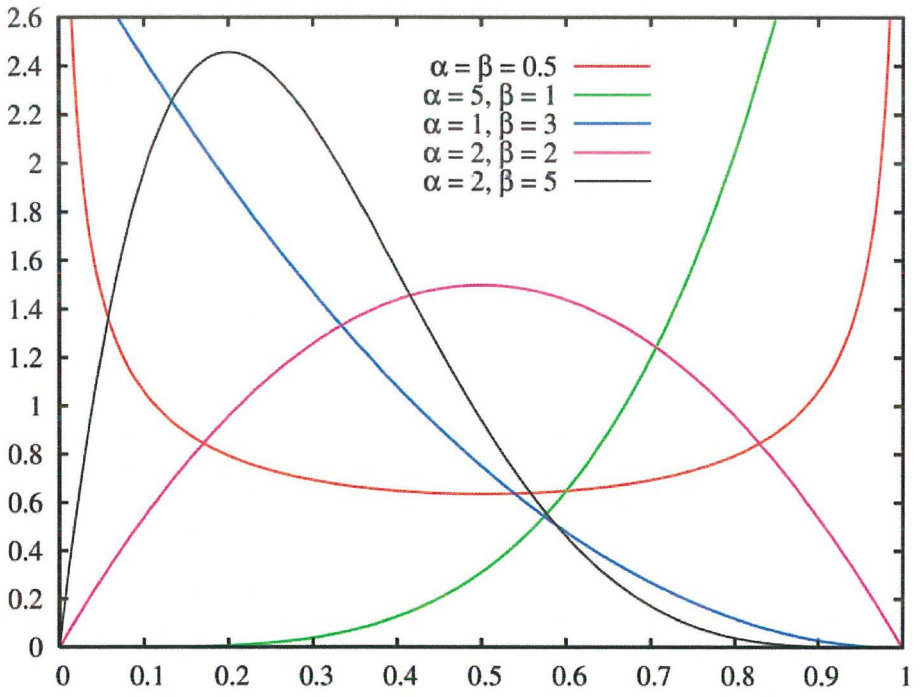

Figure 4.1: General shapes of the beta distribution as a function of α and β

chosen subscriber is the expected value of the shifted-geometric probability of churning at time t, conditionally on $\Theta = \theta$, where the expectation is with respect to the beta distirbution for Θ, $E[P(T = t|\Theta = \theta)]$. So each $P(T = t|\Theta = \theta)$ is weighted by the probability of the value of θ occuring, $f(\theta)$.

Since Θ is a continuous random variable, this is computed as

$$\begin{aligned} P(T = t|\alpha, \beta) &= \int_0^1 P(T = t|\Theta = \theta) f(\theta|\alpha, \beta) d\theta \\ &= \int_0^1 \theta (1-\theta)^{t-1} \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)} d\theta \\ &= \frac{1}{B(\alpha, \beta)} \int_0^1 \theta^{\alpha} (1-\theta)^{\beta+t-2} d\theta \end{aligned}$$

where $P(T=t|\Theta=\theta)$ is the probability distribution function of the geometric distribution and $f(\theta|\alpha,\beta)$ is the probability density function of the beta distribution. $\int_0^1 \theta^{\alpha}(1-\theta)^{\beta+t-2}\,d\theta$ is the integral expression for the beta function with parameters $\alpha+1$ and $\beta+t-1$. Therefore,

$$P(T=t|\alpha,\beta)=\frac{B(\alpha+1,\beta+t-1)}{B(\alpha,\beta)},\ t=1,2,\ldots$$

Similarly,

$$S(t|\alpha,\beta)=\frac{B(\alpha,\beta+t)}{B(\alpha,\beta)},\ t=1,2,\ldots \tag{4.4}$$

This model is called the shifted-beta geometric (sBG) model with parameters α, β having an sBG distribution.

This model can be used without having to deal with the beta function. The sBG probabilites are computed using a forward-recursion formula from $P(T=1)$, where,

$$\begin{aligned} P(T=1|\alpha,\beta) &= \frac{B(\alpha+1,\beta)}{B(\alpha,\beta)} \\ &= \frac{\Gamma(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)}\Big/\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \\ &= \frac{\Gamma(\alpha+1)}{\Gamma(\alpha)}\Big/\frac{\Gamma(\alpha+\beta+1)}{\Gamma(\alpha+\beta)} \end{aligned}$$

By using the property of recursion for the gamma function,

$$\frac{\Gamma(\alpha+1)}{\Gamma(\alpha)}=\alpha \quad \text{and} \quad \frac{\Gamma(\alpha+\beta+1)}{\Gamma(\alpha+\beta)}=\alpha+\beta$$

Then,

$$P(T=1|\alpha,\beta)=\frac{\alpha}{\alpha+\beta} \tag{4.5}$$

For the case when $t = 2, 3, \ldots,$ consider the identity

$$P(T = t) = \frac{P(T = t)}{P(T = t-1)} P(T = t-1)$$

Given the expression $P(T = t)/P(T = t-1)$, the value of $P(T = 2)$ can be computed using the value $P(T = 1) = \alpha/(\alpha + \beta)$. Then, given the value of $P(T = 2)$, the value of $P(T = 3)$ can be computed, and so on. So,

$$\begin{aligned} \frac{P(T = t)}{P(T = t-1)} &= \frac{B(\alpha + 1, \beta + t - 1)}{B(\alpha, \beta)} \Big/ \frac{B(\alpha + 1, \beta + t - 2)}{B(\alpha, \beta)} \\ &= \frac{B(\alpha + 1, \beta + t - 1)}{B(\alpha + 1, \beta + t - 2)} \end{aligned}$$

By expressing the beta functions in term of gamma functions,

$$\frac{P(T = t)}{P(T = t-1)} = \frac{\Gamma(\beta + t - 1)}{\Gamma(\beta + t - 2)} \Big/ \frac{\Gamma(\alpha + \beta + t)}{\Gamma(\alpha + \beta + t - 1)}$$

and by the recursive property,

$$\frac{P(T = t)}{P(T = t-1)} = \frac{\beta + t - 2}{\alpha + \beta + t - 1} \tag{4.6}$$

By combining Equation 4.5 and Equation 4.6,

$$P(T = t) = \begin{cases} \frac{\alpha}{\alpha+\beta} & t = 1 \\ \frac{\beta+t-2}{\alpha+\beta+t-1} P(T = t-1) & t = 2, 3, \ldots \end{cases} \tag{4.7}$$

By substituting Equation 4.4 into Equation 4.2,

$$\begin{aligned} r_t &= \frac{B(\alpha, \beta + t)}{B(\alpha, \beta)} \Big/ \frac{B(\alpha, \beta + t - 1)}{B(\alpha, \beta)} \\ &= \frac{B(\alpha, \beta + t)}{B(\alpha, \beta + t - 1)} \end{aligned}$$

Expressing the beta functions in term of gamma functions,

$$r_t = \frac{\Gamma(\beta + t)}{\Gamma(\beta + t - 1)} \Big/ \frac{\Gamma(\alpha + \beta + t)}{\Gamma(\alpha + \beta + t - 1)}$$

By the recursive property of the gamma function, the retention rate associated with the sBG model is given by,

$$r_t = \frac{\beta + t - 1}{\alpha + \beta + t - 1}$$

So it is possible to compute $S(t)$ without having to deal with the beta functions.

The retention rate under the sBG model is an increasing function of time due to heterogeneity. That is, the high churn subscribers drop out early in the observation period, with the remaining subscribers having lower churn probabilities.

Chapter 5

Model Implementation

5.1 Introduction

The number of mobile service subscriptions has been increasing over the past couple of years. By looking at the total number of mobile telephone subscriptions for the past year (2009), it is evident that this is true. Table 5.1 shows the total number of mobile telephone subscriptions and this number divided into the number of post-paid subscriptions and the number of pre-paid subscriptions for the year 2009 obtained from an article by the National Statistics Office [18].

Months	Total	Post-paid	Pre-paid
Jan - Mar	388,284	68,754	319,530
Apr - June	405,465	74,696	330,769
July - Sept	418,341	59,867	358,474
Sept - Dec	422,083	78,384	343,694

Table 5.1: Number of mobile telephone subscriptions in 2009

5.2 Data Set

The data set used for this dissertation was provided by Vodafone (Malta) Ltd. The dataset consisted of the average subscriber base for two tariffs, It's Good to Talk (IGTT) and Friends tariff, for a 20-week period. The number of subscribers for each tariff is divided into seven segments where for each segment, the initial number of subscribers is 1,000 and the probability that a subscriber is in one of the seven segments is given, as shown in Table 5.2. This dataset shows the number of subscribers who churned, however it does not indicate whether the subscribers churn at the customer level or else at the contract level. Customer level churning is when the subscriber switches to a different service provider, whereas contract level churning is when the subscriber changes the tariff scheme but still remains with the same service provider [9].

Segment	Probability
Segment 0	0.08
Segment 1	0.17
Segment 2	0.24
Segment 3	0.16
Segment 4	0.13
Segment 5	0.19
Segment 6	0.03

Table 5.2: Probabilities for each segment

The plots of all segments for both tariffs are shown in Figure 5.1. However by aggregating the data (multiplying the subscriber base for each segment by the segment probabilities and adding the answers together), each tariff corresponds to one of the curves in Figure 5.2. This is done because the segments in both tariffs are very similar to each other.

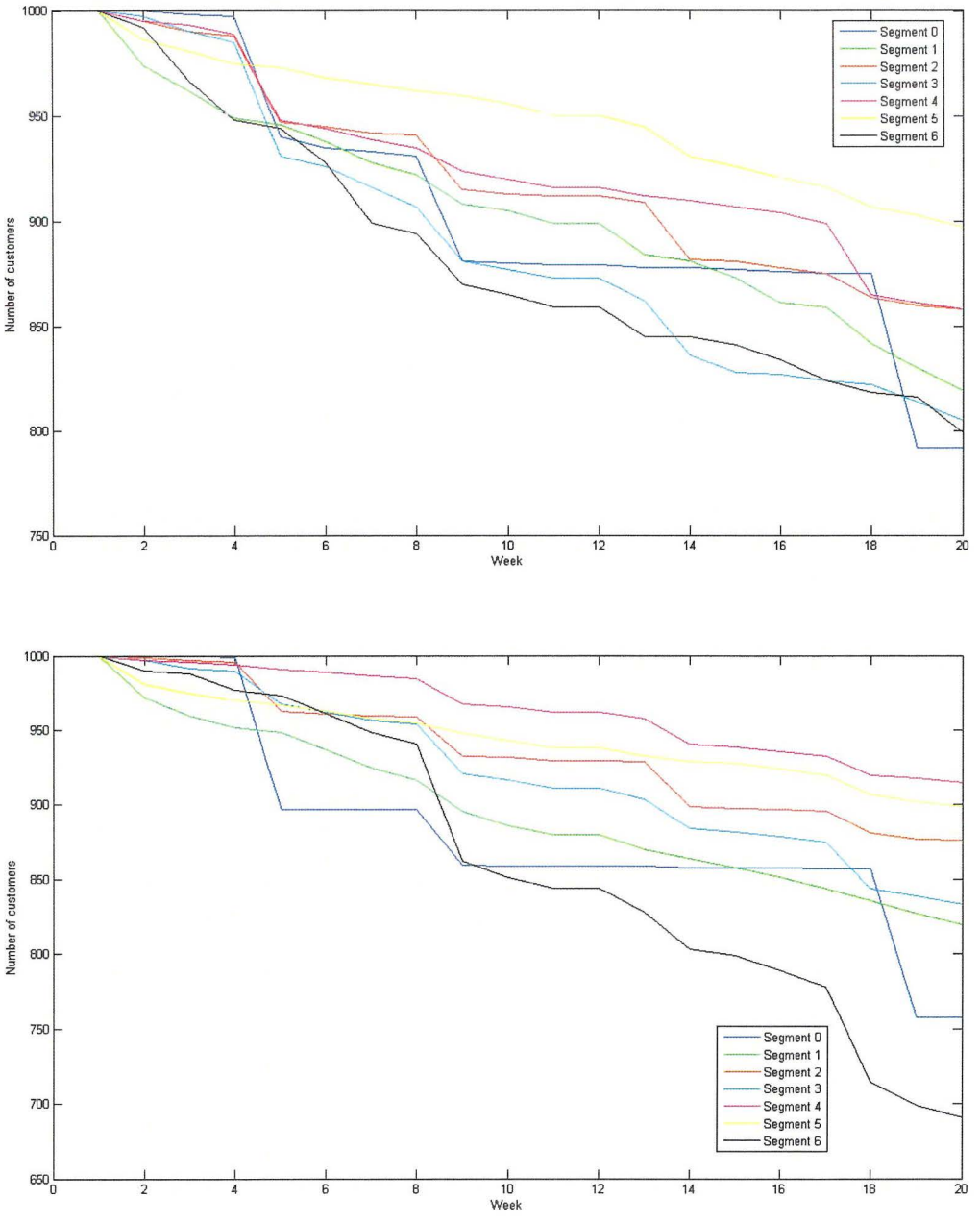

Figure 5.1: Top: IGTT segments, Bottom: Friends segments

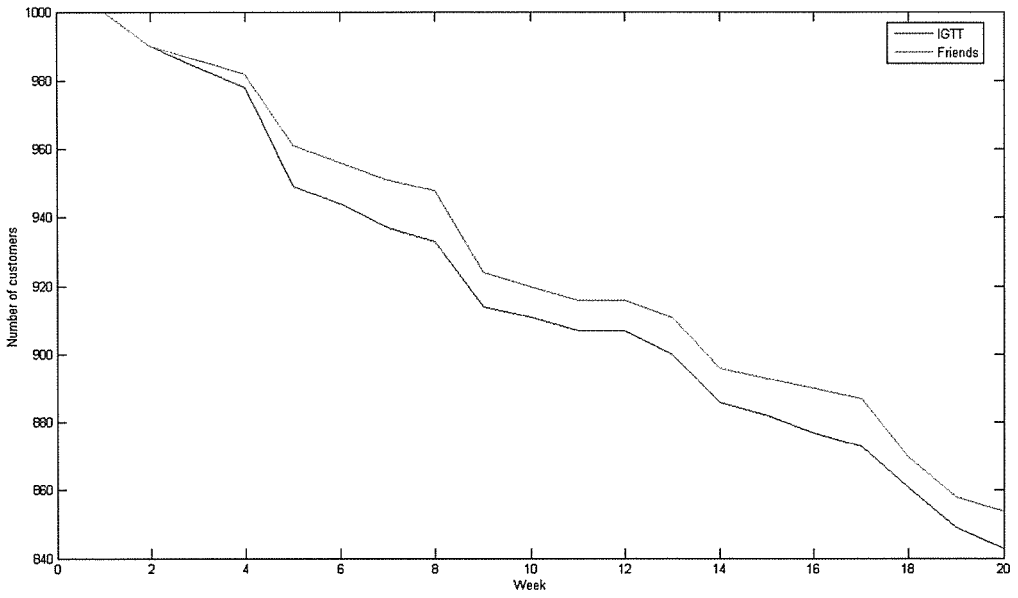

Figure 5.2: Aggregated tariffs

5.3 Implementation

Several strategies were implemented for this data including, dividing the subscribers into two segments with different fixed segment probabilities and with optimizable probabilities, and dividing the subscribers into three segments with fixed segment probabilities and with optimizable probabilities. For these churn models the dataset is assumed to follow a geometric distribution. In these cases, the first eleven weeks of the dataset were used for optimizing the parameters of the model and then the predicted values for the remaining eight weeks were computed and compared to the actual data. Finally, the data for both tariffs was modelled using the sBG distribution. In this case, the first fourteen weeks were used for optimizing the parameters of the model and then the pre-

dicted values for the remaining five weeks were computed and compared to the actual data. The purpose for having fourteen weeks to compute the optimization of the sBG model is because eleven weeks are not sufficient to obtain an accurate prediction, so the three extra weeks that were used helped to provide a much more accurate prediction. For the other models, eleven weeks were sufficient, since if fourteen weeks were used, the improvement in the predictions was very minimal. These strategies were implemented into Microsoft Excel and Matlab to compute the maximum likelihood estimations using various model parameters for the dataset.

Given different initial values for optimization, Microsoft Excel usually generates different optima since the solver implemented in Microsoft Excel does not use a global optimization algorithm. The Microsoft Excel Solver tool uses the Generalized Reduced Gradient (GRG2) nonlinear optimization code. Linear and integer problems use the simplex method with bounds on the variables and the branch-and-bound method. Hence the initial values in Microsoft Excel had to be chosen by trial-and-error until the optimal values were obtained. On the other hand, by using the simulated annealing algorithm in Matlab given by [26], the global optima could be found straightaway in most cases. Before providing the results for the mentioned churn models, a brief review of notion for the maximum likelihood estimation is given. Note that in Microsoft Excel, the log-likelihood function is being maximized, while the negative log-likelihood function in Matlab is being minimized. So, the maximized log-likelihood function in Microsoft Excel will give a negative result, while the maximized log-likelihood function in Matlab will give a positive result.

5.3.1 Maximum Likelihood Estimation

Suppose the dataset consists of a group of N subscribers for p weeks, such that there are no subscribers who churn during the initial observing week (week 0). Then n_1 subscribers churn in the first week, ..., n_p subscribers churn in the p^{th} week. Then $N - \sum_{t=1}^{p} n_t$ subscribers are still active at the end of the p^{th} week. The probability that a randomly chosen subscriber has a lifetime of one week is given by $P(T = 1|\boldsymbol{\theta})$, ..., the probability that a randomly chosen subscriber has a lifetime of p weeks is given by $P(T = p|\boldsymbol{\theta})$. By assuming that a subscriber churns independently of the behaviour of another subscriber, the probability that one randomly chosen subscriber has a lifetime of one week while another subscriber has a lifetime of two weeks is the product of the respective geometric probabilities, such that $P(T = 1|\boldsymbol{\theta})\, P(T = 2|\boldsymbol{\theta})$. Thus, it follows that given specific values of the model parameters $\boldsymbol{\theta}$, the joint probability of n_1 subscribers churning in the first week, ..., n_p in the p^{th} week, and $N - \sum_{t=1}^{p} n_t$ subscribers still being active at the end of the p^{th} week is given by

$$P(\text{data}|\boldsymbol{\theta}) = P(T = 1|\boldsymbol{\theta})^{n_1}\, P(T = 2|\boldsymbol{\theta})^{n_2} \times \cdots \times P(T = p|\boldsymbol{\theta})^{n_p}\, S(p|\boldsymbol{\theta})^{N - \sum_{t=1}^{p} n_t}$$

However, the values of $\boldsymbol{\theta}$ are unknown, although it is assumed that $\boldsymbol{\theta}$ follows a geometric distribution. Maximum likelihood estimation is used to find which values of the model parameters maximize the probability of the given dataset. The likelihood function is given by

$$L(\boldsymbol{\theta}|\text{data}) = P(T = 1|\boldsymbol{\theta})^{n_1}\, P(T = 2|\boldsymbol{\theta})^{n_2} \times \cdots \times P(T = p|\boldsymbol{\theta})^{n_p}\, S(p|\boldsymbol{\theta})^{N - \sum_{t=1}^{p} n_t}$$

and by using a numerical optimization method, the values of $\boldsymbol{\theta}$ which maximize the function are evaluated. The values which maximize the likelihood function are called

the maximum likelihood estimates of the model parameters. Usually the value of the likelihood function is very small, and thus the natural logarithm of the likelihood function is used instead. This is called the log-likelihood function and is given by

$$LL\left(\boldsymbol{\theta}|\text{data}\right) = \sum_{t=1}^{p} \ln\left[P\left(T=t|\boldsymbol{\theta}\right)\right] + \left(N - \sum_{t=1}^{p} n_t\right) \ln\left[S\left(p|\boldsymbol{\theta}\right)\right]$$

The same applies for the sBG distribution, with model parameters α and β, such that the joint probability is given by

$$P\left(\text{data}|\alpha,\beta\right) = P\left(T=1|\alpha,\beta\right)^{n_1} \times \cdots \times P\left(T=p|\alpha,\beta\right)^{n_p} S\left(p|\alpha,\beta\right)^{N-\sum_{t=1}^{p} n_t}$$

The likelihood function is given by

$$L\left(\alpha,\beta|\text{data}\right) = P\left(T=1|\alpha,\beta\right)^{n_1} \times \cdots \times P\left(T=p|\alpha,\beta\right)^{n_p} S\left(p|\alpha,\beta\right)^{N-\sum_{t=1}^{p} n_t}$$

and the log-likelihood function is given by

$$LL\left(\alpha,\beta|\text{data}\right) = \sum_{t=1}^{p} \ln\left[P\left(T=t|\alpha,\beta\right)\right] + \left(N - \sum_{t=1}^{p} n_t\right) \ln\left[S\left(p|\alpha,\beta\right)\right]$$

5.3.2 It's Good to Talk Tariff

5.3.2.1 Two segments with fixed probabilities 0.4 & 0.6

The first churn model implemented for the IGTT data was that of dividing the data into two segments with fixed probabilities, where the probability for the first segment is $\text{prob}_1 = 0.4$ and the probability for the second segment is $\text{prob}_2 = 0.6$. This means that a subscriber has 0.4 probability of being in the first segment and 0.6 probability of being in the second segment. An initial starting point for $\boldsymbol{\theta}$ must be provided to calculate $P(T = t|\boldsymbol{\theta})$, such that the initial point is within the bounds $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. The values for $P(T = t|\boldsymbol{\theta})$ were computed using the initial values and using the forward-recursion method given by Equation 4.7.

The dataset consists of $n = 1,000$ customers and the number of customers churning at the t^{th} week is $n_t = N_{t-1} - N_t$. The values of $S(t|\boldsymbol{\theta})$ were computed such that $S(1) = 1 - P(T = 1)$, for $t = 1$, and $S(t) = S(t-1) - P(T = t)$, for $t > 1$. The first eleven weeks ($p = 11$) were used to compute the log-likelihood function, so at the t^{th} week, the log-likelihood function is given by $LL(\boldsymbol{\theta}|\text{t}) = \sum_{t=1}^{11} n_t \ln[P(T = t|\boldsymbol{\theta})]$. The maximum likelihood estimates of the model where found by maximizing the log-likelihood function. The initial values for $\boldsymbol{\theta}$ were selected by trial-and-error in Microsoft Excel until the optimal values were obtained, where $\theta_1 = 0.0241$, $\theta_2 = 0.0001$, and $\text{LL} = -531.1940$. Table 5.3 shows the Microsoft Excel computation. This means that both segments have a very small churn probability.

Similarly to Microsoft Excel, an initial value for $\boldsymbol{\theta}$ had to be given to the simulated annealing algorithm in Matlab. The outputs from Matlab gave the same global optimum function value with a slight discrepancy. To assess the performance of the simulated annealing algorithm, 100 different initial points covering evenly the unit square were

entered as the initial values. The matrix **A** represents these 100 different initial values where $A_{ij} = 1$ if the initial point ij results in the global optimum and $A_{ij} = 0$ if the initial point ij does not give the global optimum. The matrix **Z** represents the optimal value for all of the 100 initial points, 'best' represents the global optimum value of the maximized log-likelihood function and 'bestmin' represents the parameter values $\boldsymbol{\theta}$, corresponding to the global optimum. For this model, best = 530.5658 and bestmin = [0.0242 0.0000]. This means that the global optimum is 530.5658 and the corresponding model parameter values are $\theta_1 = 0.0242$ and $\theta_2 = 0$ meaning that if the data is divided into two segments, churning is expected from the first segment since the churn probability of the second segment is 0. In this case **A** has seventy-five 1's, meaning that 75 initial points out of 100 gave the global optimal solution. The matrix **Z** gives the log-likelihood value of all 100 initial points. In fact, by comparing **A** to **Z** it is clear which values are not in the range of the global optimal solution and these sum up to twenty-five.

$$\mathbf{A} = \begin{matrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{matrix}, \quad \hat{\mathbf{A}} = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

To test whether the simulated annealing algorithm truely finds more global optimal solutions, the same test is carried out using the algorithm `fmincon` from Matlab. In this case, the matrix $\hat{\mathbf{A}}$ shows that 38 out of 100 initial points corresponding in obtaining a global optimal solution. The matrix $\hat{\mathbf{Z}}$ shows the log-likelihood values obtained from the optimal values represented in $\hat{\mathbf{A}}$. Since $\hat{\mathbf{A}}$ has less 1 entries than $\mathbf{A}$, it is clear that the simulated annealing algorithm improves the number of initial values which obtain the global minimum value. The matrix $\hat{\mathbf{Z}}$ indicates that the standard optimization algorithm `fmincon` reaches other local optima, in particular, $LL = 857.5971$ and $LL = 616.8264$ amongst others.

The accuracy of the model was tested by predicting the values for the remaining eight weeks, using the optimal parameter estimates obtained from Microsoft Excel and Matlab. The probabilities of the remaining weeks were calculated so that the predicted number of subscribers were computed such that $M_X(t) = M_X(t-1) - M_X(t-1)\,P[T = t-1]$, where $M_X(t)$ are the predictions from Microsoft Excel, and $M_M(t) = M_M(t-1) - M_M(t-1)\,P[T = t-1]$, where $M_M(t)$ are the predictions from Matlab. These values are given in Table 5.4 and Figure 5.4 is the plot of the exact number of customers per week and their predictions. This figure shows that for this model, the predictions are quite reasonable. Figure 5.3 shows that the global optimal value is at the bottom right corner.

Two segments of subscribers with probabilities 0.4 & 0.6					
	Segments:	1	2		
	Thetas:	**0.0241**	**0.0001**		
	Probabilities:	0.4	0.6		
	LL:	**-530.5889**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0097	10	0.9903	-46.3726
2	984	0.0095	6	0.9809	-27.9688
3	978	0.0092	6	0.9716	-28.1140
4	949	0.0090	29	0.9626	-136.5858
5	944	0.0088	5	0.9538	-23.6702
6	937	0.0086	7	0.9453	-33.3076
7	933	0.0084	4	0.9369	-19.1297
8	914	0.0082	19	0.9287	-91.3253
9	911	0.0080	3	0.9207	-14.4923
10	907	0.0078	4	0.9129	-19.4198
11	907	0.0076	0	0.9053	0.0000
12		0.0074			-90.2027
13		0.0072			
14		0.0071			
15		0.0069			
16		0.0067			
17		0.0066			
18		0.0064			
19		0.0063			

Table 5.3: Excel: Two segments of subscribers with probabilities 0.4 & 0.6

$$
\mathbf{Z} = \begin{matrix}
530.5720 & 530.6010 & 531.1850 & 530.5814 & 530.5837 & 530.6443 & 530.6187 & 531.2059 & 530.5889 & 531.2022 \\
530.5877 & 530.5908 & 530.5851 & 530.5711 & 530.5777 & 531.6780 & 531.6480 & 531.2028 & 530.5842 & 530.5680 \\
530.5883 & 530.5907 & 530.6260 & 530.6613 & 530.6026 & 530.6025 & 530.5703 & 530.6266 & 530.6078 & 530.5710 \\
530.5787 & 530.5774 & 530.5897 & 530.6209 & 530.5836 & 530.5769 & 531.2008 & 531.1891 & 530.5894 & 531.1917 \\
530.5746 & 531.2141 & 530.6239 & 531.2272 & 530.5757 & 531.6956 & 530.5743 & 530.5723 & 531.5083 & 530.5928 \\
531.1846 & 530.5659 & 530.5975 & 530.5759 & 530.5829 & 530.5888 & 530.6403 & 531.3860 & 530.5795 & 530.5706 \\
530.5722 & 530.7535 & 530.5830 & 530.5765 & 531.2906 & 531.1941 & 530.5913 & 530.5835 & 530.5785 & 530.5938 \\
530.5695 & 530.5760 & 530.5754 & 530.5863 & 530.6404 & 531.2481 & 530.6158 & 530.5658 & 530.5894 & 530.5864 \\
530.5713 & 530.5660 & 530.6557 & 530.5682 & 530.5865 & 530.5737 & 531.1960 & 530.6177 & 531.1873 & 530.5909 \\
530.6569 & 531.1864 & 531.1950 & 531.2667 & 531.1989 & 530.6328 & 530.5817 & 530.6221 & 530.5778 & 531.1961
\end{matrix}
$$

$$\hat{\mathbf{Z}} = \begin{matrix}
530.5889 & 531.1940 & 857.5971 & 531.1940 & 857.5971 & 531.1940 & 531.1940 & 550.8603 & 857.5971 & 531.1940 \\
530.5889 & 530.5889 & 857.5971 & 531.1940 & 531.1940 & 530.5889 & 531.1940 & 531.1940 & 559.9512 & 530.5889 \\
530.5889 & 530.5889 & 531.1940 & 531.1940 & 531.1940 & 616.8264 & 531.1940 & 857.5971 & 530.5889 & 530.5889 \\
530.5889 & 530.5889 & 530.5889 & 847.4652 & 530.5889 & 626.3889 & 531.1940 & 530.5889 & 857.5971 & 531.1940 \\
530.5889 & 530.5889 & 530.5889 & 531.1940 & 857.5971 & 559.2445 & 533.9439 & 530.5889 & 857.5971 & 601.1401 \\
530.5889 & 530.5889 & 530.5889 & 531.1940 & 530.5889 & 531.1940 & 530.5889 & 557.5935 & 531.1940 & 857.5971 \\
530.5889 & 530.5889 & 531.8035 & 530.5889 & 531.1940 & 530.5889 & 857.5971 & 531.1940 & 531.1940 & 857.5971 \\
530.5889 & 531.1940 & 530.5889 & 530.5889 & 531.1940 & 531.1940 & 857.5971 & 530.5894 & 531.1940 & 531.1940 \\
530.5889 & 531.1940 & 530.5889 & 530.5889 & 531.1940 & 531.1940 & 530.5889 & 530.5889 & 857.5971 & 660.2535 \\
530.5889 & 530.5889 & 531.1940 & 531.1940 & 531.1940 & 531.1940 & 857.5971 & 531.1940 & 531.1940 & 531.1940
\end{matrix}$$

$\bar{E}(t)$	$\bar{M}_X(t)$	$\bar{M}_M(t)$
1000	1000	1000
990	990.3157	990.3200
984	980.9545	980.9657
978	971.9035	971.9240
949	963.1504	963.1824
944	954.6834	954.7291
937	946.4914	946.5527
933	938.5638	938.6426
914	930.8904	930.9883
911	923.4615	923.5803
907	916.2678	916.4090
907	909.3007	909.4656
900	902.5515	902.7415
886	896.0124	896.2287
882	889.6755	889.9193
877	883.5335	883.8060
873	877.5795	877.8816
861	871.8066	872.1393
849	866.2084	866.5727
843	860.7788	861.1754

Table 5.4: Exact and Model number of subscribers

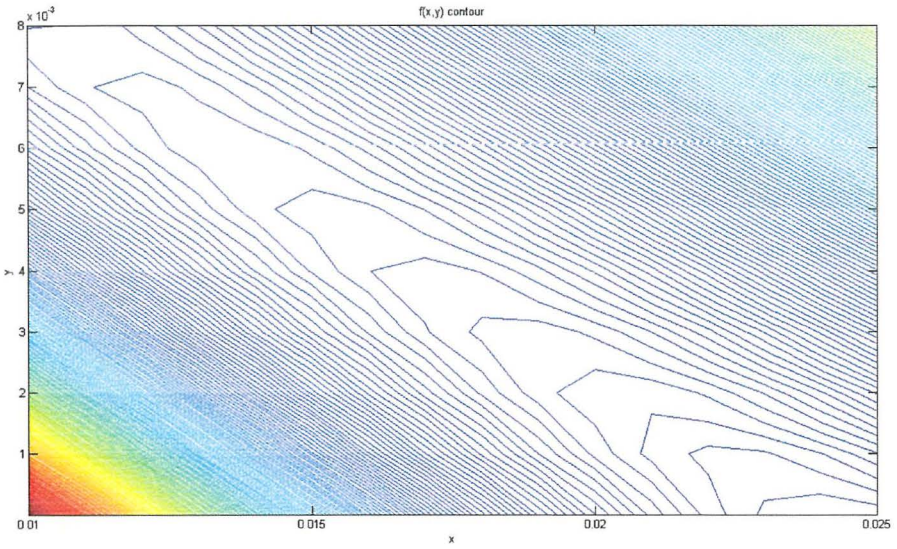

Figure 5.3: Contour figure showing the optimal value

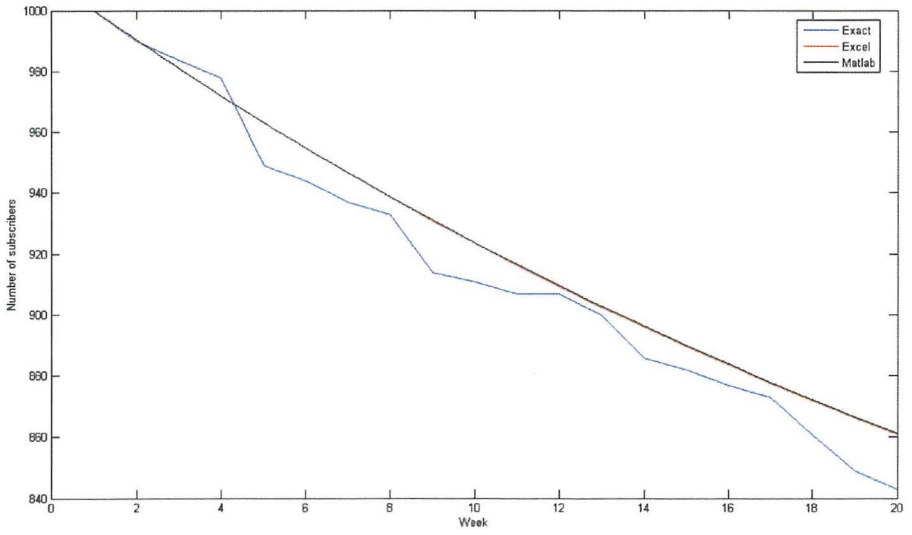

Figure 5.4: Model prediction for two segments with probabilities 0.4 & 0.6

5.3.2.2 Two segments with fixed probabilities 0.1 & 0.9

The next churn model was to divide the data into two segments with fixed probabilities, where the probability for the first segment is $\text{prob}_1 = 0.1$ and the probability for the second segment is $\text{prob}_2 = 0.9$. An initial point was given in the range $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. The optimal values obtained from Microsoft Excel are $\theta_1 = 0.1124$, $\theta_2 = 0.0019$, meaning that it is more probable for a subscriber in the first segment to churn, and the maximum log-likelihood function is $\text{LL} = -528.5553$. Table 5.5 shows the Microsoft Excel computation.

The same procedure was carried out by Matlab using the simulated annealing algorithm. The global optimum for the log-likelihood function was found to be best = 528.5553. This optimum was achieved when bestmin = [0.1124 0.0019], which represent the values of $\boldsymbol{\theta}$. In this case the model parameters are equal to those obtained from Microsoft Excel. All of the 100 initial points gave the global optimum solution. Optimization using a standard optimization algorithm indicates that only 20 out of the 100 random initial points obtain the global optimal value. Hence the simulated annealing performs better.

By computing the number of subscribers per week using the optimal values for $\boldsymbol{\theta}$ it was possible to predict the remaining eight weeks of the data. When the predictions were calculated, a plot of the exact number of subscribers, predictions from the Microsoft Excel optimal values and predictions from the Matlab optimal values was constructed to compare the predictions. The prediction values are shown in Table 5.6 and the plot of the predictions is shown in Figure 5.6. This figure shows that the prediction for two segments with fixed probabilities 0.1 and 0.9 is not very accurate, however the predictions obtained from Microsoft Excel and from Matlab are very similar to each other. Figure 5.5 shows the global optimal value at the top right corner of the figure.

Two segments of subscribers with probabilities 0.1 & 0.9					
	Segments:	1	2		
	Thetas:	**0.1124**	**0.0019**		
	Probabilities:	0.1	0.9		
	LL:	**-528.5553**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0130	10	0.9870	-43.4399
2	984	0.0117	6	0.9753	-26.6795
3	978	0.0106	6	0.9647	-27.2847
4	949	0.0096	29	0.9551	-134.7464
5	944	0.0087	5	0.9464	-23.7168
6	937	0.0079	7	0.9385	-33.8667
7	933	0.0072	4	0.9313	-19.7220
8	914	0.0066	19	0.9247	-95.3875
9	911	0.0061	3	0.9186	-15.3229
10	907	0.0056	4	0.9130	-20.7683
11	907	0.0051	0	0.9079	0.0000
12		0.0047			-87.6206
13		0.0044			
14		0.0041			
15		0.0038			
16		0.0036			
17		0.0034			
18		0.0032			
19		0.0030			

Table 5.5: Excel: Two segments of subscribers with probabilities 0.1 & 0.9

$\mathbf{E}(t)$	$\mathbf{M_X}(t)$	$\mathbf{M_M}(t)$
1000	1000	1000
990	987.0153	987.0500
984	975.4491	975.5179
978	965.1150	965.2177
949	955.8539	955.9900
944	947.5291	947.6982
937	940.0227	940.2245
933	933.2330	933.4672
914	927.0719	927.3382
911	921.4627	921.7609
907	916.3390	916.6687
907	911.6426	912.0035
900	907.3228	907.7147
886	903.3353	903.7579
882	899.6411	900.0943
877	896.2062	896.6897
873	893.0007	893.5142
861	889.9983	890.5415
849	887.1757	887.7485
843	884.5126	885.1149

Table 5.6: Exact and Model number of subscribers

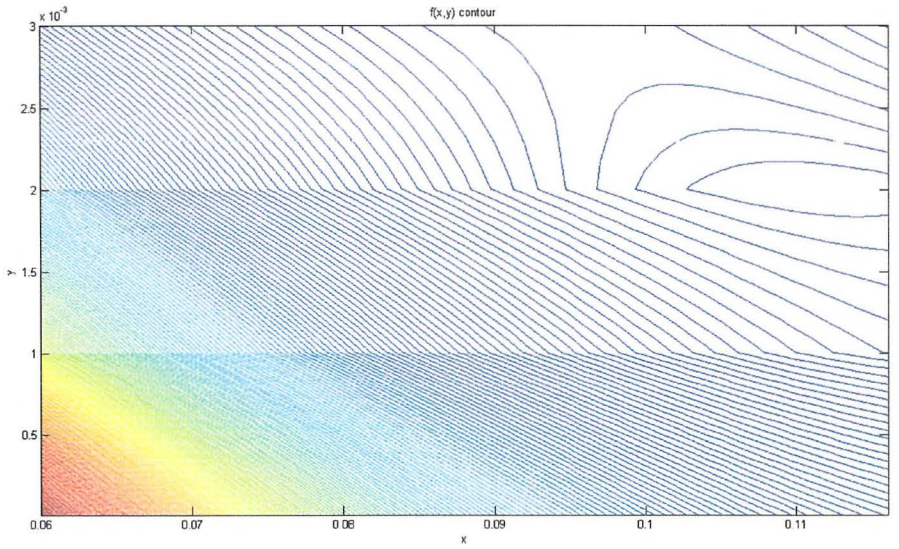

Figure 5.5: Contour figure showing the optimal value

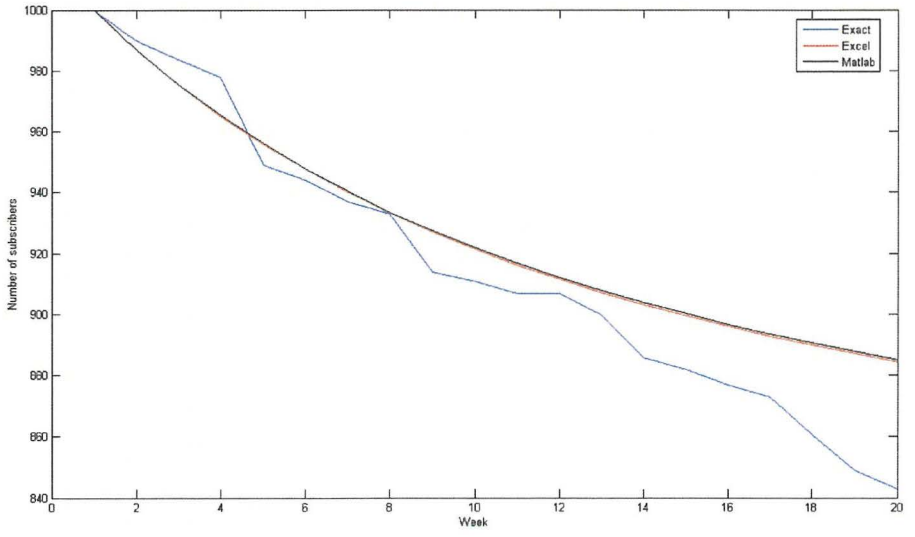

Figure 5.6: Model prediction for two segments with probabilities 0.1 & 0.9

5.3.2.3 Two segments with optimizable probabilities

The third churn model was to divide the data into two segments where the probabilities are optimizable, so that the probabilty of each segment is unknown. An initial point was provided for $\boldsymbol{\theta}$ such that it is within the bounds $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$ and another initial point for **prob** was given such that the bounds are $0 \leq \mathbf{prob} \leq 1$ and $\text{prob}_1 + \text{prob}_2 = 1$. The optimal results from Microsoft Excel show that $\theta_1 = 0.0910$, $\theta_2 = 0.0001$, $\text{prob}_1 = 0.1419$, $\text{prob}_2 = 0.8581$. This means that a higher churn probability is expected for a subscriber in the first segment, however it is more probable that a subscriber is in the second segment. The optimal log-likelihood function is $\text{LL} = -528.2415$. Table 5.7 shows the Microsoft Excel computation.

The global optimum in Matlab for the log-likelihood function is given by best = 528.2268. This optimum is achieved when bestmin = [0.0000 0.0920 0.8581 0.1419], where $\boldsymbol{\theta}$ = [0.0000 0.0920] and **prob** = [0.8581 0.1419], which are the same as those obtained in Microsoft Excel. For this model, the matrix **A** could not be computed since this model optimizes four parameters. Instead, a sum was computed giving the number of initial values which obtain the global optimal value of the log-likelihood function. This sum gave a value of 96 meaning that 96 out of 100 initial values result in the global optimal value.

Predictions for the number of subscribers for the remaining weeks were computed and are shown in Table 5.8. Figure 5.7 shows the plot of the predictions and the exact number of subscribers. This figure shows that this model is not a very accurate model, since the last few predictions diverge from the actual number of subscribers.

Two segments with optimizable probabilities					
	Segments:	1	2		
	Thetas:	**0.0910**	**0.0001**		
	Probabilities:	**0.1419**	**0.8581**	1	
	LL:	**-528.2415**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0130	10	0.9870	-43.4297
2	984	0.0118	6	0.9752	-26.6262
3	978	0.0108	6	0.9644	-27.1941
4	949	0.0098	29	0.9546	-134.1814
5	944	0.0089	5	0.9457	-23.6073
6	937	0.0081	7	0.9376	-33.7112
7	933	0.0074	4	0.9303	-19.6408
8	914	0.0067	19	0.9236	-95.0843
9	911	0.0061	3	0.9175	-15.2956
10	907	0.0056	4	0.9119	-20.7701
11	907	0.0051	0	0.9068	0.0000
12		0.0046			-88.7008
13		0.0042			
14		0.0038			
15		0.0035			
16		0.0032			
17		0.0029			
18		0.0026			
19		0.0024			

Table 5.7: Excel: Two segments with optimizable probabilities

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	987.0021	986.9452
984	975.3326	975.2462
978	964.8426	964.7494
949	955.4020	955.3209
944	946.8968	946.8435
937	939.2269	939.2143
933	932.3039	932.3429
914	926.0499	926.1492
911	920.3958	920.5627
907	915.2803	915.5208
907	910.6489	910.9678
900	906.4531	906.8543
886	902.6496	903.1360
882	899.1996	899.7737
877	896.0684	896.7321
873	893.2250	893.9796
861	890.6416	891.4880
849	888.2931	889.2320
843	886.1569	887.1887

Table 5.8: Exact and Model number of subscribers

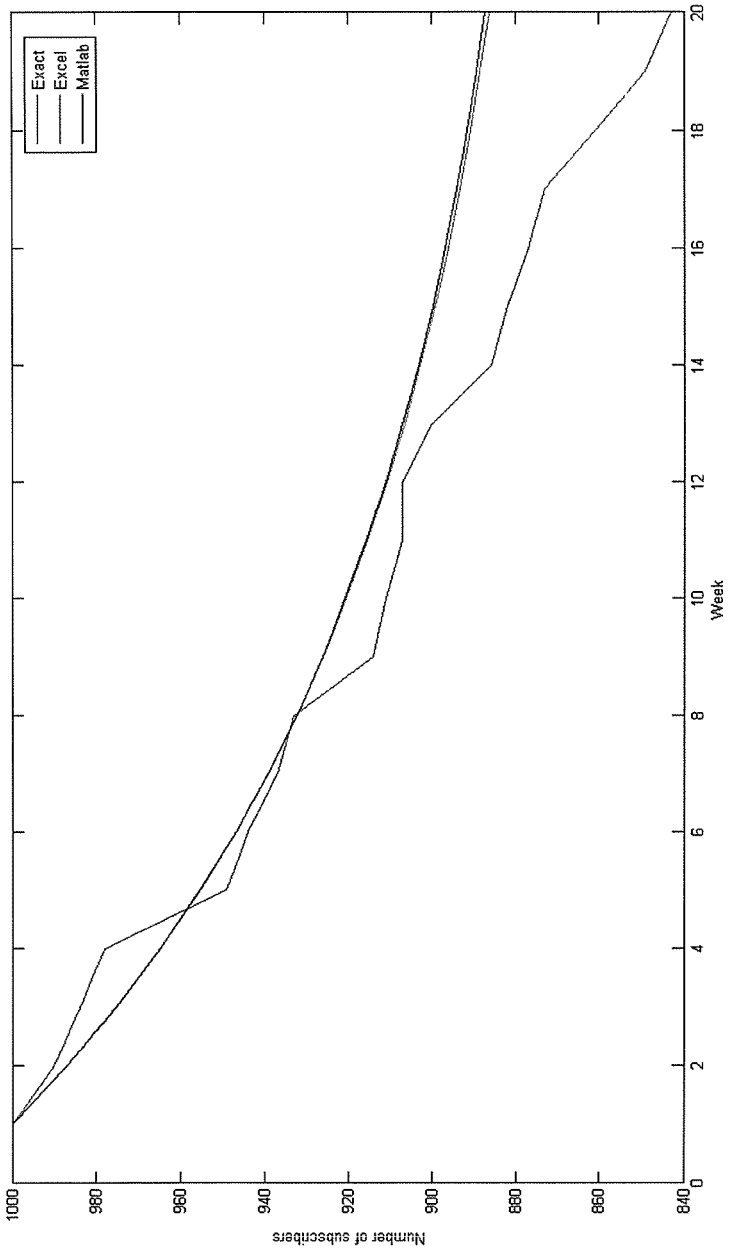

Figure 5.7: Model prediction for two segments with optimizable probabilities

5.3.2.4 Three segments with fixed probabilities 0.1, 0.3 & 0.6

Suppose that the data is divided into three segments with fixed probabilities, such that $\text{prob}_1 = 0.1$, $\text{prob}_2 = 0.3$ and $\text{prob}_3 = 0.6$. An initial starting point for $\boldsymbol{\theta}$ was given hving three components in the region $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. The initial point was entered by trial-and-error until the optimal solution was obtained. The results show that $\theta_1 = 0.1121$, $\theta_2 = 0.0001$, $\theta_3 = 0.0029$. So it is more probable to churn from the first segment, while it is less probable to churn from the second segment. The log-likelihood function corresponding to these optimal values is $LL = -528.5500$, as shown in Table 5.9.

By using the simulated annealing algorithm in Matlab, the global optimum for the log-likelihood function was found to be best = 528.5311. This optimum was achieved when bestmin = [0.1117 0.0060 0.0000] such that $\theta_1 = 0.1117$, $\theta_2 = 0.0060$ and $\theta_3 = 0$. Matlab's optimal solution shows also that it is more likely to churn from the first segment. Again, the matrices $\mathbf{A}$ and $\mathbf{Z}$ could not be computed since this model is in three dimensions. However, a sum of the initial points which resulted into the global optimum was calculated. This sum added up to 998, meaning that 998 out of 1000 random initial points reach the global optimum value.

Predictions for the remaining weeks were computed by using both optimal values found by Microsoft Excel and by Matlab. The results are shown in Table 5.10 and the corresponding plot is shown in Figure 5.8. Again, it can be seen that the predictions give a slight discrepancy from the exact values.

Three segments of subscribers with probabilities 0.1, 0.3 & 0.6					
Segments:	1	2	3		
Thetas:	**0.1121**	**0.0001**	**0.0029**		
Probabilities:	0.1	0.3	0.6	1	
LL:	**-528.5500**				
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0130	10	0.9870	-43.4435
2	984	0.0117	6	0.9753	-26.6796
3	978	0.0106	6	0.9647	-27.2828
4	949	0.0096	29	0.9551	-134.7284
5	944	0.0087	5	0.9464	-23.7123
6	937	0.0079	7	0.9385	-33.8587
7	933	0.0072	4	0.9312	-19.7166
8	914	0.0066	19	0.9246	-95.3586
9	911	0.0061	3	0.9185	-15.3180
10	907	0.0056	4	0.9130	-20.7615
11	907	0.0051	0	0.9078	0.0000
12		0.0047			-87.6901
13		0.0044			
14		0.0041			
15		0.0038			
16		0.0036			
17		0.0034			
18		0.0032			
19		0.0030			

Table 5.9: Excel: Three segments of subscribers with probabilities 0.1, 0.3 & 0.6

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	987.0200	987.0300
984	975.4538	975.4704
987	965.1164	965.1378
949	955.8496	955.8751
944	947.5173	947.5474
937	940.0024	940.0384
933	933.2037	933.2474
914	927.0334	927.0873
911	921.4152	921.4823
907	916.2830	916.3664
907	911.5788	911.6820
900	907.2520	907.3789
886	903.2585	903.4128
882	899.5595	899.7452
877	896.1211	896.3423
873	892.9132	893.1741
861	889.9097	890.2145
849	887.0875	887.4403
843	884.4262	884.8313

Table 5.10: Exact and Model number of subscribers

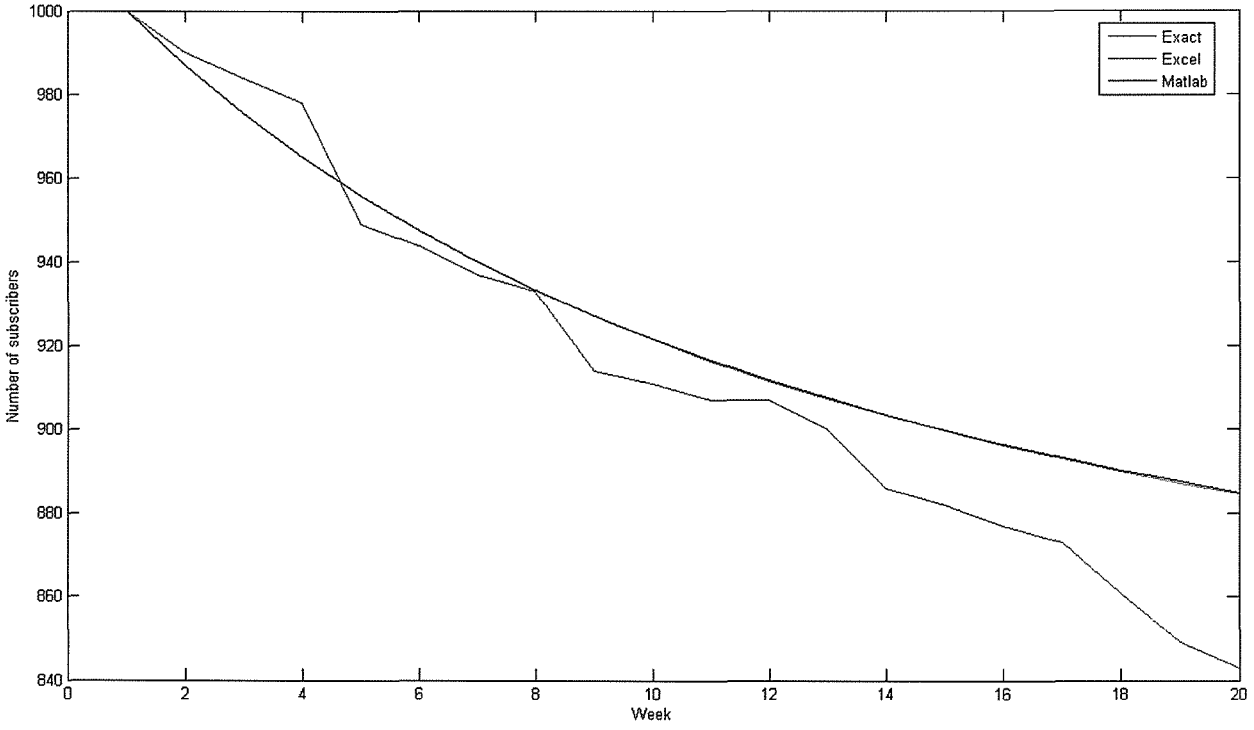

Figure 5.8: Model prediction for three segments with probabilities 0.1, 0.3 & 0.6

5.3.2.5 Three segments with optimizable probabilities

This churn model divides the data into three segments, however this time the probability of being in a segment is not fixed, but the segment probabilities are optimizable. For this model two initial points must be given, one for the probabilties of each segment, **prob**, and one for the model parameters $\boldsymbol{\theta}$. These initial values must be within the bounds $0 \leq \mathbf{prob} \leq 1$, $\text{prob}_1 + \text{prob}_2 + \text{prob}_3 = 1$, and $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. Table 5.11 shows the results obtained from Microsoft Excel, where $\boldsymbol{\theta} = [0.0001\ 0.0925\ 0.9999]$ and $\mathbf{prob} = [0.8597\ 0.1403\ 0.0000]$. The segment probabilities show that it is more likely for a subscriber to be in the first segment with a very small churn probability. Also, the third segment has probability 0, however, if a subscriber is in this segment, the subscriber has a very high chance of churning. The corresponding log-likelihood function is $LL = -528.2402$.

The same procedure was carried out using the simulated annealing algorithm in Matlab. The initial points were set within the bounds of **prob** and $\boldsymbol{\theta}$ as in Microsoft Excel. The global optimum for the log-likelihood function was achieved at best = 528.2391. This results from the model parameters bestmin = [0.0881 0.0958 0.0001 0.0404 0.0995 0.8602], where $\boldsymbol{\theta} = [0.0881\ 0.0958\ 0.0001]$ and $\mathbf{prob} = [0.0404\ 0.0995\ 0.8602]$. On the contrary of the optimal values obtained from Microsoft Excel, these values show that it is more probable of being in the third segment, but the probability of churning from this segment is very small. Once again, a sum corresponding to the number of initial values which result in obtaining the global optimal value, and in this case all initial points reach the optimal solution such that the sum is equal to 1000.

The predictions were once again calculated and Table 5.12 shows the predictions of both optimal values obtained from Microsoft Excel and Matlab. The corresponding figure is shown in Figure 5.9 suggesting that this churn model is not very accurate.

Three segments with optimizable probabilities					
Segments:	1	2	3		
Thetas:	**0.0001**	**0.0925**	**0.9999**		
Probabilities:	**0.8597**	**0.1403**	**0.0000**	1	
LL:	**-528.2402**				
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0131	10	0.9869	-43.3807
2	984	0.0119	6	0.9751	-26.6067
3	978	0.0108	6	0.9643	-27.1845
4	949	0.0098	29	0.9545	-134.1825
5	944	0.0089	5	0.9456	-23.6157
6	937	0.0081	7	0.9376	-33.7344
7	933	0.0073	4	0.9302	-19.6606
8	914	0.0067	19	0.9236	-95.2093
9	911	0.0061	3	0.9175	-15.3202
10	907	0.0055	4	0.9120	-20.8094
11	907	0.0050	0	0.9070	0.0000
12		0.0045			-88.5362
13		0.0041			
14		0.0038			
15		0.0034			
16		0.0031			
17		0.0028			
18		0.0026			
19		0.0023			

Table 5.11: Excel: Three segments with optimizable probabilities

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	986.9363	986.8226
984	975.2280	975.0295
978	964.7210	964.4612
949	955.2810	954.9790
944	946.7904	946.4618
937	939.1462	938.8035
933	932.2576	931.9109
914	926.0447	925.7020
911	920.4367	920.1045
907	915.3709	915.0542
907	910.7918	910.4942
900	906.6498	906.3743
886	902.9008	902.6495
882	899.5055	899.2797
877	896.4288	896.2293
873	893.6390	893.4664
861	891.1082	890.9624
849	888.8108	888.6917
843	886.7243	886.6313

Table 5.12: Exact and Model number of subscribers

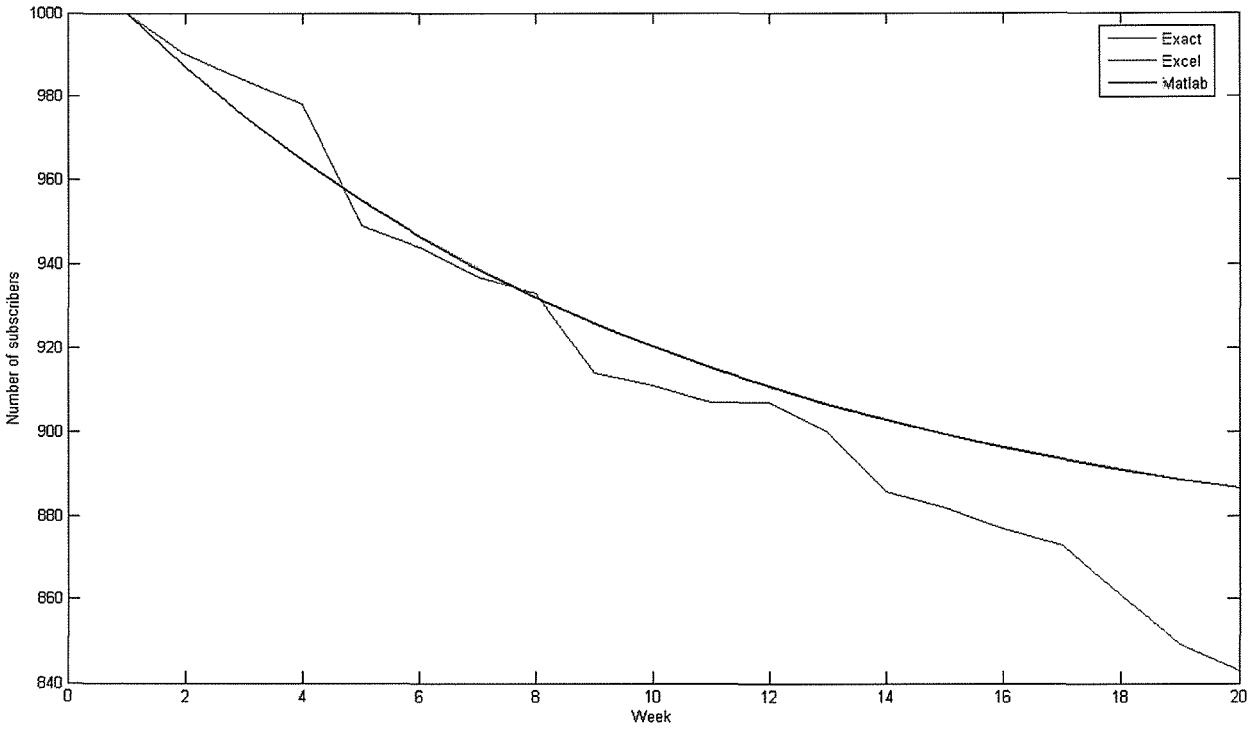

Figure 5.9: Model prediction for three segments with optimizable probabilities

5.3.2.6 Shifted-Beta Geometric Model

The shifted-beta geometric (sBG) model was the last churn model fitted to the data. For this model the first fourteen weeks, $p = 14$, of the data were used to find the optimal model parameters. In this case the data is not divided into segments and so there are no segment probabilities. An initial point was given for α and β which are the parameters of this model. The optimal value for the log-likelihood function was obtained when $\alpha = 0.2405$ and $\beta = 20.7964$, where the corresponding log-likelihood function is given by $LL = -672.1467$. The results by Microsoft Excel are shown in Table 5.13.

By applying the simulated annealing algorithm in Matlab, the optimal values for α and β were calculated such that the optimal model parameters obtained were $\alpha = 0.2319$ and $\beta = 19.8898$ and the corresponding log-likelihood function is given by best $=$ 672.1486. In this case all 100 random initial values resulted in reaching the global optimal solution.

For this model, the predictions for the optimal value obtained from Microsoft Excel and the predictions for the optimal value obtained from Matlab were plotted with the exact number of subscribers per week. Table 5.14 shows the results of the predictions and Figure 5.10 shows the plot of these predictions. From the figure it can be seen that the predictions describe the data quite well. In fact, this churn model gives the best predictions.

IGTT sBG Model					
		Alpha	**0.2405**		
		Beta	**20.7964**		
		LL:	**-672.1467**		
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0114	10	0.9886	-44.7116
2	984	0.0108	6	0.9778	-27.1746
3	978	0.0102	6	0.9676	-27.5067
4	949	0.0097	29	0.9579	-134.4859
5	944	0.0092	5	0.9487	-23.4413
6	937	0.0088	7	0.9399	-33.1596
7	933	0.0084	4	0.9316	-19.1362
8	914	0.0080	19	0.9236	-91.7568
9	911	0.0077	3	0.9159	-14.6189
10	907	0.0073	4	0.9086	-19.6606
11	907	0.0070	0	0.9015	0.0000
12	900	0.0068	7	0.8948	-34.9680
13	886	0.0065	14	0.8883	-70.4718
14	882	0.0063	4	0.8820	-20.2833
15		0.0061			-110.7712
16		0.0058			
17		0.0057			
18		0.0055			
19		0.0053			

Table 5.13: Excel: sBG Model

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	988.5660	988.4751
984	977.8990	977.7475
978	967.9154	967.7272
949	958.5436	958.3380
944	949.7225	949.5148
937	941.3990	941.2015
933	933.5269	933.3495
914	926.0661	925.9168
911	918.9811	918.8663
907	912.2407	912.1654
907	905.8172	905.7854
900	899.6859	899.7009
886	893.8248	893.8889
882	888.2140	888.3293
877	882.8359	883.0038
873	877.6744	877.8958
861	872.7149	872.9906
849	867.9443	868.2747
843	863.3506	863.7358

Table 5.14: Exact and Model number of subscribers

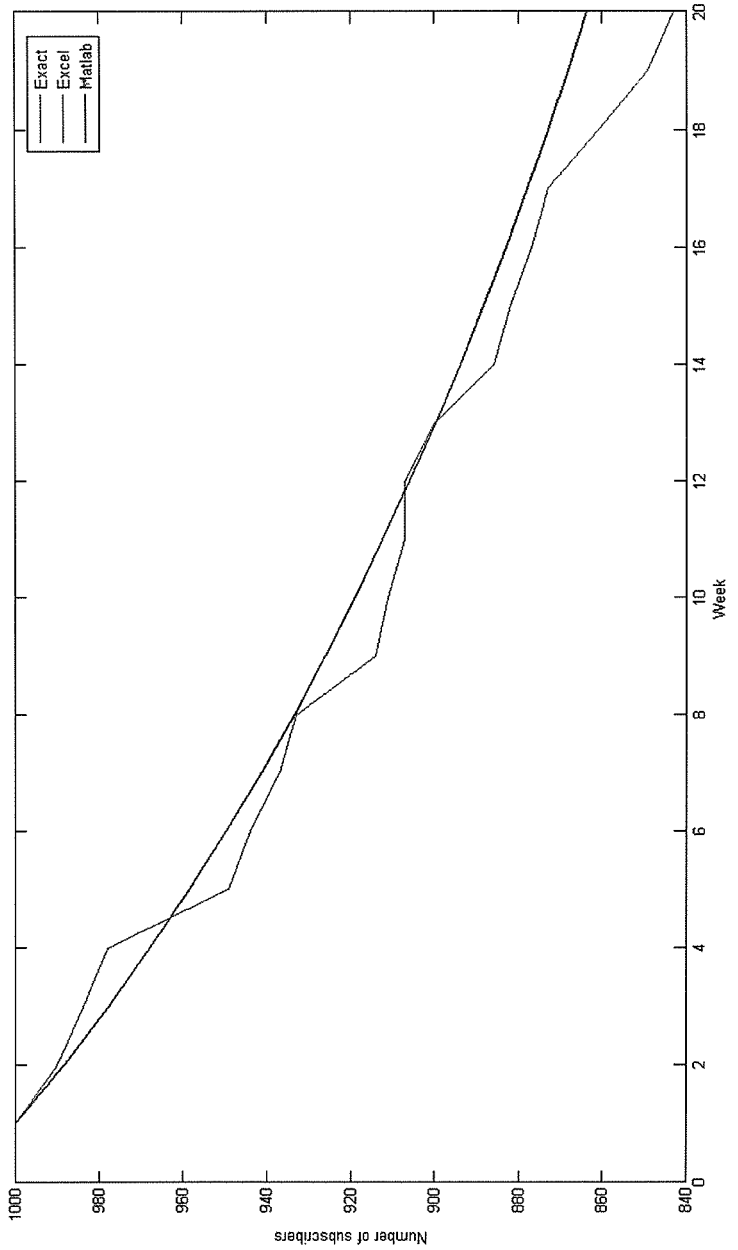

Figure 5.10: Model prediction for sBG model

5.3.3 Friends Tariff

5.3.3.1 Two segments with fixed probabilities 0.4 & 0.6

The churn models described in the previous subsections were again implemented for the Friends tariff. The first churn model for this tariff was to split the data into two segments where the first segment has probability $\text{prob}_1 = 0.4$ and the second segment has probability $\text{prob}_2 = 0.6$. The initial values was selected by trial-and-error in Microsoft Excel until the optimal values were obtained, such that $\theta_1 = 0.0212$, $\theta_2 = 0.0001$, showing that the second segment has a very small churn probability. The maximum log-likelihood function corresponding to these optimal values is LL = -489.0698. Table 5.15 shows the Microsoft Excel computation.

By computing the same model using the simulated annealing algorithm in Matlab, the global optimum for the log-likelihood function is given by best = 489.0611 where the model parameters are bestmin = [0.0213 0.0000], such that $\theta_1 = 0.0213$ and $\theta_2 = 0$. These values also confirm that the churn probability is highest for the first segment. The values of the matrix $\mathbf{A}$ are all 1, meaning that all 100 initial points gave the global optimum value. From the values of the matrix $\hat{\mathbf{A}}$, 74 out of 100 initial points attain the global optimal solution, which is not a bad result however the simulated annealing algorithm performed better.

The optimal values found in Microsoft Excel and Matlab were used to predict the remaining eight weeks. Table 5.16 shows the actual values of the number of subscribers per week and the predicted values obtained from the Microsoft Excel optimal values and from the Matlab optimal values. Figure 5.12 shows the plot of the corresponding optimal values. This figure shows that the predictions are quite reasonable. Figure 5.11 shows the global optimal value at the bottom center of the figure.

Two segments of subscribers with probabilities 0.4 & 0.6					
	Segments:	1	2		
	Thetas:	**0.0212**	**0.0001**		
	Probabilities:	0.4	0.6		
	LL:	**-489.0698**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0085	10	0.9915	-47.6337
2	986	0.0084	4	0.9831	-19.1385
3	982	0.0082	4	0.9749	-19.2236
4	961	0.0080	21	0.9669	-101.3704
5	956	0.0078	5	0.9591	-24.2421
6	951	0.0077	5	0.9514	-24.3484
7	948	0.0075	3	0.9439	-14.6728
8	924	0.0074	24	0.9365	-117.8922
9	920	0.0072	4	0.9293	-19.7337
10	916	0.0071	4	0.9223	-19.8186
11	916	0.0069	0	0.9154	0.0000
12		0.0068			-80.9957
13		0.0066			
14		0.0065			
15		0.0063			
16		0.0062			
17		0.0061			
18		0.0059			
19		0.0058			

Table 5.15: Excel: Two segments of subscribers with probabilities 0.4 & 0.6

$\bar{\mathbf{E}}(\mathbf{t})$	$\bar{\mathbf{M}}_{\mathbf{X}}(\mathbf{t})$	$\bar{\mathbf{M}}_{\mathbf{M}}(\mathbf{t})$
1000	1000	1000
990	991.4632	991.4800
986	983.1774	983.2125
982	975.1337	975.1886
961	967.3236	967.3997
956	959.7391	959.8376
951	952.3723	952.4944
948	945.2157	945.3626
924	938.2623	938.4349
920	931.5051	931.7045
916	924.9375	925.1647
916	918.5533	918.8092
911	912.3464	912.6317
896	906.3108	906.6265
893	900.4412	900.7878
890	894.7319	895.1103
887	889.1780	889.5888
870	883.7744	884.2182
858	878.5164	878.9937
854	873.3992	873.9107

Table 5.16: Exact and Model number of subscribers

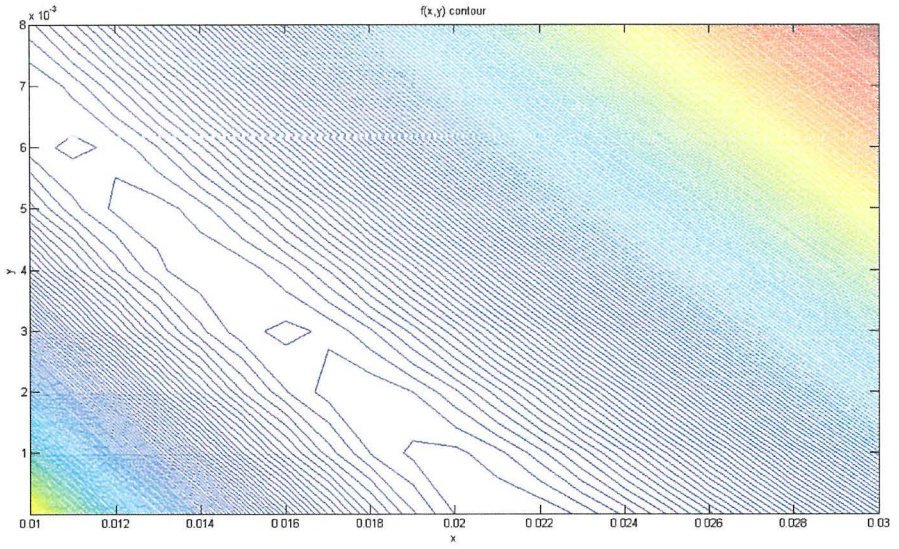

Figure 5.11: Contour figure showing the optimal value

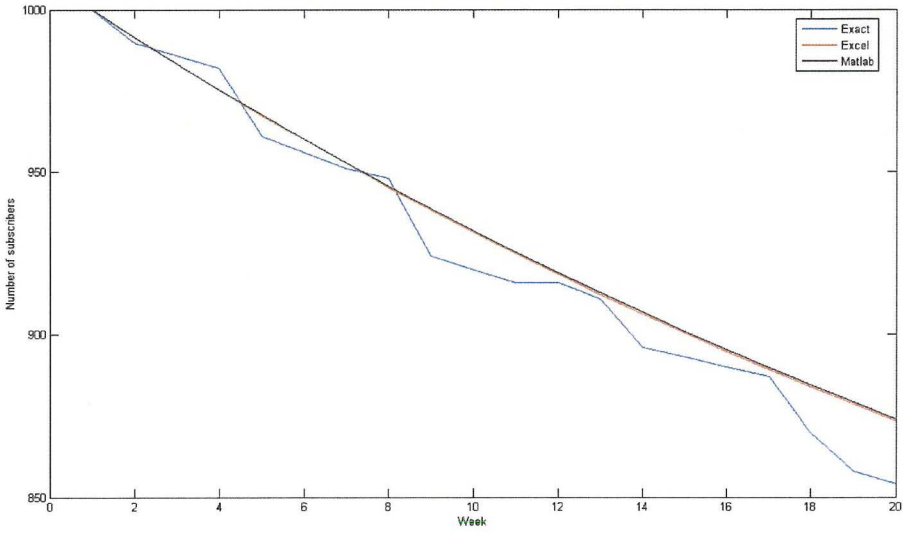

Figure 5.12: Model prediction for two segments with probabilities 0.4 & 0.6

5.3.3.2 Two segments with fixed probabilities 0.1 & 0.9

The next churn model was that of dividing the data into two segments with fixed probabilities where the first segment has probability $\text{prob}_1 = 0.1$ and the second segment has probabililty $\text{prob}_2 = 0.9$. An initial point was provided for $\boldsymbol{\theta}$, such that $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. The optimal values obtained from Microsoft Excel are $\theta_1 = 0.0732$, $\theta_2 = 0.0028$, so the first segment has a higher churn probability, and the maximized log-likelihood function corresponding to these values is $LL = -488.7240$. The computation in Microsoft Excel is shown in Table 5.17.

Similarly, using the simulated annealing algorithm in Matlab, the optimal results show that the global optimum for the log-likelihood function is achieved at best = 488.7240 when the model parameters are bestmin = [0.0733 0.0028]. These are the same values obtained from Microsoft Excel. The matrix $\mathbf{A}$ has 100 elements and in this case they are all 1's. This shows that all 100 initial points gave the optimal solution. The matrix $\hat{\mathbf{A}}$ was again calculated, and only 21 out of 100 initial points obtained the global optimum. Once again, this shows that the simulated annealing algorithm demonstrates an improvement when compared to a standard optimization algorithm, where some initial points end up in a local optimal solution.

The predictions for the remaining eight weeks were calculated and the results are shown in Table 5.18. This table shows the exact number of subscribers per week and the predictions using the optimal values that were obtained from the computation of Microsoft Excel and from Matlab. Figure 5.14 is the plot of both predictions and the actual data. However, from this figure it can be seen that the predictions are not very accurate.

Figure 5.13 shows the global optimal solution at the top right corner.

Two segments of subscribers with probabilities 0.1 & 0.9					
	Segments:	1	2		
	Thetas:	**0.0732**	**0.0028**		
	Probabilities:	0.1	0.9		
	LL:	**-488.7240**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0098	10	0.9902	-46.2300
2	986	0.0093	4	0.9809	-18.7196
3	982	0.0088	4	0.9721	-18.9429
4	961	0.0083	21	0.9638	-100.5997
5	956	0.0079	5	0.9559	-24.2204
6	951	0.0075	5	0.9485	-24.4827
7	948	0.0071	3	0.9414	-14.8434
8	924	0.0068	24	0.9346	-119.9490
9	920	0.0064	4	0.9282	-20.1868
10	916	0.0061	4	0.9221	-20.3770
11	916	0.0059	0	0.9162	0.0000
12		0.0056			-80.1726
13		0.0054			
14		0.0051			
15		0.0049			
16		0.0047			
17		0.0046			
18		0.0044			
19		0.0042			

Table 5.17: Excel: Two segments of subscribers with probabilities 0.1 & 0.9

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	990.1768	990.1500
986	980.9879	980.9360
982	972.3786	972.3031
961	964.2995	964.2016
956	956.7057	956.5866
951	949.5568	949.4175
948	942.8160	942.6573
924	936.4499	936.2726
920	930.4279	930.2329
916	924.7226	924.5103
916	919.3086	919.0797
911	914.1630	913.9180
896	909.2647	909.0040
893	904.5946	904.3186
890	900.1349	899.8441
887	895.8698	895.5644
870	891.7843	891.4647
858	887.8650	887.5314
854	884.0996	883.7522

Table 5.18: Exact and Model number of subscribers

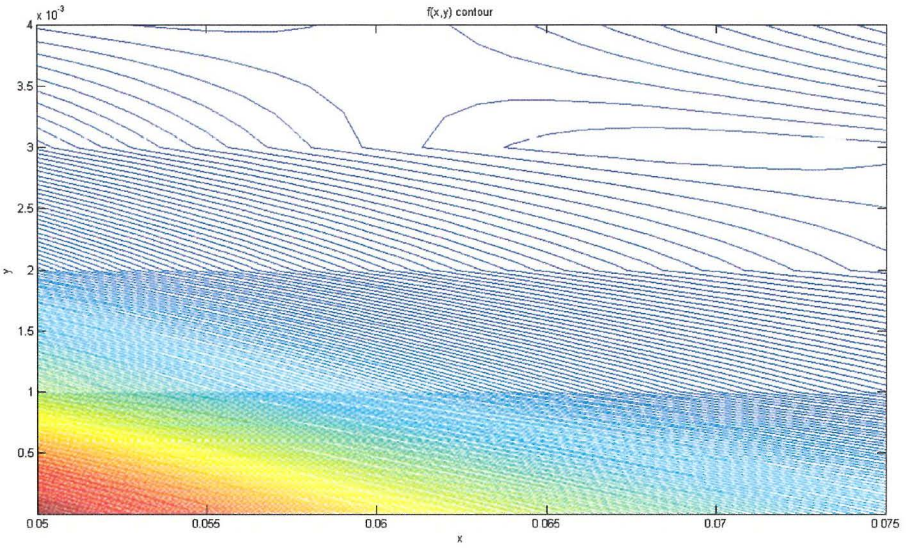

Figure 5.13: Contour figure showing the optimal value

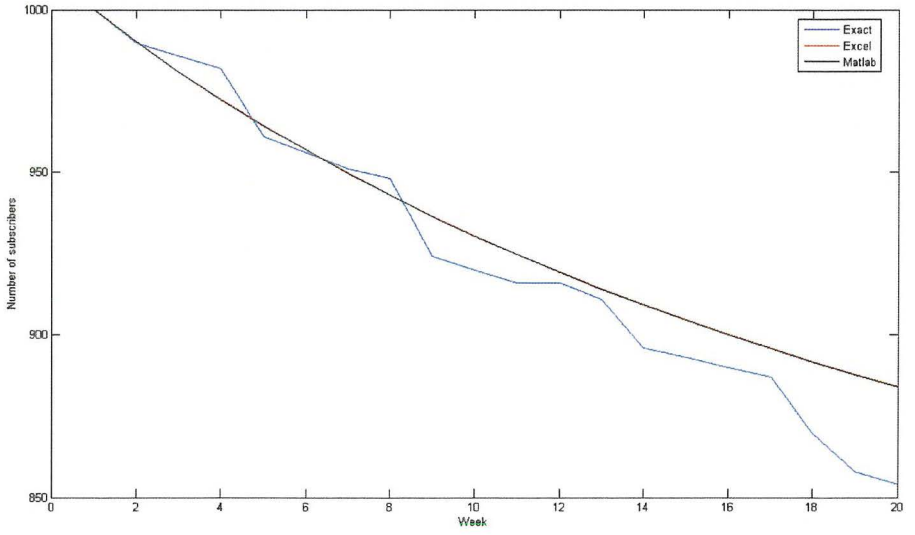

Figure 5.14: Model prediction for two segments with probabilities 0.1 & 0.9

5.3.3.3 Two segments with optimizable probabilities

The third churn model was to divide the data into two segments where the probabilities are optimizable, so that initially the probabilty of each segment is unknown. An initial point was provided for $\boldsymbol{\theta}$ such that it is within the bounds $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$ and another initial point for **prob** was given such that the bounds are $0 \leq \mathbf{prob} \leq 1$ and $\text{prob}_1 + \text{prob}_2 = 1$. The optimal results from Microsoft Excel show that $\theta_1 = 0.0023$, $\theta_2 = 0.0697$, $\text{prob}_1 = 0.8903$, $\text{prob}_2 = 0.1097$. This means that it is more probable for a subscriber to be in the first segment with a lower churn probability than the second segment. The optimal log-likelihood function corresponding to these values is $\text{LL} = -488.7149$. Table 5.19 shows the Microsoft Excel computation.

The global optimum in Matlab for the log-likelihood function is given by best = 488.9162. This optimum is achieved when bestmin = [0.0053 0.0955 0.9544 0.0456], where $\boldsymbol{\theta}$ = [0.0053 0.0955] and **prob** = [0.9544 0.0456]. This confirms that it is more probable for a subscriber to be in the first segment with a lower churn probability than the second segment. The number of initial values which obtain the global optimal value of the log-likelihood function is 75, meaning that 75 out of 100 random initial values result in the global optimal value.

Predictions for the number of subscribers for the remaining weeks are computed and are shown in Table 5.20. Figure 5.15 shows the plots of the predictions and the exact number of subscribers. This figure shows that this model is quite reasonable, and moreover, the predictions from Matlab are more accurate than the predictions in Microsoft Excel.

Two segments with optimizable probabilities					
	Segments:	1	2		
	Thetas:	**0.0023**	**0.0697**		
	Probabilities:	**0.8903**	**0.1097**	1	
	LL:	**-488.7149**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0097	10	0.9903	-46.3609
2	986	0.0092	4	0.9811	-18.7723
3	982	0.0087	4	0.9725	-18.9970
4	961	0.0082	21	0.9643	-100.8947
5	956	0.0078	5	0.9565	-24.2943
6	951	0.0074	5	0.9492	-24.5614
7	948	0.0070	3	0.9422	-14.8941
8	924	0.0066	24	0.9356	-120.3875
9	920	0.0063	4	0.9293	-20.2662
10	916	0.0060	4	0.9233	-20.4635
11	916	0.0057	0	0.9175	0.0000
12		0.0055			-78.8230
13		0.0052			
14		0.0050			
15		0.0048			
16		0.0046			
17		0.0044			
18		0.0042			
19		0.0041			

Table 5.19: Excel: Two segments with optimizable probabilities

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	990.3044	990.5869
986	981.2349	981.7009
982	972.7390	973.2901
961	964.7697	965.3083
956	957.2837	957.7145
951	950.2422	950.4722
948	943.6096	943.5488
924	937.3535	936.9151
920	931.4442	930.5453
916	925.8549	924.4161
916	920.5608	918.5067
911	915.5394	912.7984
896	910.7698	907.2745
893	906.2332	901.9200
890	901.9120	896.7214
887	897.7904	891.6665
870	893.8535	886.7445
858	890.0879	881.9455
854	886.4811	877.2607

Table 5.20: Exact and Model number of subscribers

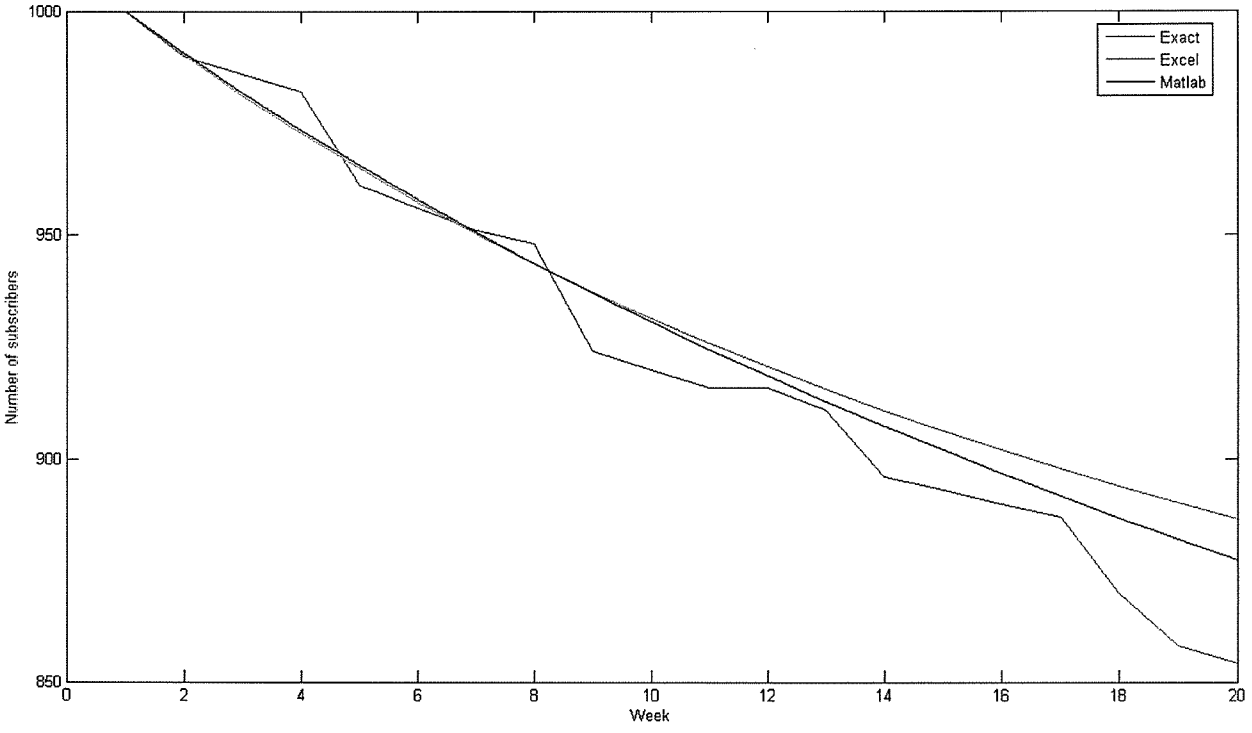

Figure 5.15: Model prediction for two segments with optimizable probabilities

5.3.3.4 Three segments with fixed probabilities 0.1, 0.3 & 0.6

Suppose that the data is divided into three segments with fixed probabilities, such that $\text{prob}_1 = 0.1$, $\text{prob}_2 = 0.3$ and $\text{prob}_3 = 0.6$. An initial starting point for $\boldsymbol{\theta}$ was given where $\boldsymbol{\theta}$ has three components in the region $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. The initial point was entered by trial-and-error until the optimal solution was obtained. The results show that $\theta_1 = 0.0727$, $\theta_2 = 0.0001$, $\theta_3 = 0.0042$. Thus it is more probable for a subscriber in the second segment to churn. The log-likelihood function corresponding to these optimal values is $LL = -488.7182$ as shown in Table 5.21.

By using the simulated annealing algorithm in Matlab, the global optimum for the log-likelihood function was found to be best = 488.7263. This optimum was achieved when bestmin = [0.0707 0.0018 0.0035] such that $\theta_1 = 0.0707$, $\theta_2 = 0.0018$ and $\theta_3 = 0.0035$. These values also show that the second segment has the highest churn probability. In this case, the matrices **A** and **Z** could not be computed since this model is in three dimensions. However, a sum of the initial points which resulted into the global optimum was calculated. This sum added up to 988, meaning that 988 out of 1000 initial points reach the global optimum value.

Predictions for the remaining weeks were computed by using both optimal values found by Microsoft Excel and Matlab. The results are shown in Table 5.22 and the corresponding plot is shown in Figure 5.16. This figure shows that this churn model does not provide very accurate results.

Three segments of subscribers with probabilities 0.1, 0.3 & 0.6					
Segments:	1	2	3		
Thetas:	**0.0727**	**0.0001**	**0.0042**		
Probabilities:	0.1	0.3	0.6	1	
LL:	**-488.7182**				
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0098	10	0.9902	-46.2369
2	986	0.0093	4	0.9809	-18.7209
3	982	0.0088	4	0.9721	-18.9429
4	961	0.0083	21	0.9638	-100.5941
5	956	0.0079	5	0.9559	-24.2180
6	951	0.0075	5	0.9485	-24.4794
7	948	0.0071	3	0.9414	-14.8411
8	924	0.0068	24	0.9346	-119.9292
9	920	0.0064	4	0.9282	-20.1835
10	916	0.0061	4	0.9220	-20.3740
11	916	0.0059	0	0.9162	0.0000
12		0.0056			-80.1981
13		0.0054			
14		0.0051			
15		0.0049			
16		0.0047			
17		0.0046			
18		0.0044			
19		0.0042			

Table 5.21: Excel: Three segments of subscribers with probabilities 0.1, 0.3 & 0.6

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	990.1835	990.2900
986	980.9975	981.1775
982	972.3883	972.6128
961	964.3069	964.5507
956	956.7094	956.9504
951	949.5558	949.7745
948	942.8099	942.9892
924	936.4385	936.5637
920	930.4117	930.4698
916	924.7022	924.6818
916	919.2850	919.1761
911	914.1374	913.9314
896	909.2384	908.9278
893	904.5692	904.1473
890	900.1123	899.5735
887	895.8517	895.1910
870	891.7728	890.9859
858	887.8621	886.9454
854	884.1075	883.0576

Table 5.22: Exact and Model number of subscribers

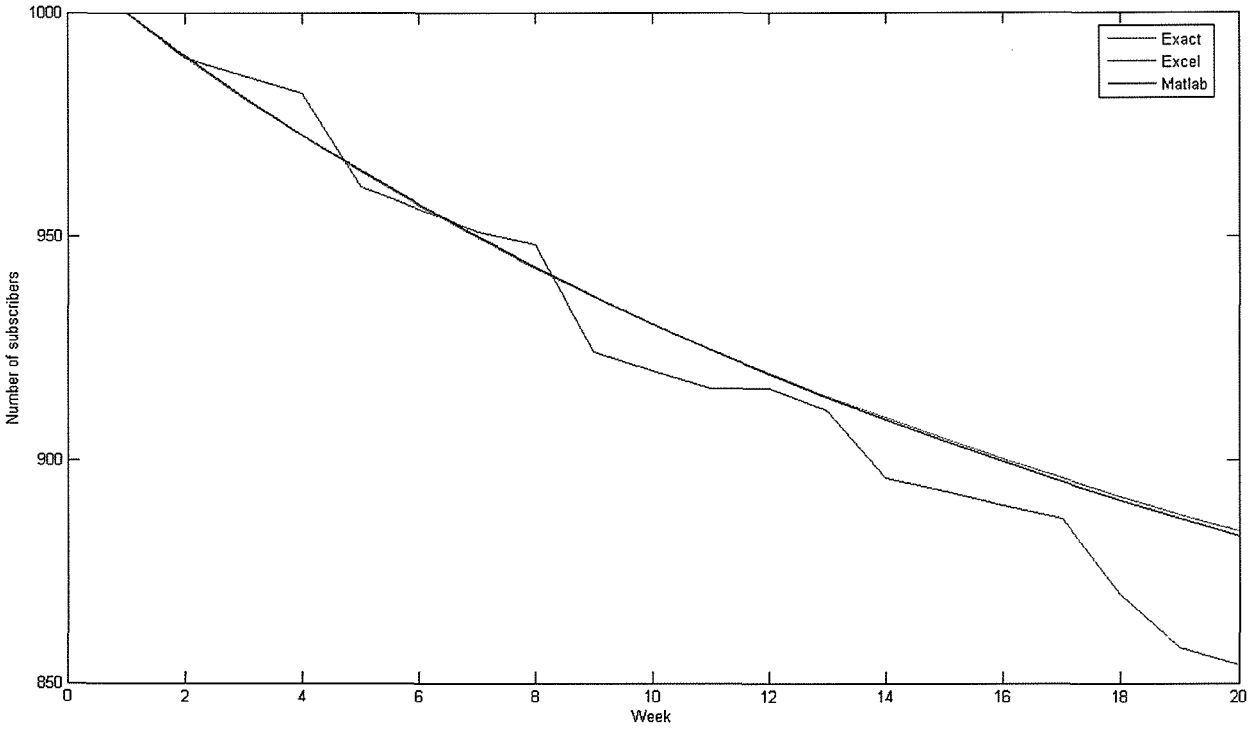

Figure 5.16: Model prediction for three segments with probabilities 0.1, 0.3 & 0.6

5.3.3.5 Three segments with optimizable probabilities

This churn model divides the data into three segments however this time the probability of being in a segment is not fixed but it can be optimized. For this model two initial points must be given, one for the probabilties of each segment **prob**, and one for the model parameters $\boldsymbol{\theta}$. These initial values must be within the bounds $0 \leq \mathbf{prob} \leq 1$, $\text{prob}_1 + \text{prob}_2 + \text{prob}_3 = 1$, and $0.0001 \leq \boldsymbol{\theta} \leq 0.9999$. Table 5.23 shows the results obtained from Microsoft Excel. In particular, the probability of being in the third segment is 0, and it is more probable for a subscriber to be in the first segment. On the other hand, the first segment has a very small churn probability, while the third segment has a high churn probability.

The same procedure was carried out using the simulated annealing algorithm in Matlab. The initial points were set within the bounds of **prob** and $\boldsymbol{\theta}$. The global optimum for the log-likelihood function was achieved at best = 488.6808. This results from the model parameters bestmin = [0.0881 0.0958 0.0001 0.0404 0.0995 0.8602], where $\boldsymbol{\theta}$ = [0.0005 0.0188 0.0720] and **prob** = [0.7614 0.1442 0.0944]. These show that it is most probable for a subscriber to be in the first segment with the least churn probability. The number of initial values which result in obtaining the global optimal value is 968, meaning that 968 out of 1000 initial points result in the global optimum value.

The predictions were once again calculated and Table 5.24 shows the predictions of both optimal values obtained from Microsoft Excel and Matlab. The corresponding figure is shown in Figure 5.17. This figure also suggests that this churn model is not very accurate to predict the number of subscribers in the future. However, the predictions obtained from Matlab give a more accurate result than those obtained from Microsoft Excel.

Three segments of subscribers with optimizable probabilies					
Segments:	1	2	3		
Thetas:	**0.0001**	**0.0540**	**0.8942**		
Probabilities:	**0.8182**	**0.1818**	**0.0000**	1	
LL:	**-488.5951**				
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0099	10	0.9901	-46.1499
2	986	0.0094	4	0.9807	-18.6802
3	982	0.0089	4	0.9719	-18.9003
4	961	0.0084	21	0.9635	-100.3817
5	956	0.0079	5	0.9555	-24.1753
6	951	0.0075	5	0.9480	-24.4500
7	948	0.0071	3	0.9409	-14.8347
8	924	0.0067	24	0.9341	-119.9947
9	920	0.0064	4	0.9278	-20.2185
10	916	0.0060	4	0.9217	-20.4377
11	916	0.0057	0	0.9160	0.0000
12		0.0054			-80.3721
13		0.0051			
14		0.0049			
15		0.0046			
16		0.0044			
17		0.0041			
18		0.0039			
19		0.0037			

Table 5.23: Excel: Three segments with optimizable probabilies

$\mathbf{E}(t)$	$\mathbf{M_X}(t)$	$\mathbf{M_M}(t)$
1000	1000	1000
990	990.0977	990.1115
986	980.8187	980.8560
982	972.1188	972.1817
961	963.9576	964.0418
956	956.2978	956.3936
951	949.1050	949.1985
948	942.3478	942.4214
924	935.9969	936.0304
920	930.0255	929.9961
916	924.4085	924.2919
916	919.1229	918.8935
911	914.1474	913.7785
896	909.4620	908.9266
893	905.0485	904.3189
890	900.8896	899.9382
887	896.9695	895.7689
870	893.2733	891.7963
858	889.7873	888.0070
854	886.4985	884.3888

Table 5.24: Exact and Model number of subscribers

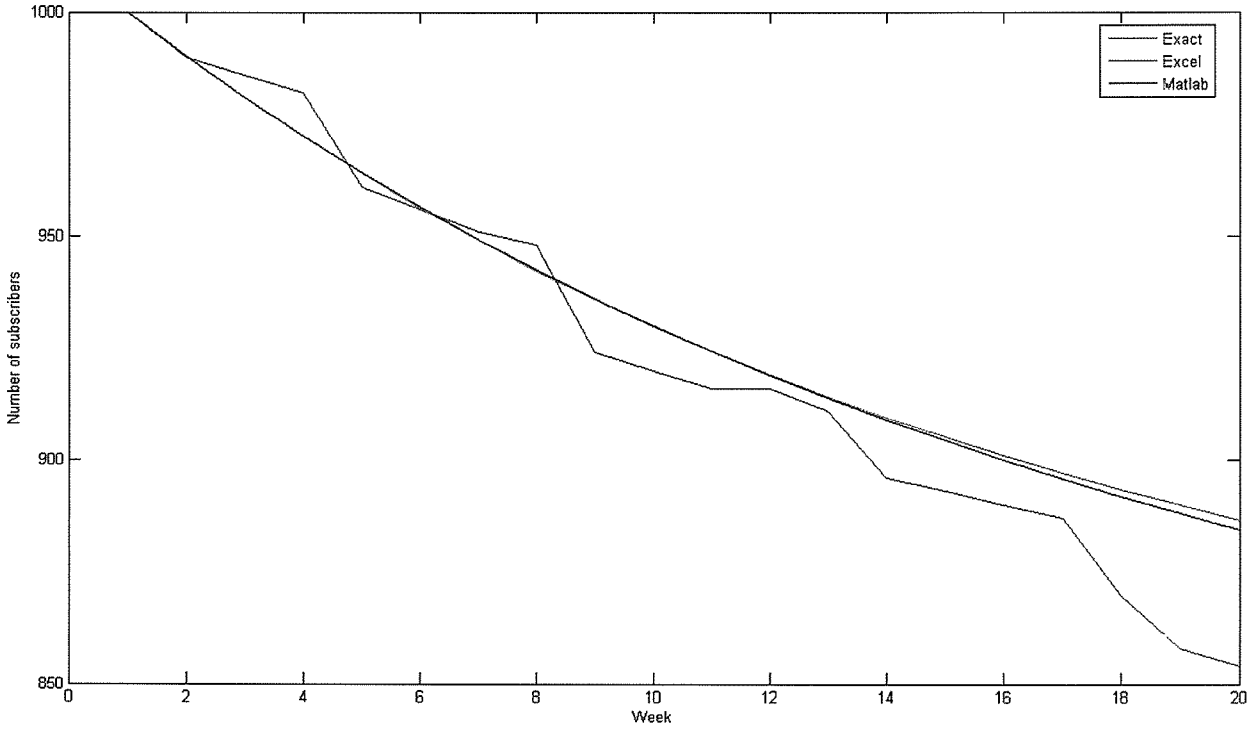

Figure 5.17: Model prediction for three segments with optimizable probabilities

5.3.3.6 Shifted-Beta Geometric Model

The shifted-beta geometric (sBG) model was the last churn model fitted to the data. For this model the first fourteen weeks of the data were used to find the optimal model parameters. In this case the data is not divided into segments and so there are no segment probabilities. An initial point was given for α and β which are the parameters of this model. The optimal value for the log-likelihood function was obtained when $\alpha = 0.4530$ and $\beta = 49.5971$, where the corresponding log-likelihood function is given by $LL = -621.9530$. The results by Microsoft Excel are shown in Table 5.25.

By applying the simulated annealing algorithm in Matlab, the optimal values for α and β were calculated. The optimal model parameters obtained were $\alpha = 0.3653$ and $\beta = 38.9580$ and the corresponding log-likelihood function is given by best $= 621.9698$. All 100 initial values obtained the global optimal solution, where the difference in the values of the log-likelihood function are negligible.

For this model, the predictions for the optimal values obtained from Microsoft Excel and the predictions for the optimal values obtained from Matlab were plotted with the exact number of subscribers per week. Table 5.26 shows the results of the predictions and Figure 5.18 shows the plot of these predictions. From this figure it can be seen that the predictions explain the data very well and in fact, this churn model gives the best predictions.

Friends sBG Model					
	Alpha	**0.4530**			
	Beta	**49.5971**			
	LL:	**-621.9530**			
Week t	$\mathbf{N_t}$	$\mathbf{P[T=t]}$	$\mathbf{n_t}$	$\mathbf{S[T=t]}$	**LL term**
0	1000		0		
1	990	0.0091	10	0.9909	-47.0489
2	986	0.0088	4	0.9822	-18.9351
3	982	0.0085	4	0.9736	-19.0483
4	961	0.0083	21	0.9653	-100.5869
5	956	0.0081	5	0.9572	-24.0855
6	951	0.0079	5	0.9493	-24.2193
7	948	0.0077	3	0.9417	-14.6104
8	924	0.0075	24	0.9342	-117.5020
9	920	0.0073	4	0.9269	-19.6851
10	916	0.0071	4	0.9198	-19.7847
11	916	0.0069	0	0.9128	0.0000
12	911	0.0068	5	0.9061	-24.9738
13	896	0.0066	15	0.8995	-75.2768
14	893	0.0065	3	0.8930	-15.1253
15		0.0063			-101.0709
16		0.0062			
17		0.0060			
18		0.0059			
19		0.0058			

Table 5.25: Excel: sBG Model

$\mathbf{E(t)}$	$\mathbf{M_X(t)}$	$\mathbf{M_M(t)}$
1000	1000	1000
990	990.9491	990.7103
986	982.2355	981.8186
982	973.8395	973.2978
961	965.7433	965.1234
956	957.9303	957.2732
951	950.3850	949.7266
948	943.0932	942.4651
924	936.0416	935.4714
920	929.2180	928.7300
916	922.6108	922.2262
916	916.2092	915.9467
911	910.0035	909.8792
896	903.9840	904.0122
893	898.1422	898.3352
890	892.4698	892.8383
887	886.9591	887.5123
870	881.6030	882.3487
858	876.3945	877.3396
854	871.3274	872.4776

Table 5.26: Exact and Model number of subscribers

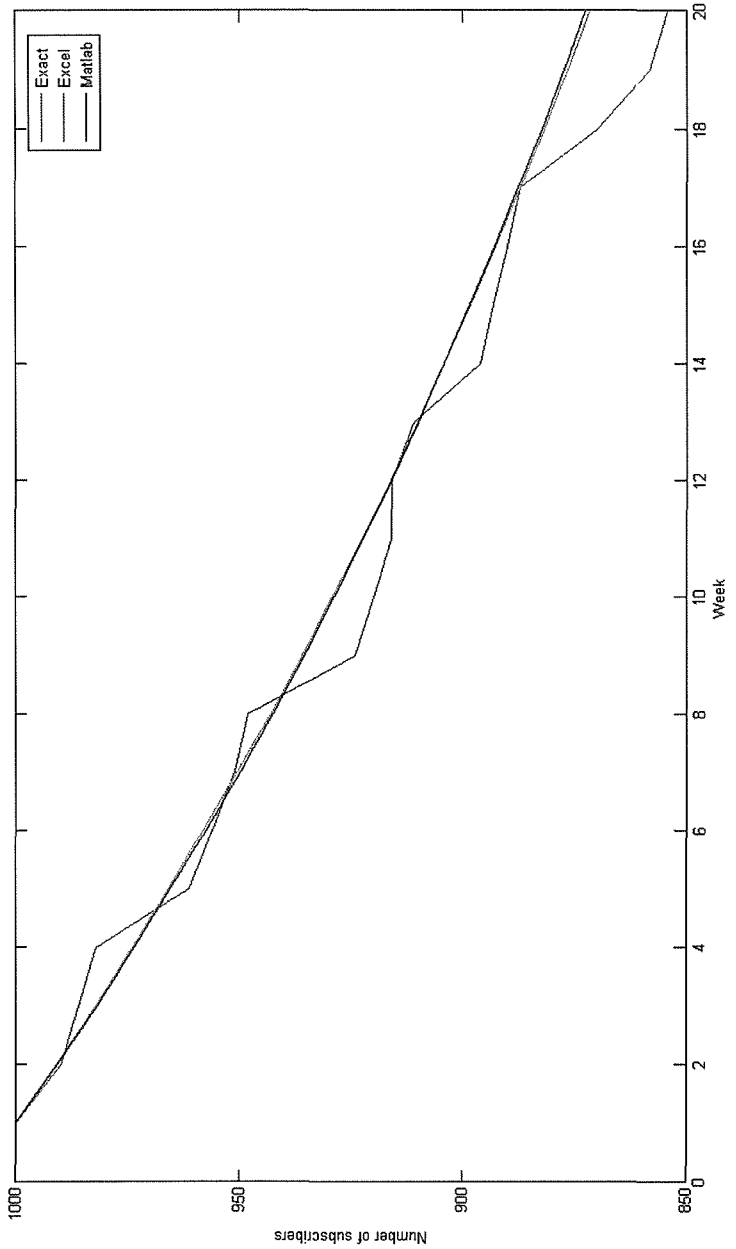

Figure 5.18: Model prediction for sBG model

Chapter 6

Conclusions

6.1 Presentation of major results

The difference between the algorithm provided in Microsoft Excel and the simulated annealing applied in Matlab was very evident during the implementation phase of this dissertation. The initial values entered in Microsoft Excel had to be entered by trial-and-error until the optimal values were achieved, whereas for the simulated annealing algorithm, for the majority of initial value gives the optimal values. There were only a very few cases in which the global optimization algorithm did not provide the optimal values as can be seen from the matrix $\mathbf{A}$ and the sum of the initial points which resulted in the optimal solutions. So it is true that the global optimization algorithm performs better than other constrained nonlinear optimization algorithms. For the cases when the data was divided into two segments with fixed probabilities, optimization was also computed with a standard optimization algorithm. The standard optimization algorithm `fmincon` showed that the algorithm does not always result in the global optimum, but in most cases it reaches a local optimum, and so, the simulated annealing

algorithm performed better than the standard optimization algorithm.

The following is a summary of the values obtained from the churn models, in particular to show which is the most accurate model. This can be done by taking the average of the absolute difference between the predicted values and the actual values of the subscriber base. The smaller the average is, implies that the churn model is more accurate.

The first churn model applied to the data was that of dividing the subscriber base into two segments with fixed probabilities 0.4 and 0.6. For the IGTT tariff, the average difference between the predicted values and actual values resulting from the Microsoft Excel optimal values is of 8 subscribers, while the average difference from Matlab's optimal values is of 9 subscribers. On the other hand, for the Friends tariff, the difference resulting from both Microsoft Excel and Matlab is of 7 subscribers.

For the second churn model, the data was divided into two segments with fixed probabilities 0.1 and 0.9. For the IGTT tariff, the average difference from Microsoft Excel is of 13 subscribers, and the difference from Matlab is of 14 subscribers. For the Friends tariff, both averages from Microsoft Excel and Matlab give a difference of 9 subscribers.

The next churn model was such that it optimizes both the segment probabilities and the churn probabilities. The average difference between the predicted values and the actual values for the IGTT tariff resulted in 13 subscribers both from the Microsoft Excel and Matlab optimal values. For the Friends tariff, a difference of 10 subscribers resulted from the Microsoft Excel optimal values, and a difference of 8 subscribers resulted from the Matlab optimal values.

The fourth model divided the data into three segments with fixed segment probabilities 0.1, 0.3 and 0.6. For the IGTT tariff, both difference from Microsoft Excel and

Matlab are of 13 subscribers. Whereas for the Friends tariff both Microsoft Excel and Matlab resulted with a difference of 9 subscribers.

For three segments where both the segment probabilities and churn probabilities are optimized, an average difference of 13 subscribers was obtained from both Microsoft Excel and Matlab for the IGTT tariff. A difference of 10 subscribers was also obtained from Microsoft Excel for the Friends tariff, whereas a difference of 9 subscribers was obtained from Matlab.

The last churn model applied to both tariffs was that of the shifted-beta geometric (sBG) model. An average difference of 7 subscribers resulted between the actual data and the predictions from Microsoft Excel and Matlab for the IGTT data, while an average difference of 6 subscribers resulted from the predictions of Microsoft Excel and Matlab for the Friends tariff.

This shows that the most accurate churn model is the shifted-beta-geometric (sBG) model, since it has the smallest difference between the actual data and the predictions.

6.2 Underlining of limitations

Even though a global optimization algorithm was applied to the data in the Matlab computations, it still does not guarantee that the algorithm results in the global optimal solution for all initial values. The values of the matrix **A** show that not all 100 random initial points always attain the global solution. Also, since the data provided only showed the number of subscribers for 20 weeks, a longer observation period will surely help to obtain a more accurate prediction.

6.3 Implications for future research

In this dissertation only two tariffs were used for the implementation part. Future implementations can focus on more tariffs and may also include post-paid/contract bound subscribers. Since no indication is given on whether the subscribers churn at the customer level or else at the contract level, future studies might consider taking into account churning at a contract level and check whether there are any similarities between the new tariffs that the churned subscribers are attracted to. Also, other churn models may be implemented in future research.

Appendix A

Data Set

IGTT Tariff										
Segment	0	1	2	3	4	5	6	7	8	9
0	1000	1000	998	997	940	935	933	931	881	880
1	1000	974	962	949	946	938	928	922	908	905
2	1000	995	990	988	947	945	942	941	915	913
3	1000	997	990	985	931	926	916	907	881	877
4	1000	995	993	989	948	944	939	935	924	920
5	1000	986	981	975	973	968	965	962	960	956
6	1000	992	967	948	944	928	899	894	870	865
Segment	10	11	12	13	14	15	16	17	18	19
0	879	879	878	878	877	876	875	875	792	792
1	899	899	884	881	873	861	859	842	830	819
2	912	912	909	882	881	878	875	864	860	858
3	873	873	862	836	828	827	824	822	814	805
4	916	916	912	910	907	904	899	865	861	858
5	950	950	945	931	926	921	916	907	903	897
6	859	859	845	845	841	834	824	818	816	799

Friends Tariff										
Segment	0	1	2	3	4	5	6	7	8	9
0	1000	1000	1000	999	897	897	897	897	860	859
1	1000	972	960	952	949	937	925	917	896	886
2	1000	999	997	996	963	961	960	959	933	932
3	1000	997	992	990	968	962	957	954	921	917
4	1000	997	996	994	991	989	987	985	968	966
5	1000	981	975	970	967	963	958	955	948	943
6	1000	990	988	977	973	961	948	941	862	851
Segment	10	11	12	13	14	15	16	17	18	19
0	859	859	859	858	858	858	857	857	758	758
1	880	880	870	864	858	852	844	836	827	820
2	930	930	929	899	898	897	896	881	877	876
3	911	911	904	884	882	879	875	844	839	833
4	962	962	958	941	939	936	933	920	918	915
5	938	938	933	929	928	924	920	907	902	899
6	844	844	828	803	799	789	778	714	699	691

Appendix B

Matlab

```
% IGTT: 2 segments of subscribers with probabilities 0.4 & 0.6
periods = 0:11;
prob = [0.4 0.6];
customers = [1000 990 984 978 949 944 937 933 914 911 907 907];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator1(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
```

```
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLn(x,prob,periods,customers,lost);
```

```
% IGTT: 2 segments of subscribers with probabilities 0.1 & 0.9
periods = 0:11;
prob = [0.1 0.9];
customers = [1000 990 984 978 949 944 937 933 914 911 907 907];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator1(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLn(x,prob,periods,customers,lost);
```

```
% IGTT: 3 segments of subscribers with probabilities 0.1 0.3 & 0.6
periods = 0:11;
prob = [0.1 0.3 0.6];
customers = [1000 990 984 978 949 944 937 933 914 911 907 907];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator1(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLn(x,prob,periods,customers,lost);
```

```
% IGTT: 2 & 3 segments of subscribers with optimizable probabilities
periods = 0:11;
customers = [1000 990 984 978 949 944 937 933 914 911 907 907];
lost(1) = 0;
for i=2:length(customers)
```

```
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator3(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) objective2(x,periods,customers,lost);
```

```
% IGTT: sBG model
customers = [1000 990 984 978 949 944 937 933 914 911 907 907 900 886 882];
week = 0:length(customers);
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generatorsbg(x,10),...
        'InitTemp',10,...
```

```
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLsbg(x,week,customers,lost);
```

```
% Friends: 2 segments of subscribers with probabilities 0.4 & 0.6
periods = 0:11;
prob = [0.4 0.6];
customers = [1000 990 986 982 961 956 951 948 924 920 916 916];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator1(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
```

```
        'Verbosity',1);
f = @(x) -LLn(x,prob,periods,customers,lost);
```

```
% Friends: 2 segments of subscribers with probabilities 0.1 & 0.9
periods = 0:11;
prob = [0.1 0.9];
customers = [1000 990 986 982 961 956 951 948 924 920 916 916];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator1(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLn(x,prob,periods,customers,lost);
```

```
% Friends: 3 segments of subscribers with probabilities 0.1 0.3 & 0.6
```

```
periods = 0:11;
prob = [0.1 0.3 0.6];
customers = [1000 990 986 982 961 956 951 948 924 920 916 916];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator1(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLn(x,prob,periods,customers,lost);
```

```
% Friends: 2 & 3 segments of subscribers with optimizable probabilities
periods = 0:11;
customers = [1000 990 986 982 961 956 951 948 924 920 916 916];
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
```

```
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generator3(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) objective2(x,periods,customers,lost);
```

```
% Friends: sBG model
customers = [1000 990 986 982 961 956 951 948 924 920 916 916];
week = 0:length(customers);
lost(1) = 0;
for i=2:length(customers)
    lost(i) = customers(i-1) - customers(i);
end
opt = struct(...
        'CoolSched',@(T) (.9*T),...
        'Generator',@(x) generatorsbg(x,10),...
        'InitTemp',10,...
        'MaxConsRej',1000,...
```

```
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
f = @(x) -LLsbg(x,week,customers,lost);
```

```
% Generator1: Generator for n segments with fixed probabilities
function y = generator1(x,scale)
%
% changes any x(i), no feasibility tests
% use: opt.Generator = @(x) generator1(x,10)
%
y = x;
pos = unidrnd(length(x));       % random position of the change
z = y(pos) + randn/scale;       % new value
while z>0.999999 || z<0.000001
    z = y(pos) + randn/scale;  % repeat until in range
end
y(pos) = z;
```

```
% Generator3: Generator for n segments with optimizable probabilities
function y = generator3(x,scale)
```

```
%
% generator for n segments, both thetas and probs optimized
% length of x must be even
% use: opt.Generator = @(x) generator3(x,10)
%
n = length(x);
m = n/2;
y = x;
pos = unidrnd(n);              % random position of the change
z = y(pos) + randn/scale;      % new value
while z>=0.999999 || z<=0.000001
    z = y(pos) + randn/scale;  % repeat until in range
end
y(pos) = z;
if pos > m                     % restoring feasibility of probs
    c = 1/sum(y(m+1:n));
    y(m+1:n) = c*y(m+1:n);
end
```

```
% Generatorsbg: Generator for sBG model
function y = generatorsbg(x,scale)
%
% changes any x(i), no feasibility tests
% use: opt.Generator = @(x) generator1(x,10)
%
```

```
y = x;
pos = unidrnd(length(x));       % random position of the change
z = y(pos) + randn/scale;       % new value
while z<0.000001
    z = y(pos) + randn/scale;   % repeat until in range
end
y(pos) = z;
```

```
% Objective2: Objective function with optimizable probabilities
function y = objective2(x,periods,customers,lost)
%
% Objective function for both thetas and probabilities optimized
%   thetas: 1st half of x
%   probs : 2nd half of x (length of x must be even)
%
n = length(x);
m = n/2;
thetas = x(1:m);
probs = x(m+1:n);
y = -LLn(thetas,probs,periods,customers,lost);
```

```
% LLn: Log-likelihood function for n segments
function x = LLn(theta,prob,month,customers,lost)
```

```
%
%  Log-likelihood for m segments
%
n = length(month);
m = length(prob);
probTt(1) = 0;
for i = 2:n
    probTt(i) = 0;
    for j = 1:m
        probTt(i) = probTt(i)
                  + theta(j)*((1-theta(j))^(month(i)-1))*prob(j);
    end
end;
Probsum = sum(probTt);
probTt(1) = [];
S(1) = 1 - probTt(1);
for i = 2:n
    S(i) = S(i-1) - probTt(i-1);
end
LLterm(1) = 0;
for i = 2:n
    LLterm(i) = lost(i)*log(probTt(i-1));
end;
LLterm(1) = [];
if Probsum == 1
```

```
    LLTerm = [LLterm -Inf];
else
    LLTerm = [LLterm customers(n)*log(1-Probsum)];
end
x = sum(LLTerm);
```

```
% LLsbg: Log-likelihood function for sBG model
function x = LLsbg(alphabeta,week,customers,lost)
%
%  Log-likelihood for m segments
%
n = length(customers);
probTt(1) = 0;
probTt(2) = alphabeta(1)/(alphabeta(1) + alphabeta(2));
for i = 3:n
    probTt(i) = ((alphabeta(2) + (week(i) - 2))/(alphabeta(1) +
alphabeta(2) + (week(i) - 1)))*probTt(i-1);
end;
Probsum = sum(probTt);
probTt(1) = [];
LLterm(1) = 0;
for i = 2:n
    LLterm(i) = lost(i)*log(probTt(i-1));
end;
LLterm(1) = [];
```

```
LLTerm = [LLterm customers(n)*log(1-Probsum)];
x = sum(LLTerm);
```

```
% IGTT: Test 2 segments of subscribers with probabilities 0.4 & 0.6
data1;
globopt = 531;
best = Inf;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01   0.1*y + 0.01]
        [minimum,zz] = anneal(f,th0,opt)
        Z(x+1,y+1) = zz;
        if (zz < globopt)
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) = 0;
        end
        if zz<best
            best = zz;
            bestmin = minimum;
        end
    end
end
A
Z
```

```
globals = sum(sum(A))
best
bestmin
```

```
% Friends: Test 2 segments of subscribers with probabilities 0.4 & 0.6
data7;
globopt = 490;
best = Inf;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01   0.1*y + 0.01]
        [minimum,zz] = anneal(f,th0,opt)
        Z(x+1,y+1) = zz;
        if (zz < globopt)
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) = 0;
        end
        if zz<best
            best = zz;
            bestmin = minimum;
        end
    end
end
A
```

```
Z
globals = sum(sum(A))
best
bestmin
```

```
% IGTT: Test 2 segments of subscribers with probabilities 0.1 & 0.9
data2;
globopt = 529;
best = Inf;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01   0.1*y + 0.01]
        [minimum,zz] = anneal(f,th0,opt)
        Z(x+1,y+1) = zz;
        if (zz < globopt)
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) = 0;
        end
        if zz<best
            best = zz;
            bestmin = minimum;
        end
    end
end
```

```
A
Z
globals = sum(sum(A))
best
bestmin
```

```
% Friends: Test 2 segments of subscribers with probabilities 0.1 & 0.9
data8;
globopt = 489;
best = Inf;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01   0.1*y + 0.01]
        [minimum,zz] = anneal(f,th0,opt)
        Z(x+1,y+1) = zz;
        if (zz < globopt)
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) = 0;
        end
        if zz<best
            best = zz;
            bestmin = minimum;
        end
    end
```

```
end
A
Z
globals = sum(sum(A))
best
bestmin
```

```
% IGTT: Test 2 segments of subscribers with optimizable probabilities
data3;            % optimization - n segments, both optimized
n = 2;            % change this line to fix n
best1 = 529;
best = Inf;
worst = -Inf;
counter = 0;
A = 0;
while counter<101
    th0 = 0.00001 + rand(1,2*n)*0.9999;
    c = 1/sum(th0(n+1:2*n));
    th0(n+1:2*n) = c*th0(n+1:2*n);
    counter = counter + 1
    [minimum,z] = anneal(f,th0,opt)
    if z<best1
       A = A+1;
       best = z;
       bestmin = minimum;
```

```
    elseif z>worst
       worst = z;
       worstmin = minimum;
    end
end
best
bestmin
A
```

```
% Friends: Test 2 segments of subscribers with optimizable probabilities
data9;            % optimization - n segments, both optimized
n = 2;            % change this line to fix n
best1 = 489;
best = Inf;
worst = -Inf;
counter = 0;
A = 0;
while counter<101
    th0 = 0.00001 + rand(1,2*n)*0.9999;
    c = 1/sum(th0(n+1:2*n));
    th0(n+1:2*n) = c*th0(n+1:2*n);
    counter = counter + 1
    [minimum,z] = anneal(f,th0,opt)
    if z<best1
       A = A+1;
```

```
        best = z;
        bestmin = minimum;
    elseif z>worst
        worst = z;
        worstmin = minimum;
    end
end
best
bestmin
A
```

```
% IGTT: Test 3 segments of subscribers with probabilities 0.1 0.3 & 0.6
data4;          % optimization - n segments, probs fixed
n = 3;          % change this line to fix n
best = Inf;
best1 = 529;
worst = -Inf;
counter = 0;
A = 0;
while counter<1001
    th0 = 0.00001 + rand(1,n)*0.9999;
    counter = counter + 1
    [minimum,z] = anneal(f,th0,opt)
    if z<best1
        A = A + 1;
```

```
        best = z;
        bestmin = minimum;
    elseif z>worst
        worst = z;
        worstmin = minimum;
    end
end
best
bestmin
A
```

```
% Friends: Test 3 segments of subscribers with probabilities 0.1 0.3 & 0.6
data10;           % optimization - n segments, probs fixed
n = 3;            % change this line to fix n
best = Inf;
best1 = 489;
worst = -Inf;
counter = 0;
A = 0;
while counter<1001
    th0 = 0.00001 + rand(1,n)*0.9999;
    counter = counter + 1
    [minimum,z] = anneal(f,th0,opt)
    if z<best1
        A = A + 1;
```

```
        best = z;
        bestmin = minimum;
    elseif z>worst
        worst = z;
        worstmin = minimum;
    end
end
best
bestmin
A
```

```
% IGTT: Test 3 segments of subscribers with optimizable probabilities
data5;          % optimization - n segments, both optimized
n = 3;          % change this line to fix n
best1 = 529;
best = Inf;
worst = -Inf;
counter = 0;
A = 0;
while counter<1001
    th0 = 0.00001 + rand(1,2*n)*0.9999;
    c = 1/sum(th0(n+1:2*n));
    th0(n+1:2*n) = c*th0(n+1:2*n);
    counter = counter + 1
    [minimum,z] = anneal(f,th0,opt)
```

```
    if z<best1
       A = A+1;
       best = z;
       bestmin = minimum,
    elseif z>worst
       worst = z;
       worstmin = minimum;
    end
end
best
bestmin
A
```

```
% Friends: Test 3 segments of subscribers with optimizable probabilities
data11;            % optimization - n segments, both optimized
n = 3;             % change this line to fix n
best1 = 489;
best = Inf;
worst = -Inf;
counter = 0;
A = 0;
while counter<1001
    th0 = 0.00001 + rand(1,2*n)*0.9999;
    c = 1/sum(th0(n+1:2*n));
    th0(n+1:2*n) = c*th0(n+1:2*n);
```

```
    counter = counter + 1
    [minimum,z] = anneal(f,th0,opt)
    if z<best1
       A = A+1;
       best = z;
       bestmin = minimum;
    elseif z>worst
       worst = z;
       worstmin = minimum;
    end
end
best
bestmin
A
```

```
% IGTT: Test sBG model
data6;
globopt = 673;
best = Inf;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01 0.1*y + 0.01]
        [minimum,zz] = anneal(f,th0,opt)
        Z(x+1,y+1) = zz;
        if (zz < globopt)
```

```
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) = 0;
        end
        if zz<best
           best = zz;
           bestmin = minimum;
        end
    end
end
A
Z
globals = sum(sum(A))
best
bestmin
```

```
% Friends: Test sBG model
data12;
globopt = 622;
best = Inf;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01 0.1*y + 0.01]
        [minimum,zz] = anneal(f,th0,opt)
        Z(x+1,y+1) = zz;
```

```
        if (zz < globopt)
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) - 0;
        end
        if zz<best
           best = zz;
           bestmin = minimum;
        end
    end
end
A
Z
globals = sum(sum(A))
best
bestmin
```

```
% Test for constrained nonlinear optimization algorithm
globopt = 531;
A = 0;
for x = 0:9
    for y = 0:9
        th0 = [0.1*x + 0.01   0.1*y + 0.01]
        [minimum,zz] = fmincon(f,th0',[],[],[],[],[0.0001 0.0001]',
                                                    [0.9999 0.9999]')
```

```
        Z(x+1,y+1) = zz;
        if (zz < globopt)
            A(x+1,y+1) = 1;
        else
            A(x+1,y+1) = 0;
        end
    end
end
A
Z
globals = sum(sum(A))
```

```
%Simulated Annealing Algorithm
function [minimum,fval] = anneal(loss, parent, options)

def = struct(...
        'CoolSched',@(T) (.8*T),...
        'Generator',@(x) (x+(randperm(length(x))==length(x))*randn/100),...
        'InitTemp',1,...
        'MaxConsRej',1000,...
        'MaxSuccess',20,...
        'MaxTries',300,...
        'StopTemp',1e-8,...
        'StopVal',-Inf,...
        'Verbosity',1);
```

```
if ~nargin
    minimum = def;
    return
elseif nargin<2,
    error('MATLAB:anneal:noParent','You need to input a first guess.');
elseif nargin<3,
    options=def;
else
    if ~isstruct(options)
        error('MATLAB:anneal:badOptions',...
            'Input argument ''options'' is not a structure')
    end
    fs = {'CoolSched','Generator','InitTemp','MaxConsRej',...
        'MaxSuccess','MaxTries','StopTemp','StopVal','Verbosity'};
    for nm=1:length(fs)
        if ~isfield(options,fs{nm}), options.(fs{nm}) = def.(fs{nm}); end
    end
end

newsol = options.Generator;
Tinit = options.InitTemp;
minT = options.StopTemp;
cool = options.CoolSched;
minF = options.StopVal;
```

```
max_consec_rejections = options.MaxConsRej;
max_try = options.MaxTries;
max_success = options.MaxSuccess;
report = options.Verbosity;
k = 1;

itry = 0;
success = 0;
finished = 0;
consec = 0;
T = Tinit;
initenergy = loss(parent);
oldenergy = initenergy;
total = 0;
if report==2, fprintf(1,'\n  T = %7.5f, loss = %10.5f\n',T,oldenergy); end

while ~finished;
    itry = itry+1;
    current = parent;

    if itry >= max_try || success >= max_success;
        if T < minT || consec >= max_consec_rejections;
            finished = 1;
            total = total + itry;
            break;
```

```
    else
        T = cool(T);
        if report==2,
            fprintf(1,'  T - %7.5f, loss - %10.5f\n',T,oldenergy);
        end
        total = total + itry;
        itry = 1;
        success = 1;
    end
end

newparam = newsol(current);
newenergy = loss(newparam);

if (newenergy < minF),
    parent = newparam;
    oldenergy = newenergy;
    break
end

if (oldenergy-newenergy > 1e-6)
    parent = newparam;
    oldenergy = newenergy;
    success = success+1;
    consec = 0;
```

```
    else
        if (rand < exp( (oldenergy-newenergy)/(k*T) ));
            parent = newparam;
            oldenergy = newenergy;
            success = success+1;
        else
            consec = consec+1;
        end
    end
end

minimum = parent;
fval = oldenergy;

if report;
    fprintf(1, '\n  Initial temperature:     \t%g\n', Tinit);
    fprintf(1, '  Final temperature:       \t%g\n', T);
    fprintf(1, '  Consecutive rejections:  \t%i\n', consec);
    fprintf(1, '  Number of function calls:\t%i\n', total);
    fprintf(1, '  Total final loss:        \t%g\n', fval);
end
```

Bibliography

[1] Alberts, L.J.S.M. Bsc (2006) *Churn Prediction in the Mobile Telecommunications Industry*: Department of General Sciences, Maastricht University, Netherlands

[2] Alkhamis, T.M., Ahmed, M.A. (2004) *Simulation-Based Optimization using Simulated Annealing with Confidence Interval*: Department of Statistics and Operations Research, Kuwait University, Kuwait

[3] Avello, E.A., Baesler, F.F., Moraga R.J. (2004) *A Meta-Heuristic Based on Simulated Annealing for Solving Multiple-Objective Problems in Simulation Optimization*: Departamento de Ingenieria Industrial, Universidad del Bio-Bio, Chile

[4] Cardell, N.S., Golovnya, M., Steinberg, D. (2003) *Churn Modelling for Mobile Telecommunications*: Salford Systems

[5] Fader, P.S., Hardie, B.G.S. (2006) *How to Project Customer Retention*: University of Pennsylvania

[6] Fader, P.S., Hardie, B.G.S. (2007) *How Not to Project Customer Retention*: University of Pennsylvania

[7] Fader, P.S., Hardie, B.G.S. (2008) *Applied Probability Models in Marketing Research: Introduction*: 19th Annual Advanced Research Techniques Forum

[8] Figini, S., Giudici, P., Brooks, S.P. (2006) *Building Predictive Models for Feature Selection in Genomic Mining*: Dipartimento di economia politica e metodi quantitativi, Universitá degli studi di Pavia, Italia

[9] Galea, K. (2007) *Modelling Telecoms Customers' Spending Patterns & Churn*: Department of Statistics & Operations Research, Faculty of Science, University of Malta

[10] Gandy, A., Jensen, U., Lütkebohmer *Survival Analysis Applied to an Actuarial Problem*: Institute of Applied Mathematics and Statistics, University of Hohenheim, Germany

[11] Jahromi, A.T. (2009) *Predicting Customer Churn in Telecommunications Service Providers*: Tarbiat Modares University & Lulea University of Technology

[12] Jones, M.H., Preston, K.W.Jr. (2004) *Stochastic Approximation with Simulated Annealing as an Approach to Global Discrete-Event Simulation Optimization*: Department of Systems and Information Engineering, University of Virginia, U.S.A.

[13] Junxiang, L. *Predicting Customer Churn in the Telecommunications Industry*: Sprint Communications Company, Overland Park, Kansas

[14] Karakaya, K., Gomez, F., Abbott, B. (2006) *Curing Customer Churn*: Diamond, Chicago

[15] Manero, C (2008) *Churn Management - The Colour of Money*: IDATE, France

[16] Mokadikwa, T. (2008) *Factors Influencing Customer Churn Rate and Retention in the Mobile Market*: Department of Entrepreneurial Studies and Management, Durban University of Technology

[17] Neslin, S.A., Gupta, S., Kamakura, W., Junxiang, L., Mason, C.H. (2006) *Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models*: Journal of Marketing Research, American Marketing Association

[18] National Statistics Office (2010) *Post and Telecommunication: Q1 2009 News Release, Q2 2009 News Release, Q3 2009 News Release & Q4 2009 News Release*: National Statistics Office, Malta

[19] Pawlak, Z. (2002) *Rough Set Theory and its Applications*: Journal of Telecommunications and Information Technology

[20] Pintér, J.D. *Continuous Global Optimization: Models, Algorithms and Software*: Second Edition, Springer US

[21] Potts, W. *Survival Data Mining*: Data Miners Inc., Cambridge

[22] Pytlak, R., Stecz, W. *Tariff Optimization Problem - Formulation and Algorithms*: Faculty of Cybernetics, Military University of Technology, Poland

[23] Reinartz, W., Thomas, J.S., Kumar, V. (2005) *Balancing Acquisition and Retention Resources to Maximize Customer Profitability*: Journal of Marketing

[24] Roh, T.H., Han, I., Jang, W.H. *The Churn Management for Telecom Market using the Knowledge Discovery in Database*: Department of Information System, South Korea

[25] Rosset, S., Neumann, E. *Integrating Customer Value Considerations into Predictive Modelling*: Amdocs Ltd

[26] Vandekerckhove, J. (2006) *General Simulated Annealing Algorithm*

[27] Van Laarhoven, P.J.M., Aarts, E.H.L. (1987) *Simulated Annealing: Theory and Applications*: Kluwer Academic Publishers, The Netherlands

[28] Wah, B.W., Chen, Y., Wang, T. (2008) *Theory and Applications of Simulated Annealing for Nonlinear Constrained Optimization*: Cher Ming Tan (Ed.), InTech, Croatia