Generating snapshots and analyzing performance in wireless communication

Timmers, D.P.M.

Award date:
2007

Eindhoven University of Technology
Department of Mathematics and Computing Science

Generating Snapshots and Analyzing Performance in Wireless Communication

by

D.P.M. Timmers

January 9, 2007
Master's Thesis in Industrial and Applied Mathematics
40 credits

Supervisors:

Prof.dr.	R.W.v.d. Hofstad
Prof.dr.ir.	E.R. Fledderus

Abstract

In this paper we study two models of the UMTS wireless network, the single cell model and the quadruple cell model. For the single cell model we derive a complete probabilistic analysis. The quadruple cell model is too complicated for a successful probabilistic analysis. However, we present a novel way to greatly reduce the running time of the simulation program for this specific model.

Contents

List of Figures

List of Acronyms

3G	Third Generation
BEP	Bit Error Probability
CDMA	Code Division Multiple Access
DS-CDMA	Direct Sequence Code Division Multiple Access
LDA	Linear Discriminant Analysis
MAI	Multiple Access Interference
OCI	Other Cell Interference
PSD	Power Spectral Density
QDA	Quadratic Discriminant Analysis
SCI	Same Cell Interference
SNR	Signal to Noise Ratio

Chapter 1

Introduction

The latest revolution in wireless communication is the implementation of the third generation (3G) HSDPA wireless network. Although parts of The Netherlands are already covered by 3G, the construction of the 3G network is far from done. The terminals of users of the network connect to access points or base stations. With the continuing increase in data traffic and speed, new basestations must be added to the network to cope with the increasing demand, this is called densification. Hence, the providers of the 3G network are in a constant search to handle the increase in demand by adding a minimal number of basestations to the existing configuration of the network. One way to find the optimal expansion is is to test different configurations of the network for several settings of the parameters. One can think of the positions and heights of the antennas but also the vertical angles of the antennas, for more examples see [42, 45]. This calls for constant evaluation of the network and for methods which are able to forecast the performance of the 3G network for different settings of the parameters.

Two important measures for the performance of a wireless network are call blocking and outage/missed traffic. The terminals of the users have a maximal transmit power. If the user requires on average a transmit power greater than the maximal transmit power, then the call is blocked. The second measure of performance is missed traffic. Missed traffic is based on the observation that during transmission, bit errors are bound to occur. The network has some mechanisms to correct bit errors. However, error-correcting codes have their limitations and even if a user is not blocked, the terminal of the user can still, at moments in time, require a transmit power greater than the maximal transmit power. In this case a user can suffer from “missed” traffic. In this paper we will focus on estimation of missed traffic.

Models of the wireless network are generally too complex for an analytical study and thus we need to run simulations to asses the performance of the network under certain settings for the parameters. There are basically two approaches to simulate wireless networks, static simulation and dynamic simulation. Static simulation can be regarded as taking photographs or snapshots of the wireless network at a specific moment in time. Snapshots give information about the number, the positions and the requirements of the users. Based on this information it gives an estimation of the performance of the wireless network. Dynamical simulation builds upon static simulation in such a way that the simulation starts from an independent snapshot and proceeds for a predefined duration of time taking dynamic effects into consideration.

Dynamic simulation programs are very powerful tools to accurately assess the performance of a wireless network and to allocate problem regions of the network. However, dynamic simulation has one major downfall and that is running time. The programs are very complex and the biggest issue is the modeling of users. The program needs to model the time users are connected to the network, the speed at which users cross the network and along which paths users progress. The sheer amount of possibilities for these dynamic effects cause the program to have high variability so that it requires long running times before conclusions can be drawn about the performance of the network.

Because static simulation is not concerned with the issue of modeling dynamic effects, it suffers from less variability and is able to rapidly evaluate the performance of wireless networks for a large set of possible settings of the parameters. However, omitting the dynamic effects is also a weakness and static simulation is bound to be less accurate than dynamic simulation. Note that static and dynamic simulation complement each other. Static simulation has the ability to select a small subset of all possible settings for the parameters which show best performance, then dynamic simulation can be used to accurately measure the performance for this small set of settings.

Dynamic simulation has drawn much attention in the literature [42, 64, 65, 67]. Moreover, there are at least two dynamical simulation programs up and running. TNO has implemented a dynamical program under the name COUGAR while Radioplan implemented the dynamical simulation program WiNeS. Despite the obvious benefits of combining static and dynamic simulation, little attention has been paid to static simulation programs. Besides a brief study by Laiho, Wacker and Novosad [42] no studies have, to our knowledge, addressed the problem of static simulation, let alone methods to further improve the performance of static simulation.

The ultimate goal is to model the 3G HSDPA network. However, due to complexity of the HSDPA network, we will start by studying the UMTS network. This paper has two main goals. First, to introduce the concept of static simulation for the UMTS network. We analyze the behavior of two specific models, the single cell model and the quadruple cell model. Although not as complicated as dynamic simulation programs, static simulation can also become involved. Therefore, the second aim of this paper is to reduce the running time of static simulation programs. This is done by classifying snapshots into groups in such a way that it enables us to generate snapshots from these groups. For example if we have a group of snapshots which show poor performance we adapt the parameters of the model, generate snapshots from this specific group and observe whether the performance improved. It goes without saying that this reduces running time.

The outline of this paper is as follows. The BEP plays an important role in the computation of missed traffic. In Chapter 2 we shall study the communication between user and basestation to derive an approximation of the BEP. Chapter 3 lays out some general rules to construct a static model of the wireless network. Chapters 4 and 5 are concerned with the analysis of two specific static models, the single cell model and the quadruple cell model. In Chapter 6 we introduce classification rules and implement these rules for the quadruple cell model. Chapter 7 deals with the problem how to generate snapshots from the groups created by the classification rule. The method to generate snapshots from prescribed groups crucially depends on the fact that users are homogeneously distributed over the network. In Chapter 8 we introduce a technique that allows us to extend the method to snapshots with an inhomogeneous distribution of the users.

Chapter 2

Code division multiple access

A wireless network consists of a number of acces points to which users, or actually their terminals, connect. A single access point should be able to communicate with several users at the same time by using a set of rules also called a protocol. The protocols which achieve multiple access are called multiplexing protocols. The protocol used in the UMTS network is a code division multiple access (CDMA) protocol. In this chapter we shall give a short introduction into this class of protocols and discuss the principles of CDMA. In Section 2.1 we explain how CDMA protocols achieve the multiple access property. In Section 2.2 we use our knowledge about CMDA to derive an approximation for the BEP.

2.1 CDMA protocol

CDMA protocols constitute a class of protocols which achieve the multiple access property by means of coding. The protocol of main interest is direct sequence CDMA, DS-CDMA. For any CDMA protocol, the bandwidth of the information-bearing signal will be increased before it is transmitted. Hence, the process spreads the bandwidth of the signal and, therefore, CDMA protocols are also known as spread spectrum multiple access protocols.

In DS systems, a narrowband signal containing a message with bandwidth B_1, is directly multiplied by a code signal with a much larger bandwidth B_2. Essential is that the code signal and the message signal are independent of one another. Thus, the transmitted signal will have a bandwidth that is equal to the bandwidth of the code signal, the signal is spreaded. When the receiver gets the transmitted signal it will synchronize and decode the signal which causes the signal to be despreaded. A sketch of the process of spreading and despreading is shown in Figure 2.1. The signal is measured by the standard measure for the power of a wave, the power spectral density PSD, which is defined to be the power per unit of frequency.

The information bearing signal consists of bits b_j while the code signal consists of symbols a_j. Usually, these symbols a_j are called chips. The rate of chips in the code signal R_c is much greater than the rate of the bits in the information signal R_b. The process gain, or spreading factor, G is defined to be the ratio of the chip rate to the bit rate

$$G = \frac{R_c}{R_b} \ .$$

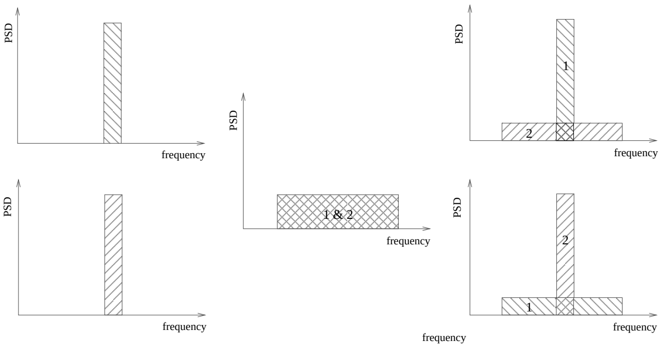


Figure 2.1: Principle of CDMA

2.1.1 Coding and decoding

To illustrate the operation of DS-CDMA, consider the information signal of the m^{th} user $b_m(t)$ which is assumed to be a sequence of binary ones (+1) and zeros (-1). The message $b_m(t)$ is a sequence of symbols b_{mk}, each of duration T_b. The data signal is given by

$$b_m(t) = \sum_{k=-\infty}^{\infty} b_{mk} \Psi\left(\frac{t - kT_b}{T_b}\right),$$

where Ψ is the unit pulse function

$$\Psi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

The signal $b_m(t)$ is multiplied by a coding sequence $a_m(t)$ which is a sequence of chips

$$a_m(t) = \sum_{k=-\infty}^{\infty} \sum_{j=0}^{G-1} a_{mk} \Psi\left(\frac{t - (kG + j)T_c}{T_c}\right),$$

where T_c is the chip period.

We assume that the multiplied signal $a_m(t)b_m(t)$ is then modulated by means of a binary phase-shift keying and upconverted to a carrier frequency f_{ca}. Hence, the transmitted signal becomes

$$s_m(t) = \sqrt{2P}\ a_m(t)b_m(t)\cos(2\pi f_{ca}t),$$

where P is the power. For technical reasons we ignore oscillator phase noise. For a complete schematic of a DS-CDMA transmitter see Figure 2.2(a). Before modulating the sequence $a_m(t)b_m(t)$, the signal passes a baseband band-pass filter which is a device that passes frequencies within a certain range and rejects, attenuates, frequencies outside this range. The filter is necessary because the multiplication of the signal $a_m(t)b_m(t)$ yields, besides the desired signal, also higher harmonics which have to be filtered out.

The receiver filters out the transmitted signals of different users using a wideband intermediate frequency filter. A part of the power of the signal is used to synchronize while the other part is used to decode the desired signal. The synchronizing is a continuous process hence the loop in Figure 2.2(b). For now, we will concentrate on how the signal is decoded.

Suppose that there are N users which transmit data, the data sent by each user is

$$s_m(t) = \sqrt{2P_m}\, a_m(t) b_m(t) \cos(2\pi f_{ca} t), \qquad m = 0, 1, \ldots, N-1,$$

For technical reasons we assume that the signals of the users are synchronized, that is all users transmit at the same time grid. We refer to Klok [39] for a detailed derivation for asynchronous systems. The receiver signal $r(t)$ consists of the transmitted signals of the N users

$$r(t) = \sum_{k=0}^{N-1} s_k(t) + n(t), \tag{2.1}$$

where $n(t)$ is a white noise process, i.e. the derivative of Brownian motion in distributional sense. The white noise represents the noisy channels of the users and all interference of other sources that is not yet taken into account and is called *additive white Gaussian noise* (AWGN).

To retrieve the data bit b_{m1}, the signal $r(t)$ is multiplied by $a_m(t)\cos\omega_c t$ and then averaged over $[0, T_b]$

$$Z_{m1} = \frac{1}{T_b} \int_0^{T_b} r(t) a(t) \cos(2\pi f_{ca} t) dt. \tag{2.2}$$

The receiver decides whether the bit b_{m1} was -1 or +1 by evaluating the sign of Z_{m1}. If Z_{m1} would be zero, the bit is set to -1 or 1 with equal probability. By substituting (2.1) into (2.2) we have

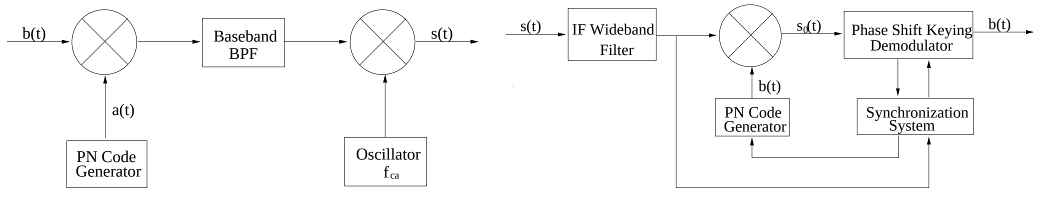
$$Z_{m1} = I_{m1} + \sum_{\substack{k=1 \\ k \neq m}}^{N-1} I_{k1} + \eta = I_{m1} + \zeta + \eta, \tag{2.3}$$

where

$$I_{k1} = \frac{\sqrt{2P_k}}{T_b} \int_0^{T_b} b_k(t) a_k(t) a_m(t) \cos^2(2\pi f_{ca} t) dt,$$

for $0 \leq k \leq N-1$ and

$$\eta = \frac{1}{T_b} \int_0^{T_b} a_m(t) \cos(2\pi f_{ca} t) dB(t).$$



(a) Transmitter (b) Receiver

Figure 2.2: A direct sequence spread spectrum transmitter and receiver

The contribution of user m is I_{m1} while the other users contribute ζ to the interference of user m. The thermal noise contribution is given by η. In Appendix A.1 we derive that

$$Z_{m1} = \sqrt{\frac{P_m}{2}}\, b_{m1} + \frac{1}{G} \sum_{\substack{k=0 \\ k\neq m}}^{N-1} b_{k1} \sum_{j=1}^{G} a_{kj} a_{mj} \sqrt{\frac{P_k}{2}} + \frac{1}{G} \sum_{j=1}^{G} a_{mj} \frac{N_j}{\sqrt{2T_c}},$$

where $(N_j)_{j=1}^G$ are independent and identically Gaussian distributed random variables with zero mean and variance 1.

Ideally, the vectors $(a_{m1}, \ldots, a_{mG})$ and $(a_{k1}, \ldots, a_{kG})$, $k \neq m$, would be orthogonal so that $\sum_{j=1}^G a_{kj} a_{mj} = 0$. However, it is more efficient to allow non-orthogonal codes. In practice, the sequences are generated by a random number generator, so that the sequences resemble random codes. To model these random sequences, let A_{kj}, $0 \leq k \leq N-1$, $1 \leq j \leq G$, be an array of i.i.d. random variables with distribution

$$\mathbb{P}(A_{kj} = +1) = \mathbb{P}(A_{kj} = -1) = 1/2.$$

Then the decision statistic Z_{m1} becomes

$$Z_{m1} = \sqrt{\frac{P_m}{2}}\, b_{m1} + \frac{1}{G} \sum_{\substack{k=0 \\ k\neq m}}^{N-1} b_{k1} \sum_{j=1}^{G} A_{kj} A_{mj} \sqrt{\frac{P_k}{2}} + \frac{1}{G} \sum_{j=1}^{G} A_{mj} \frac{N_j}{\sqrt{2T_c}}. \tag{2.4}$$

2.2 Bit error probability

An important measure for the performance for any multiple access control is the bit error probability (BEP). In case of CDMA, a bit error occurs when the signs of the statistic Z_{m1} and the bit b_{m1} are opposite. Because of symmetry we can assume without loss of generality that $b_{m1} = +1$. In this case, a bit error occurs if $Z_{m1} < 0$. To compute the BEP we evaluate (2.4).

Note that in the definition of Z_{m1} we find two sums in which the random variables A_{kj} play an important role. For ζ we see that, by independence, the expectation for the products $A_{kj}A_{mj}$ are all zero. The variance of the products is

$$\mathrm{Var}(A_{kj}A_{mj}) = \mathbb{E}[A_{kj}^2 A_{mj}^2] - \mathbb{E}[A_{kj}A_{mj}]^2 = 1. \tag{2.5}$$

For the noise contribution it follows that the expectation of $A_{mj}N_j$ is zero because of the independence of the two variables. The computation of the variance requires some more work. Due to a well-known result on conditional expectations, see among others [6], it holds that

$$\mathrm{Var}(A_{mj}N_j) = \mathbb{E}[\mathrm{Var}(A_{mj}N_j|A_{mj})] + \mathrm{Var}(\mathbb{E}[A_{mj}N_j|A_{mj}]) = \mathbb{E}[A_{mj}^2] = 1. \tag{2.6}$$

Having computed the expectation and variances we can now estimate the bit error probability. By the central limit theorem it follows that the two sums in the definition of Z_{m1} converge in distribution to a Gaussian distributed random variables. Using (2.5) and (2.6), it follows that Z_{m1} converges in distribution to a Gaussian random variable with mean

$$\mu_m = \sqrt{\frac{P_m}{2}},$$

and variance

$$\sigma_m^2 = \frac{1}{2G} \sum_{\substack{k=0 \\ k \neq m}}^{N-1} P_k + \frac{1}{2GT_c},$$

here we used that multiplying A_{kj} by b_{kj} does not alter the distribution of A_{kj}.

Having obtained the asymptotic distribution of Z_{m1}, we can compute the BEP.

$$\begin{aligned} \mathrm{BEP} = \mathbb{P}(Z_{m1} < 0) &= \frac{1}{\sqrt{2\pi}\sigma_m} \int_{-\infty}^{0} \exp\left(\frac{-(x-\mu_m)^2}{2\sigma_m^2}\right) dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{\infty}^{-\mu_m/\sigma_m} e^{-x^2/2} dx \end{aligned} \tag{2.7}$$

Hence, the quantity μ_m/σ_m can be used to approximate the BEP. Note that the BEP is bounded by 1/2, this is a natural bound. If the BEP would be larger than 1/2, then one would simply exchange the signs of the bits to obtain a BEP below 1/2.

The Gaussian approximation of the BEP is such a common procedure in the electrical engineering community that $(\mu_m/\sigma_m)^2$ has been given its own name: signal to noise ratio (SNR). This is because

$$\mathrm{SNR} = \frac{GP_m}{\sum_{\substack{k=0 \\ k \neq m}}^{N-1} P_k + 1/T_c},$$

where P_m is the power of the signal. To cause confusion, the variance of the thermal noise $1/T_c$ is simply referred to as the noise power, it is usually denoted with N_f. The term $\sum P_k$ in the denominator is called the multiple access interference (MAI).

In practice the (A_{mk}) sequences are not independent. The sequences used by CDMA are pseudo-random sequences and are also known as Pseudo Noise. However, a central limit theorem can still be used to validate the Gaussian distribution of the test statistic Z_{m1}. Therefore, the general approach in the electrical engineering community is to use the SNR to approximate the BEP where

$$\mathrm{SNR} = \frac{GP_m}{\sum_{\substack{k=0 \\ k \neq m}}^{N-1} P_k + N_f}. \tag{2.8}$$

For more information on the pseudo-random sequences we refer to [45, 53]. Viterbi [63] describes how the SNR can used to approximate the BEP in case of Pseudo Noise.

Chapter 3

Modeling a wireless network

As mentioned in the introduction of this paper, an important measure for the performance of a wireless network is missed traffic. Models of the wireless network should give an estimation of missed traffic by generating independent samples of possible traffic constellations. A traffic constellation is a complete description of the positions and number of users in the wireless network and is usually called a *snapshot*. Each snapshot, in combination with the user requirements, yields a certain level of missed traffic. Thus the model would need to give a description of the snapshots and provide an estimator of missed traffic based on the snapshots.

In this paper we will only treat models for single traffic networks. These are networks which provide a single service. Therefore, the model does not properly describe a UMTS network which supports multiple services such as e-mail, video-telephony, telephony and streaming. However, the model can be extended to a multiple service network.

The outline of this chapter is as follows. In Section 3.1 we give a description of the snapshots and show how users are placed into the cells of the wireless network. In Section 3.2 we provide a way to compute the transmit powers of the users, these powers are used in Section 3.3 to estimate missed traffic. In the final section of this chapter we shall comment on the implementation of the model in a simulation tool. The modeling decisions we make are for a great part similar to those of Laiho, Wacker and Novosad [42].

3.1 Snapshots

The area covered by the wireless network is divided into cells. Ideally, the division results in a grid of hexagonally shaped cells which are marked out by their corresponding basestations, see Figure 3.1. The distance of the basestation to the corners of its corresponding cell is R, while D_b is the distance between each pair of neighboring basestations.

To construct a snapshot, it is necessary to place a random number of users at random positions in the cells. Or to put it differently, it is necessary to model the positions and number of the users by a random process. The choice is made to use a spatial Poisson process which realizes countable subsets Π in $\mathbb{R}^2$. If the reader is not familiar with the concept of spatial Poisson processes we suggest to read Appendix B.1 before proceeding.

For each cell, users are modeled by a homogeneous spatial Poisson process Π with constant intensity λ on a circle C with radius R. Note that, although it is assumed that the cells in the network are hexagonally shaped, users are generated as if the cells were circles with radius R. It could occur that users are physically placed in a hexagonal cell of one basestation but are communicating with a neighboring basestation of that particular cell. This kind of behavior also occurs in real-life. The ongoing calls of users must be transferred to another basestation if the user moves out of the cell. The link with the old basestation is not broken before a link with the new basestation is established. Thus it happens that a user is connected to the old basestation when the user already crossed the border between the new and the old cell.

Returning to the snapshot we need to determine the number of users and the positions of the users in the circle C. The number of users N is Poissonian distributed with mean $\lambda|C|$, see (B.1). Let D represent the distance a user has to the basestation. The cumulative distribution function, hereafter abbreviated by cdf, of D is

$$F_D(d) = \mathbb{P}(D \leq d) = \frac{d^2}{R^2}. \tag{3.1}$$

The expression for the cdf was obtained by use of Theorem B.2. Studying the inverse of F_D it follows that

$$D = R\sqrt{U} \quad \text{where} \quad U \sim U(0,1). \tag{3.2}$$

Hence, the position for each user can be obtained in polar coordinates. For each user the angle θ_i is uniformly distributed on the interval $(0, 2\pi)$ and the distance to the basestation is described by either (3.1) or (3.2).

3.2 Determination of transmit powers

After a snapshot is generated and users are appointed to their respective basestations, we continue by computing the transmit powers for the users for the snapshot. The transmit power P_i^{tx} for user i depends on its position towards the basestation and the number and position of the other users in the network. The quantity which determines the transmit power for the users is the SNR.

Recall that the SNR can be used to approximate the BEP, see (2.7) and (2.8). Hence, the BEP can be upper bounded by lower bounding the SNR. The upper bound for the BEP is predetermined such that a certain desired level of service is guaranteed. Let μ

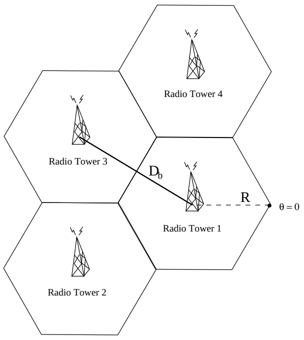


Figure 3.1: Wireless network

represent the lower bound on SNR, the bound μ is used to derive the transmit powers for the users in the snapshot.

As explained in Chapter 2, the denominator of the SNR consists of two parts, the MAI and the noise power N_f. The noise power is some given constant that is identical for each snapshot. However, the MAI needs to be computed for each snapshot separately. Therefore, we turn our attention to the computation of MAI.

Let K be the number of cells and n_j the number of users in cell j. Furthermore, P^{rec}_{kji} is the received power at basestation i from the j^{th} user which is connected to the k^{th} basestation. Consider the j^{th} user which is connected to basestation k. The MAI can be split up in two parts, same cell interference (SCI) and other cell interference (OCI). The MAI for this particular user then becomes

$$\mathrm{MAI}_{kj} = \sum_{\substack{i=1\\ i\neq j}}^{n_k} P^{rec}_{kik} + \sum_{\substack{\ell=1\\ \ell\neq k}}^{K} \sum_{m=1}^{n_\ell} P^{rec}_{\ell mk} = \mathrm{SCI}_{kj} + \mathrm{OCI}_k. \tag{3.3}$$

We have indexed OCI by the single index k since, by definition, the OCI will be the same for all users connected to basestation k. Substituting (3.3) into (2.8) we find a linear equation for all users in the network

$$\mu = \frac{G \cdot P^{rec}_{kjk}}{\mathrm{SCI}_{kj} + \mathrm{OCI}_k + N_f} \ ,$$

for $k = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, n_k$. We have put an equality sign instead of an inequality sign because the optimal solution has the transmit powers which are as small as possible.

The latter set of linear equations can be simplified. Recall that the model represents a UMTS network. Basestations of UMTS networks will update the powers of the users approximately 1500 times per minute. Hence, we can assume that the received powers P^{rec}_{kik} for $i = 1, 2, \ldots, n_k$ are identical. This behavior is also incorporated by the set of linear equations.

Claim 3.1. *The received powers at basestation k for all users connected to basestation k are the same, or*

$$P^{rec}_{kjk} = P^{rec}_{k\ell k}$$

for all $k = 1, \ldots, K$ and all $\ell, j \in \{1, 2, \ldots, n_k\}$.

Proof. Take some k and $j \neq \ell$, then the following two equations must be satisfied

$$\mu = \frac{G \cdot P^{rec}_{kjk}}{\mathrm{SCI}_{kj} + \mathrm{OCI}_k + N_f} \ ,$$
$$\mu = \frac{G \cdot P^{rec}_{k\ell k}}{\mathrm{SCI}_{k\ell} + \mathrm{OCI}_k + N_f} \ .$$

The system of equations can be reformulated to

$$G \cdot P^{rec}_{kjk} = \mu(\mathrm{SCI}_{kj} + P^{rec}_{kjk} - P^{rec}_{kjk} + \mathrm{OCI}_k + N_f) \ ,$$
$$G \cdot P^{rec}_{k\ell k} = \mu(\mathrm{SCI}_{k\ell} + P^{rec}_{k\ell k} - P^{rec}_{k\ell k} + \mathrm{OCI}_k + N_f) \ .$$

By the definition of same cell interference

$$\mathrm{SCI}_{kj} + P^{rec}_{kjk} = \mathrm{SCI}_{k\ell} + P^{rec}_{k\ell k},$$

thus subtracting the two equations yields

$$G(P^{rec}_{kjk} - P^{rec}_{k\ell k}) = -\mu(P^{rec}_{kjk} - P^{rec}_{k\ell k}),$$

which forces P^{rec}_{kjk} to equal $P^{rec}_{k\ell k}$. □

By Claim 3.1, the received powers can be determined by solving the set of equations

$$\mu = \frac{G \cdot P^{rec}_{kjk}}{(n_k - 1)P^{rec}_{kjk} + \sum_{\substack{\ell=1 \\ \ell \neq k}} \sum_{m=1}^{n_\ell} P^{rec}_{\ell mk} + N_f}\ , \tag{3.4}$$

for $k = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, n_k$. Naturally, the received power P^{rec}_{kji} at basestation i is a function of the transmit power P^{tx}_{kj} of the j^{th} user connected to basestation k. The function accounts for the fact that the transmit powers attenuate in transit from users to basestation. This effect is called path loss.

3.2.1 Path loss

The power of user j in cell k declines due to the fact that it has to be transmitted over a certain type of terrain for a certain distance. Models describing the total path loss generally have an empirical and a deterministic part [55]. The deterministic part of path loss solely depends on the distance between user and basestation and is called *propagation loss*. The empirical part of the total path loss models clutter along the propagation path and is called *shadowing* or *slow fading*.

The propagation loss is discounted for by the factor

$$\gamma_i = \frac{P_{received}}{P_{transmitted}}. \tag{3.5}$$

By the generalized Hata's equation [33], γ_i (dB) is

$$\gamma_i = K_1 + K_2 \log_{10}(f_{ca}) - K_3 \log_{10}(h_b) + [K_4 - K_5 \log_{10}(h_b)] \log_{10}(\max\{d_0, d_i\}),$$

where the K_i's are constants, f_{ca} is the carrier frequency in MHz, h_b is the base station antenna height in meters, d_i is the distance of user i to the base station in km and d_0 is the cut-off distance in km. It is assumed that the propagation loss is constant when the distance to the basestation is smaller than d_0. For a fixed base station antenna height, the propagation loss reduces to

$$\gamma_i = u_f + u_h + v_h \log_{10}(\max\{d_0, d_i\}), \tag{3.6}$$

where

$$\begin{aligned} u_f &= K_1 + K_2 \log_{10}(f_{ca}) \\ u_h &= -K_3 \log_{10}(h_b) \\ v_h &= K_4 - K_5 \log_{10}(h_b). \end{aligned}$$

In [55] it is mentioned that Hata's model for propagation loss is only valid for carrier frequencies below 1500 MHz. The UMTS network uses a carrier frequency of approximately 2000 MHz which is definitely too high. Although scientists are aware of this fact, the model is still popular and widely used to model propagation loss for UMTS networks. Hata alike propagation loss models are so popular because they give good approximations of propagation loss while computations are relatively easy.

Currently, users at equal distance to the basestation will require similar transmission powers. However, clutter along the propagation paths causes the path loss to differ for each path. The deviation from the nominal value is known as slow fading or shadowing and can be modeled as a lognormally distributed random variable [55]. The slow fading is usually defined as $10^{S/10}$ where S is normally distributed with zero mean and variance σ_S^2, this is denoted by $S \sim N(0, \sigma_S)$.

If users are not far apart from each other, the random variables which model slow fading correlate. For simplicity we ignore this effect and assume that the slow fading coefficients S_{kji} are mutually independent.

3.3 Removal of blocked users

As mentioned in the introduction, the main goal of the model is to find an estimator for missed traffic. This requires removing the users which on average require a transmit power greater than the maximal transmit power from the network. In this section we shall explain how blocked users are detected.

Dynamic simulation programs such as WINES or COUGAR have no problem in detecting blocked users. Users are one by one added to the network and based on the state of the network it can be easily assessed whether the new user is admitted to the network and. if it is admitted, whether other users will be blocked by the network. However, for a static simulation program the distinction is not so clear. A snapshot is nothing more than the state of the network at a specific moment in time. The snapshots shows all users, we can not see the order in which users arrived at the network or which users should have been blocked.

To determine whether a user needs to be removed and, if so, which one has to be removed, we first leave out the slow fading coefficients. The distribution of the slow fading factors is symmetric in the sense that $f(x) = f(x^{-1})$ for all $x > 0$. Hence, if we ignore the slow fading factors we get the "average condition" of the network for a particular snapshot. For the moment assume that we can determine the transmit powers for all users. If all users require, under these average conditions, a transmit power below the maximal allowable transmitting power $P_{\max}$ all users are accepted. Otherwise we remove the user which requires the largest transmit power from the network. This changes the conditions of the network and basically yields a new snapshot. We determine the transmit powers for this new snapshot and again consider the user with the largest transmit power. Users are iteratively removed from the network until all users require, under average conditions, a transmit power below $P_{\max}$.

The question remains how to compute the transmit powers of the users. Suppose we are given a snapshot with K basestations and to each basestation n_k, $1 \leq k \leq K$, users are connected. We are also provided with a lower bound for the BEP so that the lower bound μ can be computed. By (3.4), the received power of the j^{th} user connected to basestation k must satisfy

$$\mu = \frac{G P_{kjk}^{rec}}{(n_k - 1) P_{kjk}^{rec} + \mathrm{OCI}_k + N_f},$$

where the other cell interference satisfies

$$\mathrm{OCI}_k = \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} \sum_{i=1}^{n_k} P_{\ell i k}^{rec}.$$

Since we ignore slow fading, the transmit powers only suffer from propagation loss. Let $\gamma_{\ell i k}$ be Hata's propagation loss factor on the path between the position of the i^{th} user connected to basestation ℓ to basestation k. Then by Hata's equation it follows that

$$P_{\ell i k}^{rec} = P_{\ell i}^{tx} 10^{\gamma_{\ell i k}/10},$$

where $P_{\ell i}^{tx}$ is the transmit power of the i^{th} user connected to basestation ℓ. Then the equation for the SNR becomes

$$\mu = \frac{G P_{kj}^{tx} 10^{\gamma_{kjk}/10}}{(n_k - 1) P_{kj}^{tx} 10^{\gamma_{kjk}/10} + \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} \sum_{i=1}^{n_k} P_{\ell i}^{tx} 10^{\gamma_{\ell i k}/10} + N_f}, \tag{3.7}$$

for $k = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, n_k$. This leaves us with $n_1 + \ldots + n_K$ equalities and as many unknowns. Hence, the system of linear equations (3.7) can be solved, which gives us the transmit powers of the users for the average condition of the network.

To summarize, we start by randomly generating a snapshot by use of a homogeneous spatial Poisson process. This provides us with a set of users placed in K cells. Based on the snapshot we construct the system of linear equalities (3.7). If this system has a feasible solution, that is all transmit powers are positive, then the transmit powers are computed. If the maximal transmit power among all users exceeds $P_{\max}$, then the user which requires maximal transmit power is tagged as blocked and removed from the network. After the user is removed, we again construct a system of linear equations based on (3.7) and again the maximal transmit power among all users is determined. We repeat our previous steps until all the remaining users in the network require a transmit power below $P_{\max}$.

3.3.1 Two different estimators

After we have removed the blocked users we can go on to estimate missed traffic. Let $\mathcal{N}_U$ represent the set of users which remain in the network and are not blocked

$$\mathcal{N}_U = \{\{k, j\} \;:\; j^{th} \text{ user connected to basestation } k \text{ is not blocked}\}$$

We can proceed in two ways. The first is the most obvious, for each user in the network we draw K samples from the normal distribution. These K samples represent the slow fading factors on each of the K possible paths from the user to one of the K basestations. Let $S_{kj\ell}$ represent the slow fading factor on the path between the j^{th} user which is connected to basestation k and basestation ℓ. Then we can construct the following set of linear equations

$$\mu = \frac{G P_{kjk}^{tx} 10^{(\gamma_{kj} + S_{kjk})/10}}{(n_k - 1) P_{kj}^{tx} 10^{(\gamma_{kjk} + S_{kjk})/10} + \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} \sum_{i=1}^{n_k} P_{\ell i}^{tx} 10^{(\gamma_{\ell i k} + S_{\ell i k})/10} + N_f}, \tag{3.8}$$

for $k = 1, 2, \dots, K$ and $j = 1, 2, \dots, n_k$. Solving the system of linear equations yields the transmit powers of the users in the network. We also know that the transmit powers are bounded $P_{\max}$. Hence, the BEP for users which require a larger transmit power than $P_{\max}$ is larger than the allowable BEP. These users suffer from missed traffic, thus missed traffic is computed by

$$MT_1 = \sum_{\{k,j\} \in \mathcal{N}_U} \frac{\mathbb{1}\{P_{kj}^{tx} > P_{\max}\}}{|\mathcal{N}_U|}, \tag{3.9}$$

where the sum runs over all users which were not blocked.

However, there is a second possibility. Suppose that slow fading between users and the basestation to which they are *not* connected is not present. In this case the slow fading factor for the j^{th} user connected to basestation k is S_{kj}. Then the system of linear equations which has to be solved is

$$\mu = \frac{G P_{kj}^{tx} 10^{(\gamma_{kjk}+S_{kj})/10}}{(n_k - 1) P_{kj}^{tx} 10^{(\gamma_{kjk}+S_{kj})/10} + \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} \sum_{i=1}^{n_k} P_{\ell i}^{tx} 10^{\gamma_{\ell i k}/10} + N_f}, \tag{3.10}$$

for $k = 1, 2, \dots, K$ and $j = 1, 2, \dots, n_k$. Note that the solution to this system of equations is simply

$$P_{kj}^{tx} = 10^{-S_{kj}/10} \widetilde{P}_{kj}^{tx},$$

where $(\widetilde{P}_{kj}^{tx})_{k,j}$ is the solution to (3.7), it saves us solving an extra system of linear equations. However, the real gain of using (3.10) is that we can compute the probability $\mathbb{P}(10^{S_{kj}} \widetilde{P}_{kj}^{tx} > P_{\max})$ explicitly. This allows us to simply estimate missed traffic by

$$MT_2 = \sum_{\{k,j\} \in \mathcal{N}_U} \frac{\mathbb{P}(10^{S_{kj}/10} \widetilde{P}_{kj}^{tx} > P_{\max} | \widetilde{P}_{kj}^{tx})}{|\mathcal{N}_U|}, \tag{3.11}$$

where S_{kj} is a normally distributed variable with zero mean and variance σ_S^2 and the sum runs over all remaining users.

The random variable MT_2 is an unbiased estimator of missed traffic (with the assumption that there is only slow fading between users and the basestations to which they are connected). The expectation of MT_2 is given by

$$\mathbb{E}[MT_2] = \mathbb{E}\left[\sum_{\{k,j\} \in \mathcal{N}_U} \frac{\mathbb{E}[\mathbb{1}\{(10^{S_{kj}/10} \widetilde{P}_{kj}^{tx} > P_{\max}\} | \widetilde{P}_{kj}^{tx}]}{|\mathcal{N}_U|} \right]. \tag{3.12}$$

Using the property of expectation that $\mathbb{E}[\mathbb{E}[X|Y]] = \mathbb{E}[X]$ and rewriting the summation with help of the indicator function yields

$$\mathbb{E}[MT_2] = \mathbb{E}[\mathbb{E} \left[\sum_{\{k,j\}} \frac{\mathbb{E}[\mathbb{1}\{(10^{S_{kj}/10} \widetilde{P}_{kj}^{tx} > P_{\max}\}]}{|\mathcal{N}_U|} \cdot \mathbb{1}\{\{k,j\} \in \mathcal{N}_U\} \mid \mathcal{N}_U \right]].$$

The indicator function and $|\mathcal{N}_U|$ are both measurable with respect to $\mathcal{N}_U$. Hence, it follows that the sum and the indicator can be put in front of the conditional expectation.

$$\begin{aligned}\mathbb{E}[MT_2] &= \mathbb{E}\left[\sum_{\{k,j\}\in\mathcal{N}_U} |\mathcal{N}_U|^{-1}\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\{(10^{S_{kj}/10}\widetilde{P}_{kj}^{tx} > P_{\max}\} \mid \widetilde{P}_{kj}^{tx}\right] \mid \mathcal{N}_U\right]\right] \\ &= \mathbb{E}\left[\sum_{\{k,j\}\in\mathcal{N}_U} \frac{\mathbb{1}\{(10^{S_{kj}/10}\widetilde{P}_{kj}^{tx} > P_{\max}\}}{|\mathcal{N}_U|}\right]\end{aligned}$$

Question is why one would choose to use (3.11) instead of (3.9), the latter seems to be more accurate since it models slow fading on all possible paths between users and basestations. Besides that (3.11) looks like it is easier to work with, there is another argument. As is discussed in the next section, the two models are too complicated for an analytical study. Therefore, we have to use simulation to find estimators for missed traffic. It is likely that the variance of MT_1 is larger than the variance of MT_2 because we removed a part of the randomness from the model. The reduction of variance influences computation time directly since the variance assures a certain level of accuracy for the estimator of missed traffic. Hence, use of MT_2 will allow for a shorter running times.

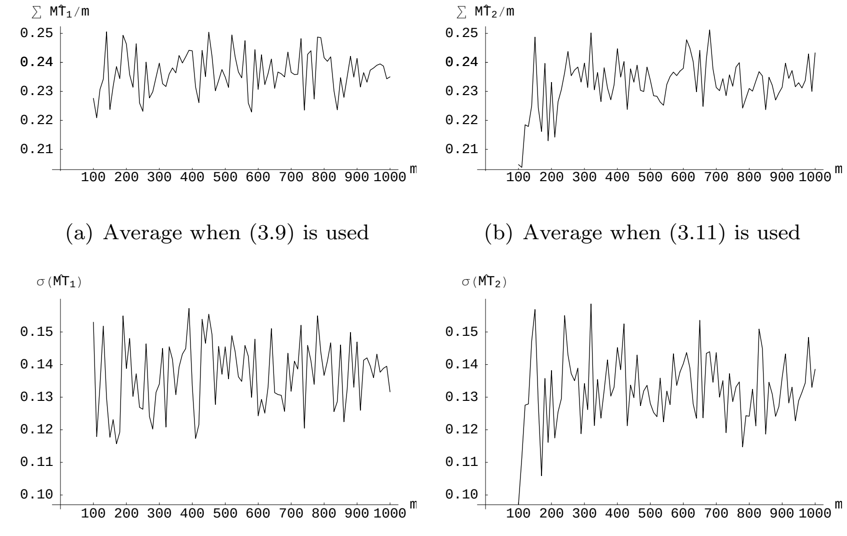
To illustrate the differences between the two models, we run simulations with m snapshots for $m = 100$ to $m = 1000$. The model of the wireless network consists of 4 cells, for a more detailed description on how the set of linear equations is formulated see Chapter 5. For each snapshot, users are positioned, the set of equations are formulated and missed traffic is estimated either by (3.9) or (3.11). The results are shown in Figure 3.2.

Comparison of the means of the estimators in Figures 3.2(a) and 3.2(b) shows that the mean of MT_2 is slightly lower than the mean of MT_1. Figures 3.2(c) and 3.2(d) show that the variance of estimator MT_2 is indeed slightly lower than for MT_1. The loss of accuracy of MT_2 by dismissing slow fading on certain paths between users and basestations is not compensated for by a significantly lower variance. Therefore, we conclude that it is better to use MT_1 to estimate missed traffic.

3.4 Implementation into a simulation tool

The network model described in the previous sections is a simplification of a UMTS network. With the exception for the model with $K = 1$, see Chapter 4, the model is still too complex for an analytical study. There are, at least, two major complications which stand in the way of a successful analytical study of the model. Firstly, there is, to our knowledge, no closed expression for the solution to either the set of linear equations (3.8) or (3.10). Secondly, the removal of users from the network affects the spatial Poisson process, by which the users are modeled, in an unpredictable manner. Therefore, we have to resort to simulation in order to study the model. In this section we shall explain the subsequent steps a simulation tool has to make to find estimates of missed traffic.

The simulation is initialized by choosing the number of cells K. The number of cells is decisive for the computation of MAI. For example if $K = 1$, there will only be SCI which raises the question how to model OCI. If $K = 4$, then cells have different numbers of neighbors, see Figure 3.1, this discrepancy has to be resolved. A solution to compute OCI in both cases is shown in Chapter 4 and Chapter 5, respectively. Note that this requires an additional assumption on the modeling of MAI.



(a) Average when (3.9) is used

(b) Average when (3.11) is used

(c) Deviation when (3.9) is used

(d) Deviation when (3.11) is used

Figure 3.2: Comparison of estimators (3.9) and (3.11)

The simulation constructs realizations of snapshots. For each realization, a number of users for each basestation $1, 2, \ldots, K$ is generated by sampling from the Poissonian distribution with mean $\lambda|C|$, see (B.1). The position of the users are determined by uniformly generating an angle θ between zero and 2π, while the distance to the basestation is generated by a uniform sample in the interval $(0, 1)$, see (3.2). This only gives the distance for a user to the basestation to which it is connected. The distances of users to the neighboring basestations can be determined by goniometry.

To compute the distances of users to neighboring cells, we need the distance between neighboring basestations. By elementary goniometry

$$D_b = 2\cos(\pi/6)R = \sqrt{3}R.$$

Furthermore, set a point of origin and a positive orientation for the angle θ_{kj}. We choose the point of origin to be the most right corner of the cell and the positive orientation to be clockwise. Suppose that the j^{th} user of basestation k of the network in Figure 3.1 has distance D_{kj} and angle θ_{kj}. Using trigonometry formulas, $D_{kj\ell}$ the distance of user j to basestation ℓ is

$$D_{kj\ell} = \begin{cases} D_{kj} & \text{if } \ell = k \\ \sqrt{D_b^2 + D_{kj}^2 - 2D_b D_{kj} \cos\alpha_{kl}} & \text{otherwise} \end{cases}$$

where $\alpha_{k\ell} = \alpha_{k\ell}(j)$ is tabulated in Table 3.1.

Based upon the positions of the users in their cells and their distances to neighboring basestations, the set of linear equations (3.7) can be formulated. Then we iteratively delete the users which for the average condition of the network would require a transmit power great than $P_{\max}$. The set $\mathcal{N}_U$ of users remain in the network. We then have two choices, either we use the already computed powers $(\widetilde{P}_k^{tx} j)_{\{k,j\}\in\mathcal{N}_U}$ and estimate missed traffic by (3.11). The other option is to first draw $|\mathcal{N}_U|$ samples from the normal distribution to obtain the slow fading factors and estimate missed traffic by (3.9). For the remainder of this paper we choose to use (3.11) to estimate missed traffic.

ℓ	α_{1l}	α_{2l}	α_{3l}	α_{4l}
2	$\theta_{kj} - 5/6\ \pi$	$\theta_{kj} + 1/6\ \pi$	$\theta_{kj} - 1/6\ \pi$	$\theta_{kj} - 1/2\ \pi$
3	$\theta_{kj} + 5/6\ \pi$	$\theta_{kj} + 1/2\ \pi$	$\theta_{kj} - 1/2\ \pi$	$\theta_{kj} - 2/3\ \pi$
4	$\theta_{kj} + 1/2\ \pi$	$\theta_{kj} + 1/3\ \pi$	$\theta_{kj} + 1/6\ \pi$	$\theta_{kj} - 4/6\ \pi$

Table 3.1: Computation of α_{kj}

Chapter 4

Single cell model

Not surprisingly, the single cell model only has one basestation. Since the model concerns one basestation, propagation loss, shadowing and transmit powers are all indexed by a single index that distinguishes users. The number of users in the cell is represented by N. The single cell model has two advantages. Firstly, it is irrelevant which estimator, (3.9) or (3.11), is used since all users are connected to one basestation. Secondly, the transmit powers of the users are mutually independent.

Since there is only one basestation, MAI solely consists of SCI. The MAI of the i^{th} user is modeled by multiplying the SCI of this user

$$\text{MAI} := \frac{N-1}{F}\ 10^{(S_i-\gamma_i)/10} P_i^{tx},$$

where $0 < F \leq 1$. Note that if $F = 1$, then the user only suffers from SCI. If F decreases, then the OCI increases.

For each user we determine its transmit power by finding the solution to the equation

$$\mu = \frac{G P_i^{tx} 10^{(S_i-\gamma_i)/10}}{\frac{N-1}{F}\ P_i^{tx} 10^{(S_i-\gamma_i)/10} + N_f}, \qquad \text{for } i = 1, 2, \ldots, N. \tag{4.1}$$

The solution to the linear equation is given by

$$P_i^{tx} = 10^{(\gamma_i - S_i)/10} \frac{F\ \mu\ N_f}{GF - (N-1)\mu}$$

Also by (4.1), the thermal noise N_f can be expressed in the transmit power of the i^{th} user

$$N_f = P_i^{tx} 10^{(S_i-\gamma_i)/10} \left(\frac{G}{\mu} - \frac{N-1}{F} \right) = \frac{G\ P_i^{tx} 10^{(S_i-\gamma_i)/10}}{\mu} - \text{MAI}$$

Let $\kappa = \kappa(n, F)$ be the noise rise or

$$\begin{aligned} \kappa = \frac{\text{MAI} + N_f}{N_f} &= \frac{G\ P_i^{tx} 10^{(S_i-\gamma_i)/10}}{G\ P_i^{tx} 10^{(S_i-\gamma_i)/10} - \mu\ \frac{N-1}{F}\ P_i^{tx} 10^{(S_i-\gamma_i)/10}} \\ &= \frac{GF}{GF - (N-1)\mu} \end{aligned} \tag{4.2}$$

The expression for the transmit power can be reduced to

$$P_i^{tx} = 10^{(\gamma_i - S_i)/10} \cdot \frac{\mu}{G} \cdot \kappa \cdot N_f. \tag{4.3}$$

4.1 Analysis without removal of blocked users

In the previous section it was mentioned that before the estimator of missed traffic is computed, users which in average condition require a transmit power larger than $P_{\max}$ are removed from the network. By ignoring this step we can find a closed expression for the distribution of missed traffic.

The most useful property of the single cell model is that when the number of users is fixed, the powers P_i^{tx} are mutually independent. The transmit power of the i^{th} user depends on the random variables D_i and S_i which are assumed to be mutually independent of D_j and S_j, $j \neq i$, respectively.

Firstly, we study the distribution of MT for a fixed number of users in the cell n. By the independence of the transmit powers, each user, which is at most R km away from the basestation, has probability $p(n, R)$ to require a transmit power greater than $P_{\max}$. The conditional density function of MT is

$$\mathbb{P}\left(MT = \frac{i}{n} \mid N = n\right) = \binom{n}{i} p(n,R)^i (1 - p(n,R))^{n-i}.$$

The conditional mean and variance of MT are

$$\begin{aligned} \mathbb{E}[MT \mid N = n] &= p(n,R) \\ \mathrm{Var}(MT \mid N = n) &= \frac{1}{n}\, p(n,R)(1 - p(n,R)). \end{aligned}$$

It remains to compute the probability $p(n, R)$.

User i has probability $p(n, D_i, R)$ to require a too large transmit power. The expression for this probability is derived in Section 4.1.1 but can be skipped on first reading. We show there that

$$\begin{aligned} p(n, D_i, R) &= \mathbb{P}(P_i^{tx} > P_{\max} \mid N = n, D_i \text{ and } D_i \leq R) \\ &= \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{\alpha + \kappa(n,F) + v_h \log_{10}(\max\{d_0, D_i\})}{\sqrt{2}\sigma_S}\right)\right), \end{aligned} \tag{4.4}$$

where

$$\alpha = u_f + u_h + \mu/G(\mathrm{dB}) + N_f(\mathrm{dBm}) - \widetilde{P}_{\max}(\mathrm{dBm}),$$

also $\kappa(n, F)$ is expressed in dB. The probability $p(n, R)$ can be obtained by integrating over the density function of D_i

$$p(n,R) = \mathbb{P}\left(P_i^{tx} > P_{\max} \mid N_i = n \text{ and } D_i \leq R\right) = \int_0^R p(n,x) f_D(x)\,dx,$$

where f_D is the density function of the random variable D_i, the derivative of (3.1). Substituting (4.4) into the above integral yields,

$$\begin{aligned} p(n,R) = \frac{1}{2} + \frac{d_0^2}{2R^2} &\operatorname{erf}\left(\frac{\alpha + \kappa(n,F) + v_h \log_{10}(d_0)}{\sqrt{2}\sigma_S}\right) \\ &+ \int_{d_0}^R \frac{x}{R^2} \operatorname{erfc}\left(\frac{\alpha + \kappa(n,F) + v_h \log_{10}(x)}{\sqrt{2}\sigma}\right) dx, \end{aligned} \tag{4.5}$$

where again $\kappa(n, F)$ is expressed in dB. The integral has an explicit yet involved solution, this solution can also be found in Section 4.1.1.

The probability mass function for MT is more complicated but can be retrieved with the help of the conditional density. The number of users is Poissonian distributed with mean $\lambda|C|$. However, the basestation has finite capacity and can only facilitate a finite number of users. By (4.2) and (4.3) it is possible to determine the maximum allowable number of users. The maximal number of users is

$$N_{\max} = \max\{n \ : \ n \geq 0 \text{ and } GF > (n-1)\mu\}.$$

Furthermore, there must be at least one user to measure performance. The probability mass function of the number of users in the cell is

$$\mathbb{P}(N = n) = \frac{(\lambda|C|)^n}{n!}\left(\sum_{i=1}^{N_{\max}} \frac{(\lambda|C|)^i}{i!}\right)^{-1} := c(F)\,\frac{(\lambda|C|)^n}{n!}, \qquad \text{for } n = 1, \ldots, N_{\max}. \tag{4.6}$$

By the law of total probability, the probability mass function for missed traffic becomes

$$\begin{aligned}\mathbb{P}(MT = x) &= \sum_{n=1}^{N_{\max}} \mathbb{P}(N = n)\mathbb{P}(MT = x|N = n) \\ &= \sum_{n=1}^{N_{\max}} \mathbb{P}(N = n)\binom{n}{nx} p(n, R)^{nx}(1 - p(n, R))^{n-nx}\mathbb{1}\{nx \in \mathbb{N}\},\end{aligned} \tag{4.7}$$

for $0 \leq x \leq 1$.

The expectation and variance of MT can be computed with the conditional expectation and variance. The expectation of MT equals

$$\mathbb{E}[MT] = \sum_{n=1}^{N_{\max}} \mathbb{P}(N = n)\mathbb{E}[MT \mid N = n] = c(F)\sum_{n=1}^{N_{\max}} \frac{(\lambda|C|)^n}{n!}p(n, R). \tag{4.8}$$

The variance of MT can be computed in the same way as was done for (2.6).

$$\begin{aligned}\text{Var}[MT] &= \mathbb{E}[\text{Var}(MT|N)] + \text{Var}(\mathbb{E}[MT|N]) \\ &= \mathbb{E}\left[\frac{p(N, R)(1 - p(N, R))}{N}\right] + \text{Var}(p(N, R))\end{aligned}$$

When substituting the distribution function of N into the above formula, we arrive at

$$\text{Var}(MT) = c(F)\sum_{n=1}^{N_{\max}} \frac{(\lambda|C|)^n}{n!}\frac{p(n, R)(1 + (n-1)p(n, R))}{n} - c(F)^2\left(\sum_{n=1}^{N_{\max}} \frac{(\lambda|C|)^n}{n!}\,p(n, R)\right)^2. \tag{4.9}$$

4.1.1 Derivation of $p(n, D_i, R)$ and $p(n, R)$

We express the power P_i^{tx} in dBm. Let $\widetilde{\mu} = 10\log_{10}(\mu/G)$, $\kappa(n, F)$ be expressed in dB and the thermal noise N_f in dBm. Furthermore, there are a total of n users connected to the basestation. By (4.3), it holds that

$$P_i^{tx} = \gamma_i - S_i + \widetilde{\mu} + \kappa + N_f.$$

Using Hata's model for propagation loss (3.6), the propagation loss factor γ_i equals

$$\gamma_i = u_f + u_h + v_h \log_{10}(\max\{d_0, D_i\}),$$

where D_i is the distance of user i to the basestation and d_0 is the cut-off distance. Let

$$\alpha = u_f + u_h + \widetilde{\mu} + N_f - P_{\max},$$

then

$$P_i^{tx} - P_{\max} = \alpha + v_h \log_{10}(\max\{d_0, D_i\}) - S_i + \kappa(n, F)$$

By the latter equation, the probability that user i suffers from missed traffic equals

$$\begin{aligned} p(n, D_i, R) &= \mathbb{P}(P_i^{tx} - P_{\max} > 0 \mid N = n, D_i) \\ &= \mathbb{P}(S_i < \alpha + \kappa(n, F) + v_h \log_{10}(\max\{d_0, D_i\})) \\ &= \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{\alpha + \kappa(n, F) + v_h \log_{10}(\max\{d_0, D_i\})}{\sqrt{2}\,\sigma_S}\right)\right) \end{aligned}$$

With the software package `Mathematica` we can find an explicit expression for (4.5).

$$\begin{aligned} p(n, R) = {} & \frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{\alpha + \kappa(n, F) + v_h \log_{10} R}{\sqrt{2}\sigma_S}\right) \\ & + \frac{1}{2R^2} 10^{\frac{2\sigma^2 \log 10 - 2v_h(\alpha + \kappa(n, F))}{v_h^2}} (A - B), \end{aligned}$$

where

$$\begin{aligned} A &= \operatorname{erf}\left(\frac{v_h^2 \log_{10} d_0 + (\alpha + \kappa(n, F))v_h - 2\sigma^2 \log 10}{v_h \sqrt{2}\sigma_S}\right) \\ B &= \operatorname{erf}\left(\frac{v_h^2 \log_{10} R + (\alpha + \kappa(n, F))v_h - 2\sigma^2 \log 10}{v_h \sqrt{2}\sigma_S}\right) \end{aligned}$$

4.2 Analysis with removal of blocked users

Recall that before missed traffic is estimated, the users which on average require a transmit power which exceeds $P_{\max}$ are removed from the network. Let N represent the total number of users in the cell before blocked users are removed, while $\mathcal{N}_U(N)$ denotes the number of users which are left after removal of the blocked users. By the independence of the positions of the users, it is still possible to find an analytical expression for the expectation of MT. However, because of the complexity of the model, we will build up the model in two steps.

Fix $N = n$ such that there are a total of n users in the network and let $MT(n, R)$ be the expectation of missed traffic when $N = n$ conditionally on the event that $\mathcal{N}_U > 0$. Recall that the MAI explicitly depends on the number of users present in the cell. Hence, starting with n users there will be a maximal allowable distance $d_{\max}(n)$. If the maximal distance exceeds $d_{\max}(n)$, then the user at maximal distance is removed from the network. In this case the maximal allowable distance will increase since the MAI decreases. We will work towards the model with a maximal distance which depends on the number of users in the cell. However, we first consider the case when the maximal allowable distance $d_{\max}(n) \equiv d_{\max}$ for all $n = 1, 2, \ldots, N$.

Let $D_{(1)}, D_{(2)}, \ldots, D_{(n)}$ be the distances of the n users rearranged in increasing order. The minimal and maximal distance of the n users will play an important role and we derive the distribution functions for $D_{(1)}$ and $D_{(n)}$. The complementary distribution function of the minimal distance $D_{(1)}$ among n users which are independently and homogeneously distributed over a circle with radius R is

$$F^c_{\min}(x; n, R) = \mathbb{P}\left(\min_{1\leq i\leq n} D_i \geq x\right) = \prod_{i=1}^{n} \mathbb{P}(D_i \geq x) = \begin{cases} 1 & \text{if } x < 0 \\ \left(1 - \frac{x^2}{R^2}\right)^n & \text{if } 0 \leq x \leq R \\ 0 & \text{if } x > R \end{cases} .$$

The distribution function of the maximal distance $D_{(n)}$ among n users which are independently and homogeneously distributed over a circle with radius R is

$$F_{\max}(x; n, R) = \mathbb{P}\left(\max_{1\leq i\leq n} D_i \leq x\right) = \prod_{i=1}^{n} \mathbb{P}(D_i \leq x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{x^{2n}}{R^{2n}} & \text{if } 0 \leq x \leq R \\ 1 & \text{if } x > R \end{cases} . \quad (4.10)$$

The network is initialized with a total of n users. The probability that the maximal distance among all n users is bounded by $d_{\max}$ is then simply $F_{\max}(d_{\max}; n, R)$. In this case the n users are uniformly distributed over a circle with radius $d_{\max}$ and none of them are removed. But this means that we satisfy the constraints of Section 4.1 and

$$\mathbb{E}[MT \mid N = n, D_{(n)} \leq d_{\max}] = p(n, d_{\max}).$$

With probability $1 - F_{\max}(d_{\max}; n, R)$, the maximal distance among all users exceeds $d_{\max}$. In this case we have to remove the user at maximal distance to the basestation. Removal of the user at distance $D_{(n)}$ leaves $n-1$ users which have distance $D_{(1)}, D_{(2)}, \ldots, D_{(n-1)}$ so that $D_{(i)} \leq D_{(n)}$ for all $i < n$. Now use the crucial property that the positions of the users in the network are mutually independent. Therefore, the $n-1$ remaining users are still uniformly distributed over a circle but no longer with radius R. Instead, the $n-1$ users are uniformly distributed over a circle with radius $D_{(n)}$. Hence, we have generated a snapshot for which the basestation has a smaller reach and we proceed as before.

The cdf of the maximal distance of $n-1$ users is $F_{\max}(x; n-1, D_{(n)})$. Now we consider $D_{(n-1)}$ and compare it with $d_{\max}$. The probability that the maximal distance among the $n-1$ users is bounded by $d_{\max}$ is $F_{\max}(d_{\max}; n-1, D_{(n)})$. In this case the conditional expectation of MT is

$$\mathbb{E}[MT \mid N = n, D_{(n)} > d_{\max}, D_{(n-1)} \leq d_{\max}] = p(n-1, d_{\max}).$$

With probability $1 - F_{\max}(d_{\max}; n, D_{(n)})$, the user at maximal distance has to be removed from the network. The previous steps can be repeated until no users are left in the network.

The procedure to compute missed traffic is depicted in Figure 4.1. The top row consists of the temporary states where it is checked if the maximal distance is bounded by $d_{\max}$. If the process is in state s_k, $2 \leq k \leq n$, and the maximal distance is bounded by $d_{\max}$, then the process goes to state s'_k. Otherwise the process goes to s_{k-1}. If the process is in state s_1 and the distance of the remaining user is not bounded by $d_{\max}$, then the process goes to state s'_0, otherwise the process goes to state s'_1. The same line of reasoning is possible when $d_{\max}$ is a function of the number of users yet to examine.

To illustrate the process we compute the expectation of missed traffic when the network is initialized with a single user.

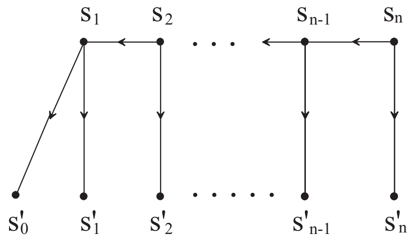


Figure 4.1: Process of removing users

Example 4.1. Suppose a single user is connected to the basestation and that it is located at distance D_1. Let d_1 be $d_{\max}(1)$, then $MT(1,R)$ is

$$MT(1,R) = \mathbb{E}[MT \mid N = 1, \mathcal{N}_U > 0] = \frac{\mathbb{P}(D_1 \leq d_1)p(1, d_1 \vee R)}{\mathbb{P}(\mathcal{N}_U > 0)},$$

where $x \vee y = \min\{x,y\}$. The event $\{\mathcal{N}_U > 0\}$ is equivalent to the event $\{D_1 \leq d_1 \vee R\}$, thus

$$MT(1,R) = \mathbb{E}[MT \mid N = n, \mathcal{N}_U > 0] = \frac{F_{\max}(d_1; 1, R)p(1, d_1 \vee R)}{1 - F^c_{\min}(d_1; 1, R)}.$$

The example is not very complex but enables us to compute $MT(n,R)$ for any $n \in \mathbb{N}$. Consider the variables

$$MT'(n,R) = (1 - F^c_{\min}(d_{\max}(1); n, R))MT(n,R).$$

With help of the following lemma we are able to compute $MT'(n,R)$ for any $n \in \mathbb{N}$.

Lemma 4.1. *The variable $MT'(n,R)$ satisfies the following recursive relation*

$$\begin{aligned} MT'(n,R) &= \mathbb{P}(D_{(n)} \leq d_{\max}(n))p(n, d_{\max}(n)) + \int_{d_{\max}(n)}^{R \vee d_{\max}(n)} f_{D_{(n)}}(x) MT'(n-1, x)dx \\ &= F_{\max}(d_{\max}(n); n, R)p(n, d_{\max}(n)) + \int_{d_{\max}(n)}^{R \vee d_{\max}(n)} f_{D_{(n)}}(x) MT'(n-1, x)dx, \end{aligned}$$

for $n \geq 2$ where

$$f_{D_{(n)}}(x) = \frac{d}{dx} F_{\max}(x; n, R)$$

Proof. Take $n \in \mathbb{N}$ and consider $MT'(n,R)$. We enter the stochastic process pictured in Figure 4.1 in state $s(n)$. If $D_{(n)} \leq d_{\max}(n)$, then we have a snapshot in which all users are uniformly distributed over a circle with radius $d_{\max}(n)$. This yields the first contribution to $MT'(n,R)$. If $D_{(n)} > d_{\max}(n)$, then we go to state $s(n-1)$ where the users are uniformly distributed over a circle with radius $D_{(n)}$. Observe that in this case the expectation of missed traffic is $MT'(n-1, D_{(n)})$. This yields the integral in the expression for $MT'(n,R)$. □

Now we proceed in exactly the same way as in Section 4.1. Based on the choice of F, we compute $N_{\max}$ and can so determine the expectation of missed traffic.

$$\mathbb{E}[MT] = \sum_{n=1}^{N_{\max}} c(F)\frac{\lambda^n}{n!}MT(n,R),$$

where $MT(n,R)$ is determined by use of Lemma 4.1.

4.3 Conclusions and recommendations

The great power of the single cell model is the assumption that MAI can be modeled by multiplying the SCI with an appropriate factor F^{-1}. This allows for a probabilistic analysis which is impossible for models which have multiple basestations. These latter models require solving the set of equations (3.4) for which no closed expression is known. A weakness of the model is that if there is only a single user connected to the basestation, then this user does not suffer from MAI.

The single cell model has the property that, conditionally on the number of users in the cell, the transmit powers are independent of one another. Based on this observation we are able to give a complete description of the distribution of missed traffic when no users are removed from the network. We have derived expressions for the distribution (4.7), the expectation (4.8) and variance (4.9) of missed traffic.

The case when blocked users are removed from the network is more complex. In Lemma 4.1 a recursive relation is derived which, theoretically, enables us to compute the expectation of missed traffic. However, the recursive relation for the expectation of missed traffic is complex and depends on taking multiple integrals and is thus computational intensive. Implementation of Lemma 4.1 shows that the instance when the network is initialized with 3 users already requires a substantial computational effort. Numerical integration methods may solve this problem but it will be hard to control errors since the errors propagate by the requirement to take multiple integrals.

The single cell model leaves us with a few issues which need to be resolved. We have found an expression for the expectation of missed traffic and analogue to Section 4.1 it should be possible to find expressions for the distribution and variance of missed traffic. More research on numerical integration methods is necessary for Lemma 4.1 to have practical value. Throughout Chapter 4 we have assumed the factor F to be some given constant. Obviously, this is not a realistic assumption and more research is needed to model F. In Section 5.3 we attempt to model F by study of the quadruple cell model. A probability distribution for F is chosen based on simulations of the quadruple cell model. In this section we also introduce the adapted single cell model which resolves the issue that a single user does not suffer from MAI.

Chapter 5

Quadruple cell model

In this chapter we shall introduce and analyze a second model of the wireless network. The model consists of 4 basestations which are placed and numbered as in Figure 3.1. The outline of this chapter is as follows. The basic problem for this model is how to model and compute the MAI, we have to make some additional assumptions which are explained in Section 5.1. In Section 5.2 we will study the distribution of missed traffic and in Section 5.3 we attempt to give a reduction from the quadruple cell model to the single cell model which was studied in Chapter 4.

5.1 Modeling MAI

It is assumed that the only users which contribute to the MAI of the j^{th} of basestation k are either connected to basestation k or to some basestation which is a direct neighbor of basestation k. The SCI of a user can be computed by Claim 3.1. However, it is impossible to compute the OCI for the simple reason that not all basestations are completely surrounded. To resolve this problem we use wrap around. Wrap around can be best explained by use of a window. Consider Figure 3.1, we put a window around the cell of basestation 1 and note the holes on the right side of the cell. These gaps are

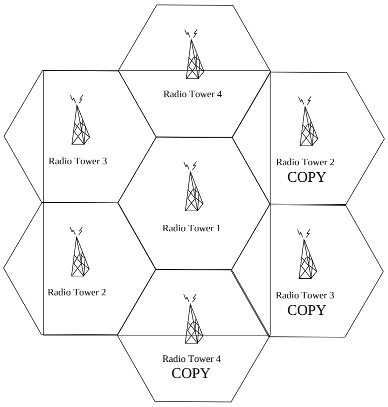


Figure 5.1: Wrap around for basestation 1

filled by using copies of the three other basestations (the copies include the users), this results in Figure 5.1.

Naturally the next step would be to move the window and place it over the cell of basestation 2, observe the gaps to the left of the cell and fill these up by copies of the other neighboring basestations. Note that in this case we would place a copy of a copy of cell 4 to the left of the cell of basestation 2. These steps are repeated until all cells are completely surrounded by 6 cells. Then it is possible to construct the system of linear equations (3.7) and, after removal of the blocked users, missed traffic is estimated by (3.11).

Estimations of missed traffic are purely based upon the transmit powers of the users, if present, which are connected to basestation 1, the so-called basestation of interest. Hence, $\mathcal{N}_U$ is the set of all users connected to basestation 1 which are not blocked. Using all users in the network to estimate missed traffic would provide a smaller variance of the estimator than when using only a subset of the users in the network. However, there is a good reason to appoint a basestation of interest.

As the title of this paper already suggests, the main goal of this paper is first classify snapshots into groups and to construct a tool which is able to generate snapshots from any group of desire. It became clear in Chapter 3 that we only directly control the spatial Poisson processes. Other parameters such as the gain G or the lower bound μ are imposed by the specifications of the network. If the estimator of missed traffic is based on all four of the cells, then we would need to classify missed traffic based on four independent spatial Poisson processes instead of one. These four Poisson processes have too many possible states, roughly the states for a single Poisson process to the power four, for a successful classification.

5.2 Missed traffic

In this section we shall study the distribution of missed traffic. Unlike with the single cell model, it is impossible to analytically analyze missed traffic and we have to turn to statistical techniques in order to draw conclusions about the distribution of missed traffic.

The two most used techniques to assess the distribution of some (continuous) random variable are probability plots and hypothesis tests. A probability plot [12] is a graphical technique to assess whether a distribution function is a good fit. The data is plotted against a theoretical distribution in such a way that the points should form approximately a straight line. Departures from this straight line indicate departures from the specified distribution. For conventional distributions such as the normal distribution, probability plots can be produced with `Statgraphics`. However, we could not find a statistical program which produces probability plots for a fully specified truncated normal distribution. Therefore, we implemented a program in `Mathematica` which produces probability plots for the truncated normal distribution. The program requires estimators for the parameters μ, σ, x_L and x_R of the truncated normal distribution, in Appendix C.2 it is explained how these parameters are estimated.

The usual approach to find the distribution of a random variable is to first draw some probability plots such that a small set of possible distribution functions remain. Then the hypothesis tests are used to see which distribution fits best. The general concept of hypothesis testing is treated in Appendix B.2.1. A goodness of fit test has hypotheses

H_0: the data follow a specified distribution,
H_a: the data do not follow a specified distribution.

In Appendix B.2.2 we focus on the three goodness of fit tests, the Kolmogorov-Smirnov (K-S), the Cramér-von-Mises (C-M) and the Anderson-Darling (A-D) test.

Figure 5.2(a) is the normal probability plot for missed traffic. Interpretation of the probability plot shows that the distribution $\hat{MT}_2$ is short-tailed, see [51, p. 71]. This is not surprising since by the removal of users which require a transmit power greater than $P_{\max}$, the estimator of missed traffic is bounded $0 \leq \hat{MT}_2 \leq 1/2$. Therefore we have also included a truncated normal probability plot, see Figure 5.2(b). The p-values for the different goodness of fit tests are tabulated in Table 5.1.

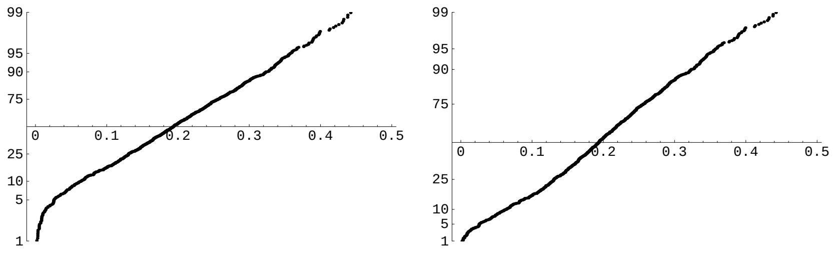
test	normal distribution	truncated normal distribution
	$\hat{\mu} = 0.195$, $\hat{\sigma} = 0.097$	$\hat{\mu} = 0.172$, $\hat{\sigma} = 0.115$ $\hat{x}_L = 0.00$, $\hat{x}_R = 0.49$
K-S	0.69	0.14
C-M	0.76	0.50
A-D	0.38	0.04

Table 5.1: p-values for goodness of fit tests of $\hat{MT}_2$

The probability plot for the truncated normal distribution is better behaved than for the normal distribution which defects for small values of missed traffic. However, the p-values of the goodness of fit tests indicate that the normal distribution fits the distribution of missed traffic better. This behavior can be heuristically explained by studying the estimators of missed traffic for individual users. The individual estimators of missed traffic are truncated normally distributed. Conditionally on $\mathcal{N}_U$, the estimators of missed traffic are also mutually independent. There are versions of the central limit theorem which imply the usual central limit theorem for sums of dependent random variables. Although missed traffic does not meet the conditions for any of these versions, the transition from the truncated normal distribution to the normal distribution best explains the behavior of missed traffic.

5.3 Reduction to single cell model

In Chapter 4 we found a closed expression for missed traffic. As was already mentioned, the expression for missed traffic is difficult. However, it still is useful to link the quadru-



(a) Normal probability plot (b) Truncated normal probability plot

Figure 5.2: Probability plots of $\hat{MT}_2$

ple cell model to the single cell model because of running time. The quadruple cell first positions users in four different cells and computes the distances from the users to all neighboring basestations. Then the systems of linear equations (3.7) must be solved to find the powers and to remove the blocked users from the network. The powers of the single cell model are directly computed by (4.3). Therefore, the computation time for the quadruple model is much greater than for the single cell model and a reduction to the single cell model perfectly fits in the search of methods to speed up the static simulation program.

Recall that the transmit power for the single cell model is given by (4.3), this yields

$$F = \frac{(\mathcal{N}_U - 1)P_{11}^{tx}\mu}{GP_{11}^{tx} - 10^{\gamma_{111}/10}N_f\mu},$$

where $\mathcal{N}_U$ is the set of users connected to basestation 1 which were not blocked. The estimate of F is obtained by considering the power of the first user connected to basestation 1. However, the choice of user does not make a difference since, by Claim 3.1, the received powers $10^{-\gamma_{1j1}/10}P_{1j}^{tx}$ are identical.

The single cell model has the property that when only one user is connected to the basestation, this user will not suffer from MAI. This property is likely to damage a possible link between the two models. Therefore we introduce the adapted single cell model. For this model we define the MAI to be

$$\text{MAI} = \left(\frac{\mathbb{1}\{|\mathcal{N}_U| = 1\}}{F_1} + \frac{\mathbb{1}\{|\mathcal{N}_U| > 1\}(n-1)}{F_2}\right)10^{(S_i - \gamma_i)/10}P_i^{tx}.$$

Note that the only change we make is that when there is one user connected to the basestation, then we model MAI interference by $(F_1)^{-1}P_1^{tx}$. The formula for the transmit power then becomes

$$P_i^{tx} = \begin{cases} \frac{F_1\mu N_f}{GF_1 - \mu} & \text{if } |\mathcal{N}_U| = 1, \\ \frac{F_2\mu N_f}{GF_2 - (\mathcal{N}_U - 1)\mu} & \text{if } |\mathcal{N}_U| > 1. \end{cases}$$

In case of the adapted single cell model, the factors F_1 and F_2 can be computed by

$$F_1 = \mathbb{1}\{\mathcal{N}_U = 1\}\frac{P_{11}^{tx}\mu}{GP_{11}^{tx} - 10^{\gamma_{111}/10}N_f\mu}$$
$$F_2 = \mathbb{1}\{\mathcal{N}_U > 1\}\frac{(n-1)P_{11}^{tx}\mu}{GP_{11}^{tx} - 10^{\gamma_{111}/10}N_f\mu}.$$

Note that F_2 is surely greater than zero but not bounded by 1. Indeed, simulations show that F_2 can be as large as 5.

We will first study the factors of the single cell and the adapted single cell model when there are at least two users connected to the basestation of interest. These snapshots will reveal information about the distribution of $F = F_2$. We then turn to the case when there is only one user connected to the basestation of interest. The snapshots with only one user connected to the basestation of interest will provide information about the distribution of F_1.

Recall from Chapter 4 that a truncated distribution is likely to arise. The variable F (F_2) is bounded between 0 and 1, where $F = 1$ when there is no OCI and as $F \to 0$ the OCI increases. The truncated normal probability plot in Figure 5.3 shows to be most the promising probability plot. The estimators for the parameters of the truncated normal distribution are displayed in Table 5.2. The second tool we have at our disposal are the goodness of fit tests. These tests also support the choice for the truncated normal distribution. Hence, we expect $F = F_2$ to be truncated normally distributed.

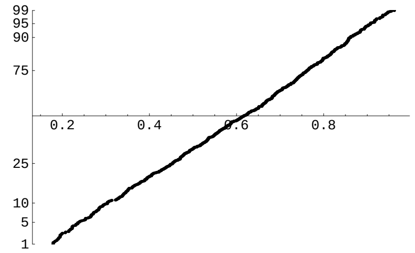


Figure 5.3: Truncated plot when $|\mathcal{N}_U| > 1$

test	Truncated normal distribution
	$\hat{\mu} = 0.624$, $\hat{\sigma} = 0.231$ $\hat{x}_L = 0.15$, $\hat{x}_R = 0.97$
K-S	0.519
C-M	0.785
A-D	0.636

Table 5.2: p-values of goodness of fit tests

Now we turn to the case when $|\mathcal{N}_U| = 1$. Study of the factor F_1 requires the introduction of another distribution, the exponential distribution. A random variable X is exponentially distributed with scale parameter λ, or $X \sim \text{Exp}(\lambda)$, if the density function of X is

$$f_X(x) = \lambda e^{-\lambda x}, \qquad \text{for } x \geq 0.$$

A random variable X is two-parameter exponentially distributed with scale parameter λ and location parameter γ, or $X \sim \text{Exp}(\lambda, \gamma)$, if the density function is

$$f_X(x) = \lambda e^{-\lambda(x-\gamma)}, \qquad \text{for } x \geq \gamma.$$

Note that the two-parameter exponential distribution is simply an exponential distribution which is truncated from the left at γ.

Trying several distribution functions, the exponential probability plot yields the most promising picture, see Figure 5.4. However, large values of F_1 distort the possibly exponential distribution of F_1. We try goodness of fit test for both the exponential and two-parameter exponential distribution. The p-values for the exponential distribution are small, the largest p-value is in the order of 10^{-11}. The p-values for the two-parameter exponential distribution are displayed in Table 5.3. Based on Table 5.3 and the probability plot, we sample the factor F_2 from the two-parameter exponentially distribution.

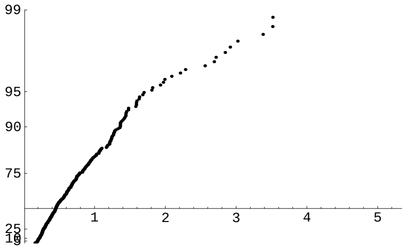


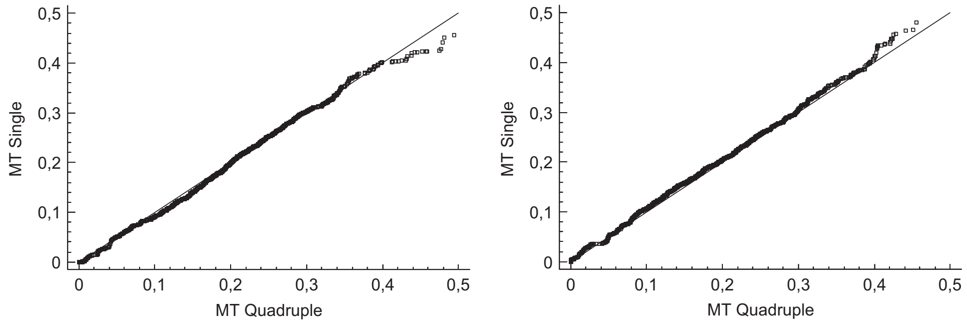
Figure 5.4: Exponential plot

test	two-parameter exponential
	$\hat{\gamma} = 0.14$, $\hat{\lambda} = 1.89$
K-S	0.021
C-M	0.089
A-D	0.012

Table 5.3: p-values goodness of fit tests

With the knowledge about the factors F, F_1 and F_2 we can generate samples from both the single cell model and the adapted single cell model. For the single cell model we sample F from the truncated normal distribution. For the adapted single cell model we sample F_1 from the two-parameter exponential distribution while F_2 is sampled from the truncated normal distribution. Comparing these two models with the quadruple cell model can be done by sampling from both models and compare these samples to a sample of the quadruple cell model. We use both graphical techniques and the K-S two-sample goodness of fit test to compare two samples.

The first graphical technique is the quantile-quantile (Q-Q) plot. The Q-Q plot is a technique similar to the probability plot. A Q-Q plot plots the empirical distribution function of the first data set against the empirical distribution function of another data set. A 45-degree reference line is also plotted. If the two sets come from a population with the same distribution, the points should fall approximately along this reference line. The greater the departure from this reference line, the greater the evidence for the conclusion that the two data sets have come from populations with different distributions. The statistical software package `Statgraphics` can be used to produce the Q-Q plots. In Figure 5.3, the Q-Q plots for both single cell models are shown.



(a) Single cell model (b) Adapted single cell model

Figure 5.5: Q-Q plots of missed traffic

The Q-Q plot for the single cell model shows that it underestimates missed traffic, this was already foreseen since a single user will not suffer from MAI. The adapted single cell model repairs this error and models OCI. However, it seems that the adapted single cell model overshoots its target. The adapted single cell model seems to overestimate missed traffic. We also note the different behavior between the single cell and the adapted single cell model appears to be present for large values of missed traffic. However, the models only differ in modeling the MAI for a single user. Therefore, one would expect the differences of the model to be present for smaller values of missed traffic.

A second graphical technique used to determine whether two data sets have a common distribution function is the density trace [35]. The density trace is an alternative to a commonly used simple density estimator, the histogram. Weaknesses of the histogram such as which choice to make for the width of the intervals or the number of intervals, caused the proposal of alternative density estimators. Defining the location density $d(x|h)$ at a point x as the fraction of the data values per unit of measurement that fall in an interval centered at x gives

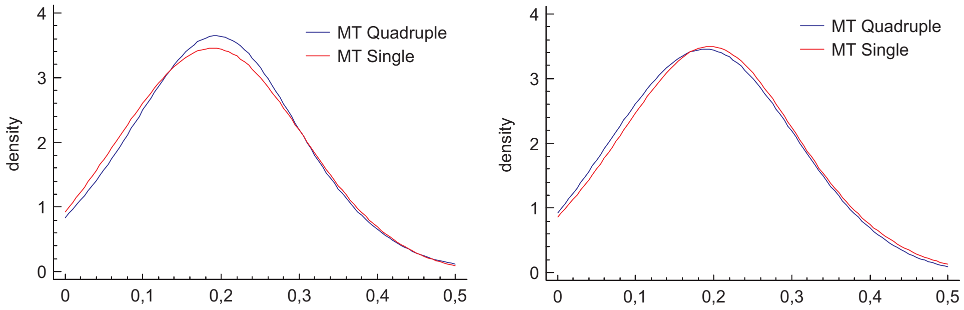
$$d(x|h) = \frac{\sum_{i=1}^{n} \delta_i}{n\lambda},$$

where n is the sample size, h is the interval width and δ_i is one when the i^{th} sample value is in the interval $[x-h/2, x+h/2]$ and zero otherwise. In order to plot the density trace, first select a value for h and then compute $d(x|h)$ on a dense grid of equally spaced x values. Connect the $d(x|h)$ by lines. The shape of the density trace is essentially driven by the interval length. It is very smooth for large values of h and “wiggly” for smaller values of h.

As the interval width is increased, data points further and further from the center value are included. In order to decrease the weight of points that are far removed from the center value each point gets an interval dependent weight w_i. The location density is given by

$$d(x|h) = \frac{\sum_{i=1}^{n} w_i \delta_i}{n\lambda}.$$

For a more detailed description about these weighting functions we refer to [12]. The statistical software package `Statgraphics` can be used to make density traces. `Statgraphics` includes all observations into a single interval and the weights are chosen according to a Gaussian distribution. The density traces for missed traffic are shown in Figure 5.3.



(a) Single cell model

(b) Adapted single cell model

Figure 5.6: Density traces of missed traffic

The density trace for the adapted single cell model seems to be a better fit than the single cell model. The density trace of the single cell model shows a higher density for smaller values of missed traffic in comparison to the density trace of missed traffic in the quadruple cell model. This makes sense since a single user is likely to cause smaller values of missed traffic. When there is only one user present in the model, the single cell model will give rise to an even lower value of missed traffic since this user does not suffer from MAI. The density trace of the adapted single cell model shows that this behavior is corrected. The density of smaller values of missed traffic is only slightly higher than the density for the quadruple cell model.

The last technique to compare the distributions of missed traffic is the K-S two sample test, see Appendix B.2.2 for a description of this test. The test is performed with `Statgraphics` which returns approximate p-values. The p-value for the single cell model is 0.225 while the p-value for the adapted single cell model is 0.396.

The Q-Q plots, density traces and p-values for the two-sample K-S test indicate that the distribution of missed traffic in the quadruple cell model is the same as the distribution of missed traffic in the adapted single cell model. However, studying the number of users in the cell of interest we see significant differences. It seems as if the single cell and adapted single cell model cause a shift of the distribution of $\mathcal{N}_U$ to the left.

Model/$\lvert\mathcal{N}_U\rvert$	1	2	3	4	5	6	7	8	9	Total
Single cell	88	195	290	255	115	41	15	1	0	1000
Adapted Single Cell	92	224	260	239	120	43	15	0	0	993
Quadruple cell	92	188	257	195	127	84	27	10	2	982

Table 5.4: The number of users $\lvert\mathcal{N}_U\rvert$

5.4 Conclusions and recommendations

The quadruple cell model which is likely to be more accurate than the single cell model because of the fact that multiple basestations are modeled. These neighboring basestations should give better approximations of OCI than simply multiplying the SCI by some factor as is done with the single cell model. The accuracy comes at the price that the computation of the transmit powers is more complex than for the single cell model. Simulations show that the use of the single cell model instead of the quadruple cell model roughly reduces running time by a factor five.

In order to compute OCI we use wrap around, see Figure 5.1. This causes the network to have a strange structure in which identical copies of only four cells form the entire network. To avoid this specific structure of the network, we propose to generate seven basestations instead of four. The new model places the basestation of interest and surrounds it by six basestations. The basestations at the periphery only have three neighboring basestations. Instead of placing identical copies of these three neighboring basestations, we assume that the OCI is symmetric. That is, the contribution to the OCI from users connected to the basestations which are not present in the model is identical to the contribution of users connected to the three neighboring basestations which are present in the model. In this way we avoid the peculiar structure of the network.

We observe that the reduction from the quadruple cell model to the both the single cell and adapted single cell model shows great potential. The results of Q-Q plots, density traces and the two sample K-S test give no reason to doubt the fact that missed traffic for both models is similarly distributed. The reduction to the adapted single cell model shows to be more accurate than the reduction to the single cell model. Study of the number of users which are not blocked cause us to doubt that the distribution of $\lvert\mathcal{N}_U\rvert$ is the same. More research is needed to explain the difference between the number of users and how it affects missed traffic for the (adapted) single cell model. We also propose to, analogue to Chapter 4, give a complete probabilistic analysis of the adapted single cell model because the reduction shows better potential than the single cell model.

Chapter 6

Discriminant analysis

The simulation program to estimate missed traffic in a wireless network should make it possible to rapidly draw conclusions for the performance of the network for several settings of the parameters. For example, suppose we have some reference model A, for which we have chosen a setting of the parameters. If simulation shows that for a certain subset of the snapshots model A performs poorly, then we would like to test whether a change of the parameters would improve performance for these snapshots. Obviously, we do not want to save all the snapshots of model A or keep on generating random snapshots with the changed parameters until the snapshots of interest arise by pure chance. Ideally, we would like to divide the snapshots into groups and make it possible to generate samples from each group separately. In this way we can study the performance for snapshots belonging to a certain group and rapidly test if a change of parameters would improve performance for this particular group of snapshots.

In this chapter we shall provide the first part of a solution to this problem, namely how to classify the snapshots into groups. The second part of the solution is to generate snapshots from a group of interest, this problem will be resolved in Chapter 7. In Section 6.1 we introduce a general classification rule to divide the snapshots over groups, the discussion is based on a paper by Friedman [27]. Section 6.2 deals with the case when the data in the groups is multivariate normally distributed. In Section 6.3 a hypothesis test is introduced to test data for multivariate normality. In this last section of this chapter we shall apply two classification rules to the quadruple cell model and analyze the results.

6.1 General classification rule

The purpose of classification is to assign objects to one of G groups based on a set of measurements $X = (X_1, X_2, \ldots, X_p)$ obtained from each object of observation. The variables X_i are the so-called feature variables. We will use classification techniques to study the separability of labeled groups of observations in the measurement space.

An object is assumed to be a member of a unique class and an error is incurred if it is assigned to a different one. The cost or loss associated with such an error is defined to be

$$L(g, \hat{g}), \qquad 1 \leq g, \hat{g} \leq G, \tag{6.1}$$

where g is the correct group and $\hat{g}$ is the assignment that was actually made. Usually $L(g,g)$ is taken to be zero and $L(g,\hat{g}) \geq 0$. A special but commonly occurring cost function is

$$L(g,\hat{g}) = 1 - \delta(g,\hat{g}), \tag{6.2}$$

which is zero if $g = \hat{g}$ and 1 otherwise.

The vectors of valued measurements which are assigned to the same group g are seldom identical but share a common probability density function $f_g(X)$. The usual goal is to minimize the expected misclassification loss (6.1) over the sample to be classified. If the class conditional densities $f_g(X)$ are known, then it is possible to calculate misclassification risk and derive an assignment or classification rule to minimize it. The expected loss or risk incurred in classifying an object with measurement vector X as $\hat{g}$ is

$$R(\hat{g} \mid X) = \sum_{g=1}^{G} \mathbb{P}(X \in g \mid X) L(g,\hat{g}),$$

The expression for $R(\hat{g} \mid X)$ can be written in an alternative manner using the following lemma:

Lemma 6.1. *Let X be a random variable with density function f_X and $B_1, B_2, \ldots$ be events. Suppose that*

i. B_i and B_j are disjoint whenever $i \neq j$,

ii. $\bigcup_i B_i$ is the entire sample space,

iii. $\mathbb{P}(B_i) > 0$ for each k.

Then

$$\mathbb{P}(B_i|X) = \frac{f_X(x|B_i)\mathbb{P}(B_i)}{\sum_j f_X(x|B_j)\mathbb{P}(B_j)} \ .$$

Proof. The proof is by using Bayes Theorem [46] which states that for random variables Y en Z with strictly positive joint probability density function on $\mathbb{R}^2$, we have

$$f_{Y|Z}(y|z) = \frac{f_{Z|Y}(z|y)f_Y(y)}{f_Z(z)}.$$

By the above formula it holds that

$$\mathbb{P}(B_i|X = x) = \frac{f_X(x|B_i)\mathbb{P}(B_i)}{f_X(x)} \ .$$

The random variables B_j are mutually independent and span the entire sample space. By the law of total probability it follows that

$$f_X(x) = \sum_j f_X(x|B_j)\mathbb{P}(B_j).$$

This completes the proof. □

Using Lemma 6.1 with B_g the event that $X \in g$ we see that the risk or expected loss incurred in classifying an object with measurement vector X as $\hat{g}$ is

$$
\begin{aligned}
R(\hat{g} \mid X) &= \sum_{g=1}^{G} \frac{f_g(X|B_g)\mathbb{P}(B_g)}{\sum_j f_g(X|B_g)\mathbb{P}(B_g)} \; L(g,\hat{g}) \\
&= \frac{\sum_{g=1}^{G} f_g(X)\pi_g \cdot L(g,\hat{g})}{\sum_{i=g}^{G} f_g(X)\pi_g}, \qquad (6.3)
\end{aligned}
$$

where π_g is the unconditional prior probability of observing a group g member.

The classification rule resulting from choosing $\hat{g}$ to minimize (6.3) is known as the Bayes rule and it achieves minimal misclassification risk among all possible rules [7, p. 159]. Therefore, the optimal rule minimizes the numerator of (6.3). When cost function (6.2) is used, misclassification risk is simply the fraction of assignments that are incorrect. The optimal rule is to choose $\hat{g}$ such that

$$
f_{\hat{g}}(X)\pi_{\hat{g}} = \max_{1 \le g \le G} f_g(X)\pi_g. \qquad (6.4)
$$

The class conditional densities $f_g(X)$ are often unknown. More often, we are able to obtain a sample of observations from each class that are correctly classified by some external mechanism. The objective is to use these observations as a training sample to construct a classification rule by obtaining suitable estimates of the class conditional densities $f_g(X)$. Since these estimates generally deviate from the true population densities, such a rule will not likely achieve minimal risk, except perhaps asymptotically. Sometimes the unconditional class probabilities are also unknown. If the training data can be regarded as a random sample from the pooled population, then the prior probabilities can be estimated by the fraction of each class in the pooled sample.

6.2 Linear and quadratic discriminant analysis

The most often applied classification rules are based on the multivariate normal distribution. This is when the group conditional density functions ($1 \le g \le G$) are given by

$$
f_g(X) = (2\pi)^{-p/2}|\Sigma_g|^{-1/2}e^{-1/2(X-\mu_g)^T\Sigma_g^{-1}(X-\mu_g)}, \qquad (6.5)
$$

where μ_g and Σ_g are the group g population mean vector and covariance matrix. We denote $X \sim \mathcal{N}_p(\mu, \Sigma)$ if X is a p-dimensional random variable and X obeys the law of the multivariate normal distribution with mean vector μ and covariance matrix Σ. When the covariance matrices are identical, i.e.

$$
\Sigma_g = \Sigma, \qquad 1 \le g \le G,
$$

the distribution is called homoscedastic normal, otherwise it is called heteroscedastic normal.

Assuming the simple loss function (6.2) and substituting (6.5) into (6.4) leads to the classification rule

$$
d_{\hat{g}}(X) = \min_{1 \le g \le G} d_g(X), \qquad (6.6)
$$

with

$$d_g(X) = (X - \mu_g)^T \Sigma_g^{-1} (X - \mu_g) + \log |\Sigma_g| - 2 \log \pi_g. \tag{6.7}$$

The quantity $d_g(X)$ is often called the discriminant score for the g^{th} class, whereas $d_g(X) + 2 \log \pi_g$ is referred to as the discriminant function.

Using the classification rules (6.6) and (6.7) is called quadratic discriminant analysis (QDA) since it separates the disjoint regions of the measurement space corresponding to each group by quadratic boundaries. An important special case occurs when all of the class covariance matrices are presumed to be identical (homoscedasticity). Then the classification rule is called linear discriminant analysis (LDA) because the classification rule becomes

$$d_{\hat{g}}(X) = \min_{1 \le g \le G} d_g(X),$$

with discriminant functions

$$d_g(X) = (\mu_g - X)^T \Sigma^{-1} \mu_g - (\mu_g)^T \Sigma^{-1} X - 2 \log \pi_g,$$

resulting in linear decision boundaries.

Quadratic and linear discriminant analysis can be expected to work well if the class conditional densities are approximately normal and good estimates, for classification purposes, can be obtained for the population parameters defining the distributions. In the classification context the ellipsoidal symmetry shape (Gaussian shape) associated with the normal distribution appears to be the important aspect rather than its detailed shape, see [37, 41].

In most applications of LDA or QDA the prior probabilities are estimated as

$$\hat{\pi}_g = \frac{n_g}{n},$$

with

$$n_g = \sum_{c(v)=g} 1, \qquad n = \sum_{g=1}^{G} n_g. \tag{6.8}$$

Here v labels the observations in the training sample and $c(v)$ is the class of the v^{th} observation. The parameters associated with the class densities are estimated by their sample analogues:

$$X_{g.} = \frac{1}{n_g} \sum_{c(v)=g} X_v \tag{6.9}$$

The unbiased estimator of the covariance matrix Σ_g is

$$S_g = \frac{1}{n_g - 1} \sum_{c(v)=g} (X_v - X_{g.})(X_v - X_{g.})^T. \tag{6.10}$$

When the covariance matrices are assumed to be identical, the unbiased estimator of Σ is

$$S = \sum_{g=1}^{G} \sum_{c(v)=g} \frac{n_g - 1}{n - G} S_g \tag{6.11}$$

6.3 A test for multivariate normality and homoscedasticity

The question of adequacy of QDA or LDA is one that can be fairly easy assessed. In principle one might test this in the same manner as one tests for univariate normality by using goodness of fit tests. However, the goodness of fit tests suffer from both computational and distributional errors [25]. Recall that the success of QDA and LDA depends on the fact that the class conditional density functions have a Gaussian shape rather than the specific shape of the densities. Therefore, we first perform two tests whether the class conditional density functions are Gaussian shaped. A third test is performed to test for homoscedasticity.

Skewness [58] can be used to assess the asymmetry of the probability distribution of a real-valued random variable. Roughly speaking, a distribution has positive skew (right-skewed) if the right tail is longer or fatter and negative skew (left-skewed) if the left tail is longer or fatter. Skewness, represented by γ_1, is defined as

$$\gamma_1 = \frac{\mathbb{E}[(X-\mathbb{E}[X])^3]}{\mathbb{E}[(X-\mathbb{E}[X])^2]^{3/2}} = \frac{\mathbb{E}[(X-\mathbb{E}[X])^3]}{\sigma^3} \ .$$

Clearly, for symmetric distributions the skewness parameter is zero and its size can be viewed as a measure of asymmetry.

Kurtosis [58] can be used to assess whether the distribution is heavy-tailed. Kurtosis, represented by γ_2, is defined as

$$\gamma_2 = \frac{\mathbb{E}[(X-\mathbb{E}[X])^4]}{\mathbb{E}[(X-\mathbb{E}[X])^2]^2} - 3 = \frac{\mathbb{E}[(X-\mathbb{E}[X])^4]}{\sigma^4} - 3.$$

Positive kurtosis indicates a "peaked" distribution and negative kurtosis indicates a "flat" distribution. The normal distribution is known to have zero kurtosis.

Suppose for the moment that we have a single sample $X_1, X_2, \ldots, X_n$ with sample mean

$$X. = \frac{1}{n}\sum_{i=1}^{n} X_i,$$

and covariance matrix

$$S = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - X.)(X_i - X.)^T.$$

Then the sample skewness and kurtosis are

$$b_1 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left((X_i - X.)^T S^{-1}(X_j - X.)\right)^3,$$

and

$$b_2 = \frac{1}{n}\sum_{i=1}^{n}\left((X_i - X.)^T S^{-1}(X_i - X.)\right)^2,$$

Mardia [49] shows that asymptotically, under a normal model,

$$A = \frac{nb_1}{6} \sim \chi^2_\nu,$$

with

$$\nu = p(p+1)(p+2)/6.$$

A random variable X is chi squared distributed with ν degrees of freedom, or $X \sim \chi^2_\nu$, if the density function of X is given by

$$f_X(x) = \frac{2^{-k/2}}{\Gamma(k/2)}\, x^{k/2-1} e^{-x/2}, \qquad x \geq 0.$$

Note that the skewness is not signed, it can only be tested whether skewness is present or absent, it can not be further classified in left or right skewed. Mardia [49] also proves that the sample kurtosis converges in distribution to a normal distribution.

$$B = \frac{b_2 - p(p+2)}{\sqrt{8p(p+2)/n}} \sim \mathcal{N}(0,1)$$

Returning to the classified data, the actual situation is slightly more complex since the sample data contains more than one population. The model is

$$X_{gj} \sim \mathcal{N}_p(\mu_g, \Sigma_g).$$

Provided that all n_g are sufficiently large, b_1 and b_2 may be computed separately for the samples from each source, yielding, say, b_{1g}, b_{2g} for $1 \leq g \leq G$. Fatti, Hawkins and Raath [25] suggest to perform the skewness and kurtosis test separately for each group.

The standard test for homoscedasticity is Bartlett's test [3]. It is sensitive to the assumption of normality and gives excessive Type I errors with heavy-tailed data. This is not a serious drawback if the skewness and kurtosis test are performed before Bartlett's test. Recall definitions (6.8) to (6.11), the test statistic for Bartlett's test is

$$M = \sum_{g=1}^{G} \frac{n_g - 1}{2} \, \log \frac{|S|}{|S_g|}.$$

Under the null hypothesis, the distribution of M can be approximated very closely in terms of the chi square distribution

$$\rho \log M \sim \chi^2_f,$$

with $f = \frac{1}{2}(G-1)p(p+1)$ and

$$\rho = 1 - \left(\sum_{g=1}^{G} \frac{1}{n_g} - \frac{1}{N-g} \right) \frac{2p^2 + 3p - 1}{6(p+1)(G-1)}.$$

Anderson [3] gives a criteria which ensures a good approximation by the chi squared distribution by introducing

$$\omega_2 = \frac{p(p+1)}{48\rho^2} \left((p-1)(p+2) \left(\sum_{g=1}^{G} \frac{1}{n_g^2} - \frac{1}{(N-g)^2} \right) - 6(G-1)(1-\rho)^2 \right),$$

Anderson then proves the following equality

$$\mathbb{P}(-\rho \log M \leq z) = \mathbb{P}(\chi^2_f \leq z) + \omega_2 \left(\mathbb{P}(\chi^2_{f+4} \leq z) - \mathbb{P}(\chi^2_f \leq z)\right) + O((N-G)^{-3}).$$

Hence, if ω_2 is sufficiently small in comparison with ρ, then the test statistic is well approximated by the chi square distribution.

6.4 Performance of discriminant analysis

Suppose we have a training set which consists of snapshots $S_1, S_2, \ldots, S_M$. The estimates of missed traffic for these snapshots are used to divide $S_1, S_2, \ldots, S_M$ over G groups. The performance of the network is measured by missed traffic. Therefore, we partition the interval $[0, 1/2]$ into G intervals and assign snapshot S_i to group g if missed traffic belongs to the g^{th} interval of the partition. We shall apply both LDA and QDA to see which method has the highest success rate.

Recall from Section 5.2 that missed traffic seems to fit the normal distribution and a logical partition would be based on the mean and variance of missed traffic. Let $\hat{\mu}$ and $\hat{\sigma}$ represent the sample mean and deviation for missed traffic of the training set. Then consider the following partition

$$\{(0, \hat{\mu} - c_{-i}\hat{\sigma}), \ldots, (\hat{\mu} - c_{-1}\hat{\sigma}, \hat{\mu}), (\hat{\mu}, \hat{\mu} + c_1\hat{\sigma}), \ldots, (\hat{\mu} + c_j\hat{\sigma}, 1/2)\},$$

for some i and j. This division has the advantage that it is possible to approximate the number of observations that will be assigned to each group. The groups are numbered from left to right, thus group 1 contains all observations with the smallest values for missed traffic while group G contains those with the highest values for missed traffic. For the remainder of this section, we set $i = j = 1$ and choose $c_{-1} = c_1 = 1$. Then the two smaller groups 1 and 4 will approximately contain 16% of the observations while the two larger groups 2 and 3 approximately contain 34% of the observations.

We need to select a set of possible feature variables which can be used to assign snapshots to the correct bin. Obviously, the most discriminative feature variables would entail information about the transmit powers of the users. However, the data must not only be classified but it should also be possible to generate snapshots from the groups. We can not directly control the transmit powers since they are computed by solving the set of equations (3.4). However, we do control the spatial Poisson process by which the users are modeled. Hence, we seek feature variables which are related to the Poisson process and are likely to influence the estimate of missed traffic.

The possible feature variables which we consider are

$X_1 = N_1$, the number of users which are connected to basestation 1

$X_2 = \sum_j D_j / N_1$, the mean distance,

$X_3 = \sum_j D_j^2 / N_1$, the mean squared distance,

where D_j is the distance of the j^{th} user connected to the basestation of interest. By Claim 3.1, the number of users which are connected to the basestation of interest directly affects SCI. The choice for feature variables X_2 and X_3 is also evident, the combination of the two can be used to estimate the mean and variance of the sum of all distances in the cell. A measure for the distance of the users provides information about propagation loss. We could also have chosen X_3 to be the standard deviation of the distances in the cell. However, by the central limit theorem it follows that X_3 tends in distribution to a normally distributed variable while this is not true for the standard deviation. Furthermore, it is important to stress that we consider the total group of users, including the users which will be removed. This is because we have no direct control over the process where users are removed from the network.

The variable X_1 can be used as feature variable for the discriminant analysis, this case is treated by Section 6.4.1. However, the variable X_1 is discrete and thus likely to deteriorate the performance of discriminant analysis. Therefore, we also try discriminant analysis when we condition on X_1, that is for each different value of X_i we apply discriminant analysis. The results of this approach are studied in Section 6.4.2. Based on the results in these two sections we choose a model which is studied in further detail in Section 6.4.3.

6.4.1 Discriminant analysis for variable number of users

The set of possible feature variables is small enough so that it allows us to apply LDA and QDA for every possible combination of feature variables. We have three measures M_1, M_2 and M_3 to evaluate the performance of a discriminant method. Measure M_1 is the percentage of the total observations that is assigned to the correct group. The second measure depends on the misclassified observations. Misclassified observations should be likely to be in the border area between the correct group and a neighboring group. We define M_2 to be the percentage of all observations that are not assigned to the correct group but to a direct neighboring group of the correct group. The measure M_3 simply is the sum of M_1 and M_2. In Appendix D we have tabulated the results of LDA and QDA which are omitted in this chapter.

Tables 6.1 and 6.2 tabulate the overall results for every possible set of feature variables. The first three columns show which feature variables are included in the discriminant analysis. For example the fourth row indicates that the discriminant analysis is based on feature variables X_1 and X_2. The last three columns show the values for the measures M_1, M_2 and M_3.

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	34.6	13.6	48.2
0	1	0	65.7	33.0	98.7
0	0	1	68.5	30.0	98.5
1	1	0	66.6	31.8	98.4
1	0	1	68.5	29.9	98.4
0	1	1	66.0	32.4	98.4
1	1	1	43.9	41.5	85.4

Table 6.1: Performance of LDA

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	31.7	20.0	51.7
0	1	0	64.3	34.6	98.9
0	0	1	66.0	32.6	98.6
1	1	0	66.8	32.0	98.8
1	0	1	67.9	30.5	98.4
0	1	1	62.0	35.9	97.9
1	1	1	15.0	32.6	47.6

Table 6.2: Performance of QDA

Note that when either the analysis is solely based on X_1 or on all three possible feature variables, both quadratic and linear discrimination perform badly. The differences between the other choices of feature variables are not decisive. For LDA the best setting seems to be to take X_3 as the only feature variable while the best results for QDA are obtained for the set of feature variables X_1 and X_3. We also observe that LDA performs better than QDA.

Based on the overall results we focus on LDA and exclude the models X_1 and (X_1, X_2, X_3). However, the differences between the models are not significant enough to automatically point out the best model. Since the models show similar overall performance, the performance for individual groups might be conclusive. The groupwise results for both LDA are tabulated in Tables 6.3-6.6.

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	18.7	35.5	54.2
0	1	0	54.2	37.0	91.2
0	0	1	62.0	36.8	98.8
1	1	0	56.0	35.6	91.6
1	0	1	62.0	29.0	91.0
0	1	1	54.8	36.8	91.6
1	1	1	54.8	45.2	100.0

Table 6.3: Performance LDA for group 1

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	28.0	72.0	100.0
0	1	0	64.7	35.5	100.0
0	0	1	70.4	29.6	99.7
1	1	0	66.3	33.7	99.7
1	0	1	70.6	29.4	99.7
0	1	1	67.4	32.6	99.5
1	1	1	93.8	6.2	100.0

Table 6.4: Performance LDA for group 2

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	66.6	31.5	98.1
0	1	0	74.8	25.2	100.0
0	0	1	72.9	27.1	100.0
1	1	0	74.8	25.2	100.0
1	0	1	71.3	28.7	100.0
0	1	1	72.9	27.1	100.0
1	1	1	0.0	0.0	0.0

Table 6.5: Performance LDA for group 3

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	0.0	37.4	37.4
0	1	0	61.6	38.4	100.0
0	0	1	61.6	38.4	100.0
1	1	0	61.6	38.4	99.3
1	0	1	64.6	35.4	100.0
0	1	1	60.3	39.7	100.0
1	1	1	0.0	0.0	0.0

Table 6.6: Performance LDA for group 4

Table 6.3 shows that for group 1 the models X_3 and (X_1, X_3) show the best performance for M_1. However, M_3 is significantly lower when using both the variables X_1 and X_3, thus we favor the model X_3. The results for the second group show that the models X_3 and (X_1, X_3) perform best but are not significantly different. For group 3 it seems that all models have more or less the same performance although the models X_2 and (X_1, X_2) are slightly better. For group 4, the models show similar performance. Hence, studying the groupwise performance of LDA we see that the models X_3 and (X_1, X_3) are favorable. The model with a single feature variable is simpler than with two feature variables. Therefore, we choose X_3 to be the only feature variable for linear discriminant analysis.

We have also tabulated the groupwise results for QDA, these results are displayed in the Appendix D as Tables D.1 to D.4.

6.4.2 Discriminant analysis for fixed number of users

The overall performance of the discriminant analysis for the model X_2 improves when the feature variable X_1 is added to the model while adding X_1 hardly affects the performance for the feature variable X_3. The variable X_1 is discrete, thus one would expect that adding X_1 would cause the performance to deteriorate because the normality assumption is certainly not satisfied. The p-values for the skewness test in Table D.9

support the idea that adding X_1 to either X_2 or X_3 harms the normality assumption. However, it could be that X_1 entails such vital information that it improves the performance of discriminant analysis despite harming the normality assumption. Therefore, it may be that the performance of LDA and QDA improves when we construct separate discriminant functions for each possible value of X_1. That is, for each possible value of $X_1 = n$, $n \geq 1$, we have a set of discriminants functions which are based on the collection of snapshots for which $X_1 = n$.

We collect from our training set all snapshots which have 4 users connected to the basestation of interest. We test the performance of LDA and QDA for discriminant functions based on this subset of snapshots. The results are then compared with the results obtained when we would have used the discriminant functions based on the entire set. Since we condition on X_1, only the variables X_2 and X_3 can be used as feature variables and there are three possible models. The results for both discriminant functions are displayed in Tables 6.7-6.10.

Model	M_1	M_2	M_3
X_2	59.1	38.8	97.9
X_3	56.3	41.6	97.9
(X_2, X_3)	57.0	40.9	97.9

Table 6.7: LDA, discriminant functions for $X_1 = 4$

Model	M_1	M_2	M_3
X_2	59.1	39.5	98.6
X_3	57.0	40.9	97.9
(X_2, X_3)	57.7	40.2	97.9

Table 6.8: LDA, overall discriminant functions

Model	M_1	M_2	M_3
X_2	55.6	41.5	97.1
X_3	57.0	40.1	97.1
(X_2, X_3)	50.0	47.1	97.1

Table 6.9: QDA, discriminant functions for $X_1 = 4$

Model	M_1	M_2	M_3
X_2	62.0	37.3	99.3
X_3	58.5	39.9	98.6
(X_2, X_3)	54.9	43.0	97.9

Table 6.10: QDA, overall discriminant functions

The overall values for the performance measures of LDA indicate that the discriminant functions which are based on the entire training set have a better performance than the when they are based on the snapshots for which X_1 equals 4. The conclusion for QDA is opposite, the performance for the overall discriminant function is worse. Although QDA improves when constructing separate discriminant function for $X_1 = 4$, the performance of QDA is not significantly better than the performance for the overall discriminant functions for LDA. Since constructing separate discriminant functions requires more work without a gain.

We have also tabulated the performance measure for each group separately, for LDA see Tables D.5 and D.6. These results show that it is better to use the linear discriminant functions which are based on the entire training set. The worse performance of the discriminant functions could partially be explained by the fact that the training set for which $X_1 = 4$ is too small. However, using a larger training set of 500 measurements only slightly increases the performance of these discriminant functions and the linear overall discriminant functions still perform better.

6.4.3 Final model

In Sections 6.4.1 and 6.4.2 we have tried several sets of feature variables to find out which one gives the best performance. Based on the fact that LDA seems to perform better than QDA we have to choose among the models (X_1, X_3) and the model which is based on the single feature variable X_3. When the discriminant function are solely based on X_3, the performance of LDA is slightly better than (X_1, X_3). Moreover, it is also a simpler model than (X_1, X_3). Therefore, we choose to apply linear discriminant analysis with feature variable X_3. Table 6.11 displays the results of LDA when using the single feature variable X_3.

Actual group	Group size	Predicted group			
		1	2	3	4
1	166	103 (62.0%)	49 (29.5%)	12 (7.2%)	2 (1.2%)
2	371	17 (4.6%)	261 (70.4%)	92 (24.8%)	1 (0.3%)
3	317	0 (0.0%)	71 (22.4%)	231 (72.9%)	15 (4.7%)
4	146	0 (0.0%)	0 (0.0%)	56 (38.4%)	90 (61.6%)
	p_g	85.8%	68.5%	59.0%	83.3%

Table 6.11: LDA with feature variable X_3

Purely looking at the percentages of correctly assigned observations, we see that LDA performs best for bins 2 and 3. We also note that observation which should have been assigned to group 2 have high probability to get assigned to group 3 while the probability to be assigned to bins 1 and 4 is quite small. A similar conclusion can be drawn for observations which should have been assigned to group 3. For the observations in group 1 we see that the LDA shows the worst behavior, observations are assigned to each bin and the probability to get assigned to the correct group or to a direct neighboring group lies around 0.9. For the other bins this probability is close to 1.

The results can also be interpreted in a different way. We estimate the probability to generate a snapshot from a group g by the relative frequency of observations that fall into the group g. For example, randomly generating a snapshot we have probability 0.16 to construct a snapshot from group 1. Next consider the relative probability to sample from group g this group is predicted by LDA. For this we consider the columns under the predicted groups. For example consider the column under predicted group 1. A total of 120 snapshots were assigned to group 1. Of these 120 snapshots, 103 actually belonged to group 1. Hence, we have a probability of 0.86 for a snapshot which is assigned to group to actually belong to group 1. Let p_g denote the probability for a snapshot assigned to group g to actually be a member of group g. These probabilities are tabulated in the last row of Table 6.11. These probabilities indicate that we need a factor 5 less snapshots for groups 1 and 4 while for groups 2 and 3 we need to generate only half as many snapshots.

The discriminant functions are displayed in Table 6.12. The second column shows the discriminant function while the third column displays the intervals where the discriminant functions are minimal. The allocation regions of the groups are as expected. Observations belonging to group 1 are assigned to the interval with smallest values while observations of group 2 are assigned to the interval with the second smallest value and so on.

Group	Score function	Interval
1	$13.08 - 65.49X_3$	$(0, 0.32)$
2	$24.28 - 100.25X_3$	$(0.32, 0.50)$
3	$37.84 - 127.39X_3$	$(0.50, 0.69)$
4	$66.74 - 169.16X_3$	$(0.69, 1)$

Table 6.12: Score functions and intervals

We are more interested in the performance of LDA for feature variable X_3 than whether X_3 satisfies the conditions of LDA. However, the p-values for X_3 for the skewness and kurtosis test in Tables D.9 and D.10 give no reason to doubt the Gaussian shape of the density function of X_3. Table D.11 shows that Bartlett's test for homoscedasticity is failed.

In the introduction to this section we emphasized that the choice for feature variable X_3 depends on the fact that we do not control the process in which blocked users are removed from the network. The performance of LDA improves when we use

$$X_3' = \sum_j D_j^2 / |\mathcal{N}_U|,$$

where the sum runs over all users which were not blocked. To illustrate the better performance we have generated a training set of 1000 sample values which is used to compute the discriminant functions. These discriminant functions are then used on another set of 1000 sample values to test performance. In this case, $M_1 = 83.2$ while $M_2 = 16.8$ which gives $M_3 = 100$.

Actual group	Group size	Predicted group			
		1	2	3	4
1	153	121 (79.1%)	32 (20.9%)	0 (0.0%)	0 (0.0%)
2	354	13 (3.7%)	297 (83.9%)	44 (12.4%)	0 (0.0%)
3	329	0 (0.0%)	28 (8.5%)	289 (87.8%)	12 (3.6%)
4	164	0 (0.0%)	0 (0.0%)	39 (23.8%)	125 (76.2%)
	p_g	90.2%	83.2%	77.6%	91.2%

Table 6.13: LDA with feature variable X_3'

6.5 Conclusions and recommendations

The results of Section 6.4 show that snapshots can be divided over groups by use of the feature variable X_3, the sum of squares of the distances from all users connected to the basestation of interest. Note that users which are blocked are included in X_3. In this case 60% of the snapshots are assigned to the correct group. Moreover, the results indicate that the running time can be reduced by a factor 2 to 3 when trying to generate snapshots from on of the G groups.

To measure the performance of the classification rules we used measures M_1 and M_2. The measure M_1 shows the percentage of observations which were assigned to the correct group while M_2 displays the percentage of observations which were not assigned to the correct group but to a neighboring group of the correct group. It would be interesting to have some distance measure which enables us to measure how far observations are away from the correct group.

We observe that if we were to be able to directly sample from the allocation regions we have one major problem. This problem can be best explained by considering the results in Table 6.11 for group 1. Here we see that only 60% of the snapshots which should have been assigned to group 1 are also assigned to group 1. Hence, when we would sample from the allocation region of group 1 we only sample from this subset of 60% of all snapshots which belong to group 1. This leaves the question whether sampling from the sampling regions gives an representative sample of snapshots from group 1.

The results of Section 6.4 also show that the performance of linear discriminant analysis significantly improves when we leave out the blocked users. That is when data is classified by use of X_3', the sum of squares of the distances of the users which are not blocked (sum runs over $\mathcal{N}_U$). In this case 83% of the snapshots are assigned to the correct group. The difficulty of using X_3' instead of X_3 is that, to generate snapshots from the groups, one needs to directly control the positions of the users which are not blocked. Therefore, more knowledge on the process of removing blocked users is required.

The quadruple cell model still allows to generate large sets of sample values of missed traffic. However, future models will become more complex and the running time of the simulation program will increase. This calls for methods to determine the minimum sample size required to distinct the G allocation regions of the sample space. We also used the standard maximum likelihood estimators for the mean and covariance matrix (the covariance matrix estimate is scaled by a factor to remove bias). Although seemingly reasonable, this approach can be justified only on intuitive grounds and enjoys no optimality properties. Information on how to determine minimal sample size and different estimates of the mean vector and covariance matrix can be found in [27, 36, 50, 66]

Classification rules should not only make it possible to divide snapshots over groups but should also allow to generate samples from the groups. Therefore, we consider the allocation regions which arise from the use of discriminant analysis. For a single feature variable no problems arise because, for any classification rule, the allocation regions will be simple intervals. The sole use of the feature variable X_3 shows best performance for LDA. This allows for other classification rules which might perform better than LDA. More examples of classification rules are presented by McLachlan [50].

Future research may give reason to use multiple feature variables. However, use of multiple feature variables can give problems for classification rules other than LDA. The allocation regions for LDA are simple linear boundaries, for example for $G = 2$ the boundary of the allocation region is a simple straight line or hyperplane. Other classification rules give more complicated allocation regions. For example for $G = 2$ and with QDA the region of allocation may be the interior of an ellipse or the region between two hyperbolas. The “simple” allocation regions which arise by use of LDA should make it likelier to generate snapshots from a group of desire than when using other classification rules.

Chapter 7

Generating snapshots with a certain characteristic

In Chapter 6 we divided the snapshots over four groups based on their value for missed traffic. We then showed that it is possible to, with high probability, generate snapshots from these groups by controlling the statistic X_3 defined to be

$$X_3 = \sum_{j=1}^{N_1} D_j^2/N.$$

Here N represents the total number of users connected to basestation 1, blocked users included, and D_j is the distance of the j^{th} user connected to the basestation of interest. The clue is that one should be able to generate snapshots for which X_3 lies in some prescribed interval, see Table 6.12.

In this chapter we shall introduce a method to sample from the distribution when we condition on the sum of squares of the distances of the users. That is, we provide a solution to the problem how to sample from the distribution described by the following pdf

$$f_{D_1,D_2,\ldots,D_n}(d_1, d_2, \ldots, d_n \mid \alpha \leq \sum D_i^2 \leq \beta). \tag{7.1}$$

The outline of this chapter is as follows. We use convex geometry to sample from the conditional distribution. Therefore, Section 7.1 gives a short introduction into this topic of mathematics. In Section 7.2 we shall use the theory of convex geometry to study the sample space of the distribution function (7.1). Moreover, a theoretical solution to the problem is provided. In Section 7.3 it is explained how the theoretical answer can be implemented in a simulation tool. The algorithms introduced in Section 7.3 are given in pseudo-code such that the reader can transcribe it into the language of his/her choice.

7.1 Convex geometry

In this section we shall treat the basic definitions and theorems of convex geometry. The foundations of convex geometry shall be used to analyze the sample space of the conditional distribution. For more information on convex geometry we refer to [20, 30, 31, 48]. We will make no distinction in notation for scalars and vectors, from the context it will be clear whether the variable is meant to be a vector or scalar.

7.1.1 Definitions and theorems

Given two points $x, y \in \mathbb{R}^n$, the line segment between x and y is the set of points

$$[x, y] = \{\lambda x + (1 - \lambda)y \mid \lambda \in [0, 1]\}.$$

A set $C \subseteq \mathbb{R}^n$ is *convex* if and only if for any pair of points $x, y \in C$ the closed line segment $[x, y] \subseteq C$.

Let $x_1, x_2, \ldots, x_k, z \in \mathbb{R}^n$, then z is called a *convex combination* of $x_1, x_2, \ldots, x_k$ when

$$z = \sum \lambda_i x_i, \quad \text{for } \lambda_1, \ldots, \lambda_n \text{ such that } \sum \lambda_i = 1 \quad \text{and } \lambda_i \geq 0 \ \forall i.$$

The *convex hull* of the set $\{x_1, x_2, \ldots, x_k\} \subseteq \mathbb{R}^n$ is the set of all convex combinations, or

$$hull\{x_1, x_2, \ldots, x_k\} = \left\{\sum \lambda_i x_i \mid \sum \lambda_i = 1 \text{ and } \lambda_i \geq 0 \ \forall i\right\}.$$

Similarly, z is an *affine combination* of $x_1, x_2, \ldots, x_k$ if and only if

$$z = \sum \lambda_i x_i, \quad \sum \lambda_i = 1.$$

The *affine hull* of the set $\{x_1, x_2, \ldots, x_k\} \subseteq \mathbb{R}^n$ is the set of all affine combinations, or

$$aff(x_1, x_2, \ldots, x_k) = \left\{\sum \lambda_i x_i \mid \sum \lambda_i = 1 \text{ and } \lambda_i \in \mathbb{R} \ \forall i\right\}.$$

Let $P \subseteq \mathbb{R}^n$, then P is a *polyhedron* if there is a system of finitely many inequalities $Ax \leq b$ such that

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

Moreover, P is a *polytope* if there exists a finite set $X \subseteq \mathbb{R}^n$ such that

$$P = hull\{X\}.$$

A *face* of a polytope is defined with the help of the inequalities $Ax \leq b$.

Definition 7.1. *Let A be a real $m \times n$ matrix, let $b \in \mathbb{R}^n$ and let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Let $a_1, a_2, \ldots, a_m$ be the rows of A. Then F is a face of P if and only if*

$$F = \{x \in P \mid a_i x_i = b_i \textit{ for all } i \in J\} \quad \textit{for some } J \subseteq \{1, 2, \ldots, m\}.$$

Faces of dimension 0 are called vertices, faces of dimension 1 edges and faces of dimension $d - 1$ facets.

Let $C \subseteq \mathbb{R}^n$. A vector $x \in C$ is an extreme if there are no $y, z \in C \setminus \{x\}$ such that $x \in [y, z]$. The set of all extreme points of C is denoted with $\text{Ex}(C)$. An important theorem concerning the set of extreme points of C is due to Minkowski, see [31].

Theorem 7.1. *Let $C \subseteq \mathbb{R}^n$ be a compact convex set. Then*

$$C = hull\{\text{Ex}(C)\},$$

where a set is compact if and only if it is bounded and closed.

Thus, the extreme set contains all vertices of the polytope P.

As it turns out, the building blocks of polytopes are simplices. Therefore, we will shortly introduce these special kind of polyhedrons. The n-simplex S_n with vertices $v_0, v_1, \ldots, v_n$ is defined to be the convex hull of the $n+1$ affinely independent vectors

$$S_n = hull\{v_0, v_1, \ldots, v_n\},$$

The vertices $v_0, v_1, \ldots, v_n$ are affinely independent if $\sum_i \lambda_i v_i = 0$ and $\sum_i \lambda_i = 0$ implies that $\lambda_0 = \lambda_1 = \ldots = \lambda_{n+1} = 0$.

The volume of a simplex S_n, denoted by $\text{Vol}(S_n)$, can be computed by the following formula [29, p. 403].

Theorem 7.2. *Suppose that S_n is a n-simplex with vertices $v_0, v_1, \ldots, v_n$. Let $B = (\beta_{ik})$ denote the $(n+1) \times (n+1)$ matrix given by $\beta_{ik} = ||v_i - v_k||^2$. Then*

$$2^n (n!)^2 \text{Vol}(S_n)^2 = |\det(\hat{B})|, \tag{7.2}$$

where $\hat{B}$ is the $(n+2) \times (n+2)$ matrix obtained from B by bordering B with a top row $(0, 1, \ldots, 1)$ and left column $(0, 1, \ldots, 1)^T$.

The basis of the method to sample from the conditional distribution is to decompose a polytope into simplices. The following deceivingly simple statement guarantees the existence of such a decomposition. For a proof of the theorem see [31]. Furthermore, let $\text{int}(S)$ denote the interior of the set S.

Theorem 7.3. *Any convex polytope P has a simplicial decomposition. That is*

$$P = \bigcup S_i \quad \textit{and} \quad \text{int}(S_i) \cap \text{int}(S_j) = \emptyset \textit{ for all } i \neq j,$$

where the S_i's are simplices.

In the literature, the division of a convex polytope into simplices is called a triangulation.

7.2 Theoretical solution

In this section we shall give a solution how to sample from the conditional distribution (7.1). The description only has theoretically value. How the solution can be implemented is explained in Section 7.3.

The solution to the problem of sampling from (7.1) lies in the properties of the random variable D_i, the distance of user i to the basestation in its cell. Based on the properties of D_i, we firstly show that the problem reduces to uniformly sampling a vector from some convex hull $P_\alpha(n)$. We then compute the vertices of the polytope $P_\alpha(n)$. Finally in Section 7.2.3, we combine the previous results and introduce an algorithm to sample from (7.1).

7.2.1 Analysis of the sample space

The sample space of the pdf (7.1) is defined as follows

$$\Omega_{\alpha,\beta}(n) = \{d_1, d_2, \ldots, d_n \mid \alpha \leq \sum_{i=1}^{n} d_i^2/n \leq \beta \text{ and } 0 \leq d_i \leq R \; \forall i = 1, 2, \ldots, n\}$$

for some α and β. By (3.2), the distance D_i is equal to $R\sqrt{U_i}$ where U_i is uniformly distributed on $(0,1)$. Hence, $\Omega_{\alpha,\beta}(n)$ can also be described as

$$\Omega_{\alpha',\beta'}(n) = \{u_1, u_2, \ldots, u_n \mid \alpha' \leq \sum_{i=1}^{n} u_i \leq \beta' \text{ and } 0 \leq u_i \leq 1 \ \forall i = 1, 2, \ldots, n\}, \quad (7.3)$$

where $\alpha' = n\ \alpha/R^2$ and $\beta' = n\ \beta/R^2$.

We have translated our problem to sampling from the sample space defined by (7.3). Although the sample space is described by uniformly distributed random variables, the points are not uniformly distributed in $\Omega_{\alpha',\beta'}(n)$. This can be seen from the distribution function of the sum of n uniformly distributed random variables [21, p. 245]. Let $U^{(n)} = U_1 + U_2 + \ldots + U_n$ where the U_i are i.i.d. uniformly distributed random variables, then

$$f_{U^{(n)}}(x) = \frac{1}{2(n-1)!} \sum_{k=0}^{n} \binom{n}{k} (x-k)^{n-1} \operatorname{sign}(x-k), \quad (7.4)$$

for $0 \leq x \leq n$ and where

$$\operatorname{sign}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ 1 & \text{if } z > 0 \end{cases}$$

We have plotted the density function (7.4) for several choices of n in Figure 7.1, it rapidly converges to a Gaussian shaped density function.

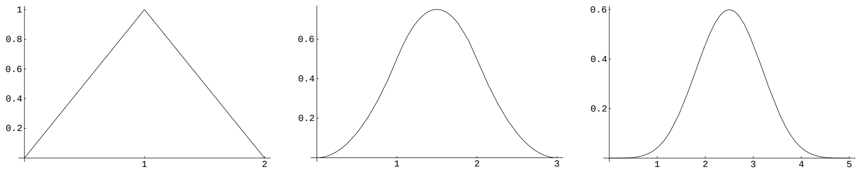
By the density function (7.4) it follows that not all vectors $(u_1, u_2, \ldots, u_n)$ have equal value for the density. The density depends on the sum of the entries of the vector. Conditioning on the sum of the entries yields a sample space

$$P_\gamma(n) = \{u_1, u_2, \ldots, u_n \mid \sum_{i=1}^{n} u_i = \gamma \text{ and } 0 \leq u_i \leq 1 \ \forall i = 1, 2, \ldots, n\},$$

where $\alpha' \leq \gamma \leq \beta'$. Now each vector $(u_1, u_2, \ldots, u_n) \in P_\gamma(n)$ has equal density since the pdf

$$f_{U_1,U_2,\ldots,U_n}(u_1, u_2, \ldots, u_n) = \frac{f_{U_1}(u_1) f_{U_2}(u_2) \cdots f_{U_n}(u_n)}{\mathbb{P}((U_1, U_2, \ldots, U_n) \in P_\gamma(n))}$$

is constant for all vectors in $P_\gamma(n)$. Moreover, the sample space $P_\gamma(n)$ can be described



(a) $n = 2$ (b) $n = 3$ (c) $n = 5$

Figure 7.1: Density function for sum of n uniformly distributed random variables

by a finite number of inequalities.

$$P_\gamma(n) = \{u_1, u_2, \ldots, u_n \mid \sum_{i=1}^{n} u_i \leq \gamma,\ -\sum_{i=1}^{n} u_i \leq -\gamma, \tag{7.5a}$$

$$-u_i \leq 0 \text{ and } u_i \leq 1\ \forall i = 1, 2, \ldots, n\}, \tag{7.5b}$$

Hence, the sample space is a polyhedron. The polyhedron is also contained in a finite set, namely the hypercube HC_n defined by

$$HC_n = \{x \in \mathbb{R}^n \mid 0 \leq x_i \leq 1 \text{ for all } i = 1, 2, \ldots, n\}.$$

Finally, $P_\gamma(n)$ is convex so that we can conclude that $P_\gamma(n)$ is a convex polytope. Therefore, by Theorem 7.3, there exist a simplicial decomposition of $P_\gamma(n)$.

7.2.2 Vertices of the sample space

The convex polytope $P_\gamma(n)$ can be seen as the intersection of the hypercube HC_n with a simplex S_n which is spanned by the vertices $\{\gamma \cdot e_1, \gamma \cdot e_2, \ldots, \gamma \cdot e_n\}$. Examples of the intersection of H_3 with S_3 for different choices of γ are shown in Figures 7.2(a)-7.2(c).

Usually it is difficult to find the vertices of a polytope. However, due to the fact that $P_\gamma(n)$ is the intersection of a hypercube with a simplex, the vertices have specific structure.

Lemma 7.1. *The vertices of $P_\gamma(n)$ are given by all permutations of the vector*

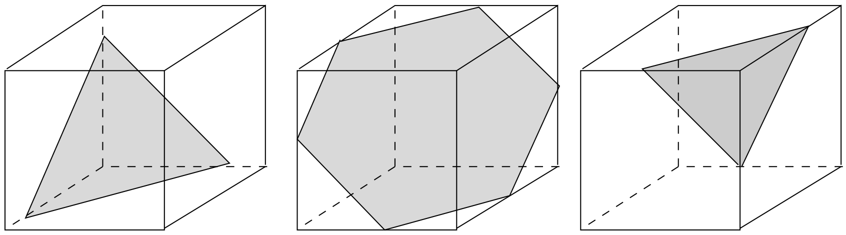
$$v_0 = (\underbrace{1, \ldots, 1}_{\lfloor\gamma\rfloor}, \underbrace{0, \ldots, 0}_{n-\lfloor\gamma\rfloor-1}, \gamma - \lfloor\gamma\rfloor)^T.$$

Proof. To prove that all permutations of v_0 are vertices it suffices to show that they are extreme points. Suppose that there are $y, z \in P_\gamma(n) \setminus \{v_0\}$ such that $v_0 = \lambda y + (1-\lambda)z$ for some $0 < \lambda < 1$. Obviously,

$$\lambda y_i + (1-\lambda) z_i = 1, \qquad \text{for all } i = 1, 2, \ldots, \lfloor\gamma\rfloor,$$

but y_i and z_i themselves are in the interval $[0, 1]$ and $0 < \lambda < 1$ thus $y_i = z_i = 1$. Similarly, it is possible to show that $y_i = z_i = 0$ for all $i = \lfloor\gamma\rfloor + 1, \lfloor\gamma\rfloor + 2, \ldots, n-1$. Since the vectors y and z are elements of the polytope, it immediately follows that $y_n = z_n = \gamma - \lfloor\gamma\rfloor$ which is in contradiction to the fact that $y, z \in P_\gamma(n) \setminus \{v_0\}$. The latter line of reasoning can be done for any permutation of v_0, thus all permutations of v_0 are extreme points. It remains to show that there are no other extreme points.

Recall Definition 7.1 and the fact that vertices are faces of dimension zero. The polytope $P_\gamma(n)$ is described by the constraints (7.5a) and (7.5b). Note that all the vertices we found sofar satisfy $n + 2 - \lceil\gamma - \lfloor\gamma\rfloor\rceil$ constraints. This is no coincidence, Schrijver [56] proves that faces of the same dimension satisfy exactly the same number of constraints. If there were to be another vertex of $P_\gamma(n)$ than it would also need to satisfy $n+2-\lceil\gamma - \lfloor\gamma\rfloor\rceil$ constraints. However, by construction there are no more points in the sample space which satisfy that many constraints. Therefore, we conclude that there are no other vertices. □



(a) $0 < \gamma \leq 1$ (b) $1 < \gamma \leq 2$ (c) $2 < \gamma \leq 3$

Figure 7.2: Examples of $P_\gamma(3)$

7.2.3 Sampling algorithm

As was shown in Section 7.2.1, the problem of sampling from the conditional distribution function (7.1) translates to uniformly sampling from the convex polytope $P_\gamma(n)$ described by the inequalities (7.5a) and (7.5b). If the polytope $P_\gamma(n)$ is a simplex, there would be no problem. A solution to the problem of uniformly sampling from a simplex is long known. The method to uniformly sample from a simplex has a natural extension to the more general case of uniformly sampling from convex polytopes.

By Theorem 7.3, any convex polytope P can be decomposed into simplices. For each of these simplices S_i, the uniform sample has probability $p_i = \mathrm{Vol}(S_i)/\sum_j \mathrm{Vol}(S_j)$ to be in the simplex S_i. The sampling algorithm randomly selects a simplex S_i with probability p_i from the collection of all simplices in the simplicial decomposition of P. The algorithm continues by drawing a uniform sample from the simplex S_i. The subsequent steps of the sampling algorithm are displayed as Algorithm 1.

Algorithm 1 Sampling algorithm

```
1: Draw a sample $\gamma$ of random variable $U^{(n)}$ with pdf (7.4) such that $\alpha \leq \gamma \leq \beta$.
2: Compute the vertices of the polytope $P_\gamma(n)$ described by (7.5a)-(7.5b).
3: Find a simplicial decomposition of $P_\gamma(n)$, thus $P_\gamma(n) = \bigcup S_j$.
4: Compute the volumes of the simplices and randomly pick a simplex from the collection of simplices$\{S_j\}_j$ such that the $i^{th}$ simplex has probability $\mathrm{Vol}(S_i)/\mathrm{Vol}(P_\gamma)$ to be selected.
5: Uniformly sample from the simplex which was picked in 4.
```

Careful study of the sampling algorithm shows that it does not necessarily sample uniformly over the polytope $P_\gamma(n)$. Suppose that the polytope $P_\gamma(n)$ is not a simplex but has a simplicial decomposition $\{S_i\}_i$. Let $\mathcal{F}$ be defined as

$$\mathcal{F} = \{x \in F \mid F \text{ is a facet of at least two simplices}\}.$$

When executing the sampling algorithm, the density for $x \in \mathcal{F}$ is at least $2/\mathrm{Vol}(P_\gamma)$ which would contradict the uniform distribution. Fortunately, the event $\{x \in \mathcal{F}\}$ is a zero-probability event which guarantees the correct working of the sampling algorithm.

7.3 Implementation

In Section 7.2.3 we introduced the sampling algorithm (Algorithm 1) which returns a sample from the distribution function (7.1). The description of the algorithm consists of five steps. Steps 2 and 4 do not require further explanation, they are covered by Theorem 7.2 and Lemma 7.1. However, steps 1, 3, and 5 do require some explanation on how these can be implemented in a simulation program. To each step we dedicate a subsection where we show how it can be implemented and provide an algorithm in pseudo-code so that the implementation can be transcribed into the programming language of choice.

7.3.1 Sample sum of uniform variables

The first step in Algorithm 1 is to find a sample γ from $U^{(n)}$, the sum of i.i.d. uniformly distributed random variables such that $\alpha \leq \gamma \leq \beta$. Or to put it differently, given $n \in \mathbb{N}$ and $\alpha, \beta \in (0, n)$ such that $\alpha \leq \beta$, draw a sample of the random variable $U^{(n)} = \sum_{i=1}^{n} U_i$ such that

$$\alpha \;\leq\; \sum_{i=1}^{n} U_i \;\leq\; \beta.$$

Partition the interval $[\alpha, \beta]$ in the following way

$$\{I_1, I_2, \ldots, I_{\lceil\beta\rceil - \lfloor\alpha\rfloor}\} = \{[\alpha, \lfloor\alpha\rfloor + 1), [\lfloor\alpha\rfloor + 1, \lfloor\alpha\rfloor + 2), \ldots, [\lfloor\beta\rfloor, \beta]\}.$$

The distribution function of the sum of n i.i.d. uniformly distributed random variables is given by (7.4). For each subinterval I_i of (α, β), compute the probability p_i for $U^{(n)}$ to fall into that specific interval. Also compute the conditional cumulative distribution function F_i for each subinterval I_i, that is

$$F_i(u) = \mathbb{P}(U^{(n)} \leq u \mid U^{(n)} \in I_i).$$

To sample $U^{(n)}$ we firstly draw a sample from a uniformly distributed random variable on $(0, 1)$. Based upon the outcome and the probabilities p_i we choose the interval I_i and thus the conditional distribution function F_i. Sampling from the cdf F_i can be done by drawing another uniform sample on $(0, 1)$, say U, and computing $F_i^{-1}(U)$ such that $F_i^{-1}(U) \in I_i$.

The final step, computing $F_i^{-1}(U)$, can only be done analytically for $n \leq 4$ because there is no general formula for the zeros of a polynomial with degree greater than 4. Therefore, we have to resort to numerical rootfinding methods. There are many rootfinding algorithms to choose from [4, 9]. We choose to use Brent's method.

The method is too complicated to treat in this paper, but a complete description can be found in [8]. The input for Brent's method is an interval $[a, b]$ in which a root must be found and a number δ such that a certain accuracy is guaranteed. For a function f and some interval $[a, b]$, the method returns a value $x \in [a, b]$ such that $|f(x)| \leq \delta$.

The choice for Brent's method is based upon three reasons. The first reason is that the rootfinding algorithm should only return the unique root in the interval I_i, this restriction is certainly satisfied by Brent's method. Secondly, Brent's method is known to be safeguarded, it falls back on the bisection method if necessary which, although slowly, always converges. Thirdly, Brent's method is standardly delivered with mathematical software packages such as `Mathematica` and `MatLab`.

7.3.2 Finding a simplicial decomposition

In the previous section, we have discussed step 1 of Algorithm 1 and have obtained a sample γ from the random variable $U^{(n)}$. Let $V(A)$ be the set of vertices of the convex hull of A. Step 2 of the algorithm is handled by Lemma 7.1 so we have a set $V(P_\gamma(n))$ which contains the vertices of $P_\gamma(n)$. The next step of the algorithm is to find a triangulation of $P_\gamma(n)$. In this section we shall introduce an algorithm to find such a triangulation. Recall that a triangulation of $P_\gamma(n)$ is a simplicial decomposition of $P_\gamma(n)$.

Finding simplicial decompositions of convex hulls continues to attract attention in the literature, for example see [5, 13, 38, 29, 57, 61]. One of the most straightforward algorithms is due to Seidel [57] and is also known as *beneath-and-beyond.* The correct working of beneath-and-beyond is proved in [38]. The algorithm finds a triangulation of the polytope P defined to be the convex hull of the vertex set $V = \{v_1, v_2, \ldots, v_k\}$, $k \geq n$, where $v_i \in \mathbb{R}^n$. Beneath-and-beyond requires an ordering of the vertices such that $v_1, v_2, \ldots, v_n$ are affinely independent.

To explain the algorithm, we need two more definitions. A facet F of a polytope P is *visible* from v if and only if the affine hyperplane spanned by the vertices of F separates v from P. A facet F of P *bounds* P if and only if the affine hyperplane spanned by the vertices of F does not intersect the polytope P. For example, consider the polytope $P = \mathit{hull}\{v_1, v_2, v_3\}$ of Figure 7.3(a). The facets which bound P are F_1, F_2 and F_3. The affine hull of two different points is the line through them. Therefore, the only facet that is visible from v_4 is F_3. The definition of visibility agrees with intuition and drawing sketches in two or three dimensions, one could suspect that for each vertex there is only one visible facet. However, this is generally not true for higher dimensions where there can be multiple visible facets.

Before going into details, we will give an intuitive description of the algorithm. Let $P_\gamma(n,k)$ be the convex polytope described by (7.5a) and (7.5b) with vertex set $V = \{v_1, v_2, \ldots, v_k\}$, $k \geq n$, which is described by Lemma 7.1. Also, $P_\gamma(n, n+i)$ is the convex hull of the vertex set $\{v_1, v_2, \ldots, v_{n+i}\}$. The algorithm decomposes $P_\gamma(n,k)$ into $(n-1)$-simplices. The algorithm is initialized by setting S_n to be the $(n-1)$-simplex spanned by the vertices $v_1, v_2, \ldots, v_n$. The triangulation T_n is set to be $T_n = \{S_n\}$ and the boundary B_n is the set of all facets of S_n.

Assume that, for some $i \geq 0$, a triangulation T_{n+i} of the polytope $P_\gamma(n, n+i)$ is already obtained. Consider the vertex v_{n+i+1} and locate all facets F_j which are contained in B_{n+i} and are visible from v_{n+i+1}. Connect v_{n+i+1} to each visible facet F_j by connecting it to all its vertices. For each visible facet F_j this will yield a simplex $S_{n+i+1,j}$ and thus the triangulation $T_{n+i+1} = (\bigcup S_{n+i+1,j}) \bigcup T_{n+i}$ decomposes $P_\gamma(n, n+i+1)$ into simplices. Furthermore, B_{n+i+1} is the set of all facets F which bound P_{n+i+1}.

In Figure 7.3 we pictured a single increment of beneath-and-beyond. We have been given the simplex $\mathit{hull}\{v_1, v_2, v_3\}$ with the boundary $\{F_1, F_2, F_3\}$ and we are at the point to add v_4 to the triangulation. The first step is to determine the visible facets. The facet F_3 is the only visible facet since it separates v_4 from v_2. Therefore, v_4 is connected to v_1 and v_3 and the facets F_4 and F_5 are added to the boundary. Note that the facet F_2 is no longer part of the boundary and the new boundary becomes $\{F_1, F_3, F_4, F_5\}$.

After the short introduction of the beneath-and-beyond algorithm, we can give a detailed description of the implementation of the algorithm. The algorithm is also displayed in pseudo-code as Algorithm 2 on page 61. The program can be split up in four parts namely Initialization, Determining visible facets, Supplementary boundary and triangulation and Updating triangulation and boundary. To each part we devote a paragraph in which we explain the logic of these steps and, when necessary, give detailed descriptions of procedures. The numbers in the paragraph titles refer to the line numbers of Algorithm 2.

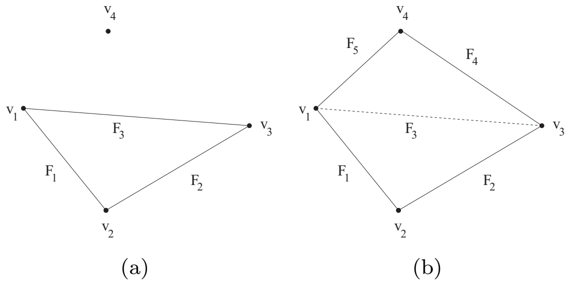


Figure 7.3: Increment of beneath-and-beyond

Initialization (1-3)

The algorithm requires an ordering of the vertices $\{v_1, v_2, \ldots, v_k\}$ such that the first n vectors are affinely independent. The ordering is used by defining $P_\gamma(n, n)$ to be the simplex spanned by the vertices $v_1, v_2, \ldots, v_n$. Using Lemma 7.1 we show that we can do better than affine independence and find a set of n linearly independent points to initialize the algorithm.

Since $\gamma \in \mathbb{N}$ is a zero-probability event, we assume that $\gamma \notin \mathbb{N}$. However, random number generators may cause problems and thus is the procedure Sort stopped when $\gamma \in \mathbb{N}$. Let $m = \lfloor \gamma \rfloor$, the points in the linearly independent set are given by

$$(\underbrace{0, \ldots, 0}_{i-1}, \gamma - m, \underbrace{0, \ldots, 0}_{n-m-i}, \underbrace{1, \ldots, 1}_{m})^T \tag{7.6}$$

for $i = 1, \ldots, n - m$, and

$$(\underbrace{0, \ldots, 0}_{n-m-1}, \underbrace{1, \ldots, 1}_{i}, \gamma - m, \underbrace{1, \ldots, 1}_{m-i})^T \tag{7.7}$$

for $i = 1, 2, \ldots, m$.

After the vertices are sorted in proper order, the program continues by setting $T = \{S_n\}$ where S_n is the convex hull spanned by the vectors $\{v_1, v_2, \ldots, v_n\}$. The boundary B_n is set to be the set of all facets of S_n. Hence, B_n consists of the convex hulls of all distinct subsets of $\{v_1, v_2, \ldots, v_n\}$ with cardinality $n - 1$.

Determining visible facets (5-10)

Suppose $P_\gamma(n, n + i)$ is already triangulated for some $i \geq 0$ and we want to add the vertex v_{n+i+1}. The lines of code from 5 to 10 construct the set $\mathcal{F}$ which contains the visible facets. The set $\mathcal{F}$ is obtained from the boundary B_{n+i} by deleting all facets which are not visible from v_{n+i+1}. Question is how to determine whether a facet is visible. In the following discussion we will use shorthand notation and omit the indices. The procedure Visible, which determines if a facet is visible, can be found at the end of this paragraph.

```
Procedure Sort($V$)
    if $\sum v_i \notin \mathbb{N}$ then
        Construct set of $n$ vectors $V' = \{v_1, v_2, \ldots, v_n\}$ as in (7.6) and (7.7)
    else
        break
    end if

    Returns: $V' \cup (V \setminus V')$
end Procedure
```

Let A be the affine hull of the vertices of some facet F. Then A separates $\mathbb{R}^n$ into two affine halfspaces A^+ and A^- with $A^+ \cup A^- = \mathbb{R}^n$ and $A^+ \cap A^- = A$. We will always assume that $P \subset A^+$ and thus F is visible from v if and only if $v \in A^-$. Furthermore, there exists a vector c such that c is perpendicular to the facet F but c is still parallel to the polytope P. This is a direct consequence of the fact that a facet is a $(n-2)$-dimensional object while the polytope is an $(n-1)$-dimensional object. In the Euclidean space $\langle c, y\rangle = ||c||_2||y||_2 \cos(\theta)$, where θ is the angle between vectors c and y. The polytope P is contained in A^+ if and only if for all vertices v_j of P, the inner product with c is non-negative (non-positive). Therefore, the affine hull A separates v from P if and only if the inner product $\langle c, v\rangle$ is positive (negative).

By the following lemma we can check whether a facet is visible to a vertex by simply comparing signs of determinants.

Lemma 7.2. *Let $x_1, x_2, \ldots, x_{n-1} \in \mathbb{R}^n$ be the vertices of a facet F of the induced triangulation of P. Furthermore, c is such that c is perpendicular to F while it lies parallel to P and $\mathbf{n}$ is perpendicular to all the points in the polytope P.*
Then there exists some $\delta \in \mathbb{R}$ such that for all $y \in \mathbb{R}^n$

$$\langle c, y\rangle = \delta \det(x_2 - x_1, x_3 - x_1, \ldots, x_{n-1} - x_1, \mathbf{n}, y),$$

where $\det(a_1, a_2, \ldots, a_k)$ *denotes the determinant of the $k \times k$-matrix with rows $a_1, a_2, \ldots, a_k$.*

Proof. Let $x'_i = x_{i+1} - x_1$. Since the set $x_1, x_2, \ldots, x_{n-1}$ is a facet of a simplex in the triangulation of P, the points are affinely independent. The vectors $x'_1, x'_2, \ldots, x'_{n-2}$, c and $\mathbf{n}$ are linearly independent and they form a basis of $\mathbb{R}^n$. The basis can be transformed to an orthogonal basis, for example by the Gram-Schmidt procedure. By the description of the Gram-Schmidt procedure, this will only require adding multiples of rows. Let $\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_{n-2}, c, \mathbf{n}$ be the orthogonal base of $\mathbb{R}^n$, note that c and $\mathbf{n}$ are not altered since they are already perpendicular to the facet F. The vectors $\widetilde{x}_i$ are obtained by adding multiples of rows, hence

$$\det(x'_1, x'_2, \ldots, x'_{n-2}, \mathbf{n}, y) = \det(\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_{n-2}, \mathbf{n}, y).$$

Let $y \in \mathbb{R}^n$, then it can be written as

$$y = \frac{\langle \widetilde{x}_1, y\rangle}{||\widetilde{x}_1||^2}\,\widetilde{x}_1 + \frac{\langle \widetilde{x}_2, y\rangle}{||\widetilde{x}_2||^2}\,\widetilde{x}_2 + \ldots + \frac{\langle \widetilde{x}_{n-2}, y\rangle}{||\widetilde{x}_{n-2}||^2}\,\widetilde{x}_{n-2} + \frac{\langle \mathbf{n}, y\rangle}{||\mathbf{n}||^2}\,\mathbf{n} + \frac{\langle c, y\rangle}{||c||^2}\,c.$$

By adding multiple of the first $n-1$ rows to y we get

$$\begin{aligned}\det(\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_{n-2}, \mathbf{n}, y) &= \det(\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_{n-2}, \mathbf{n}, \frac{\langle c, y\rangle}{||c||^2}\ c) \\ &= \langle c, y\rangle\, ||c||^{-2} \det(\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_{n-2}, \mathbf{n}, c)\end{aligned}$$

This completes the proof, since the determinant and norm do not depend on y. □

The vector $\mathbf{n}$ which is perpendicular to all vectors in the polytope P is $(1, 1, \ldots, 1)^T$. This is because the polytope $P_\gamma(n, k)$ is contained in the $(n-1)$-dimensional hyperplane H_γ where

$$H_\gamma = \{x \in \mathbb{R}^n \mid (1, 1, \ldots, 1)(x_1, \ldots, x_n)^T = \gamma\}.$$

Consider the $(n-1)$-dimensional hyperplane H_0

$$H_0 = \{x \in \mathbb{R}^n \mid (1, 1, \ldots, 1)(x_1, \ldots, x_n)^T = 0\}.$$

If H_γ were to be parallel to H_0 we are done because the all-one vector is perpendicular to H_0 and thus also perpendicular to H_γ. To show that H_γ and H_0 are parallel hyperplanes, we calculate the distance between any $x \in H_\gamma$ and the hyperplane H_0. The Euclidean distance between a point and a hyperplane is defined to be

$$d(x, H_0) = \min_{y \in H_0} \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}.$$

By use of Cauchy-Schwarz we see that

$$\sum_{i=1}^{n} (x_i - y_i)^2 \geq \left(\sum_{i=1}^{n} (x_i - y_i) \right)^2 = \gamma^2.$$

Therefore, we conclude that H_0 and H_γ are parallel to each other and, consequently, the all-one vector is perpendicular to all vectors in the polytope $P_\gamma(n, k)$.

Deciding whether to connect v to a facet F of the boundary can be done by computing determinants. The function sg is defined by

$$sg(V(F), y) = \text{sign}(\det(x_2 - x_1, x_3 - x_1, \ldots, x_{n-1} - x_1, \mathbf{n}, y), \tag{7.8}$$

for $y \in \mathbb{R}^n$ and $V(F) = \{x_1, x_2, \ldots, x_{n-1}\}$. Recall that $V(A)$ is the set of vertices for a convex hull A. We can determine whether the facet F_j of the boundary of the polytope $P_\gamma(n, n+i)$ is visible in the following way. If $sg(V(F_j), v_{n+i+1}) = 0$, then the volume of $hull\{V(F_j) \cup \{v_{n+i+1}\}\}$ is zero, because of the affine dependency of the vectors, and we can safely set the facet F_j to be not visible.

If $sg(V(F_j), v_{n+i+1}) \neq 0$, then we track down the unique simplex S in T_{n+i} such that F_j is a facet of S. Hence, the simplex S is $hull\{x_1, x_2, \ldots, x_n\}$. Then compute $sg(V(F_j), x_n)$ and compare it with $sg(V(F_j), v_{n+i+1})$, if they are opposite then F_j is visible from v_{n+i+1}. It is crucial that we use the vector x_n to compute $sg(V(F_j), x_n)$ because generally the vertices of the polytope $P_\gamma(n, n+i)$ are affinely dependent. If we were to arbitrarily choose a vertex u of $P_\gamma(n, n+i)$, then this vertex could be affinely dependent from the vertices of the facet and $sg(V(F), u) = 0$ which proved no information. The previous steps are executed by the procedure VISIBLE which returns true if F_j is visible from v_{n+i+1} and false otherwise.

Supplementary boundary and triangulation (11-16)

After the visible facets are determined, simplices S_j are created by connecting v_{n+i} to each visible facet F_j, that is $S_j = hull\{V(F_j) \cup \{v_{n+i}\}\}$. For every simplex S_j, the set

```
Procedure VISIBLE($V(F), v, T$)          ▷ $T$ triangulation and $F$ (boundary) facet

    $y \leftarrow sg(V(F), v)$                                      ▷ defined by (7.8)
    bool←False

    if $y \neq 0$ then
        Search $S \in T$ such that $S = hull\{V(F) \cup \{x_n\}\}$.
        if $sg(V(F), x_n) = -y$ then
            bool←True
        end if
    end if

    Returns: bool
end Procedure
```

of facets $\mathcal{F}_j$ consists of all distinct subsets of $n-1$ points of the vertex set of S_j. If a facet F occurs in at least two distinct sets $\mathcal{F}_j$ and $\mathcal{F}_k$, $j \neq k$, then it has to be removed because it will certainly not bound $P_\gamma(n, n+i+1)$. The set B' is the subset of $\mathcal{F}_j$ of all facets which have cardinality 1.

Updating triangulation and boundary (17-18)

The final step of the program is to set T_{n+i+1} and B_{n+i+1}. The triangulation of $P_\gamma(n, n+i+1)$ consists of the already existing triangulation of $P_\gamma(n, n+i)$ together with the simplices which were created by connecting v_{n+i+1} to the visible facets. The new boundary B_{n+i+1} consists of the union of the previous boundary B_{n+i} and the collection of facets B'. However, note that the set of visible facets $\mathcal{F}$ is a subset of B'. Therefore, the final step is to delete these facets to obtain the boundary of $P_\gamma(n, n+i+1)$.

7.3.3 Uniformly sampling from a simplex

We have now completed the first three steps of Algorithm 1. The first part of step 4 of the algorithm is handled by Lemma 7.2 so we continue by randomly picking a simplex S from the triangulation of $P_\gamma(n)$ with probabilities proportional to the volumes of the simplices in the triangulation. The final step of the program is to obtain a uniform sample from the simplex S.

Given affinely independent points $v_1, v_2, \ldots, v_n$ which represent the vertices of the simplex S, we would like to uniformly sample from S. We claim that Algorithm 3 does exactly that. The correct working of the algorithm is based on the following theorem.

Theorem 7.4. *Let $(U_{(i)})_{i=1}^{n-1}$ be i.i.d. uniformly distributed variables in the interval $[0,1]$ sorted in increasing order. Furthermore, the random variables W_i, $1 \leq i \leq n$, are defined to be*

$$W_i = \begin{cases} U_{(1)} & \text{if } i = 1, \\ U_{(i)} - U_{(i-1)} & \text{if } i = 2, \ldots, n-1, \\ 1 - U_{(n-1)} & \text{if } i = n. \end{cases} \tag{7.9}$$

Algorithm 2 Beneath-and-beyond in $\mathbb{R}^n$

```
Input: $V = \{v_1, v_2, \dots, v_k\}$                ▷ Vertices of polytope $P_\gamma(n,k)$, $k \geq n$

V ← Sort(V)
$T \leftarrow \{\{v_1, v_2, \dots, v_n\}\}$
$B \leftarrow$ set of all facets of simplex $hull\{v_1, v_2, \dots, v_n\}$

for $i = 1$ to $k - n$ do

    $\mathcal{F} \leftarrow B$
    for $F_j \in \mathcal{F}$ do
        if Visible($F_j, v_{n+i}, T$)=False then
            $\mathcal{F} \leftarrow \mathcal{F} \setminus \{F_j\}$
        end if
    end for

    $S_j \leftarrow hull\{V(F_j) \cup \{v_i\}\}$    for all $F_j \in \mathcal{F}$    ▷ New simplices
    $T' \leftarrow \bigcup S_j$
    for $S_j \in T'$ do
        $\mathcal{F}(S_j) \leftarrow$ set of all facets of simplex $S_j$.
    end for
    $B' \leftarrow \{F \in \bigcup \mathcal{F}(S_j) \mid F \in \mathcal{F}(S_k)$ and $F \notin \mathcal{F}(S_m)\ \forall m \neq k\}$

    $T \leftarrow T \cup T'$
    $B \leftarrow (B \cup B') \setminus \mathcal{F}$
end for

Returns: $T$                ▷ Triangulation of polytope $P_\gamma(n,k)$
```

Then

$$X_i = \sum_{i=1}^{n} W_i \cdot v_i \tag{7.10}$$

is a uniform point in the simplex spanned by the vertices $v_1, \dots, v_n$.

Proof. The proof of this theorem depends on linear transformations between random variables. The following theorem on transformations [6] tells us how a (linear) transformation affects the density function.

Theorem 7.5. *Suppose that $X = (X_1, X_2, \dots, X_k)$ is a vector of continuous random variables with joint pdf $f_X(x_1, x_2, \dots, x_k) > 0$ on A, and $Y = (Y_1, Y_2, \dots, Y_k)$ is defined by the one-to-one transformation*

$$Y_i = u_i(X_1, X_2, \dots, X_k) \qquad i = 1, 2 \dots, k.$$

If the Jacobian is continuous and nonzero over the range of transformation, then the joint pdf of Y is

$$f_Y(y_1, y_2, \dots, y_k) = f_X(x_1, x_2, \dots, x_k)|J|,$$

where $x = (x_1, x_2, \dots, x_k)$ is the solution of $y = u(x)$.

Firstly, consider $(0, U_{(1)}, \ldots, U_{(n-1)}, 1)$ and let Ω_U be the space of all possible partitions of the interval $(0, 1)$ into n intervals. We use the proof by Feller [26, p. 74] to show that $(0, U_{(1)}, \ldots, U_{(n-1)}, 1)$ is uniformly distributed over Ω_U.

Let $U_{(1)}, U_{(2)}, \ldots, U_{(n-1)}$ be $n-1$ random points rearranged in increasing order. The points $U_{(1)}, U_{(2)}, \ldots, U_{(n-1)}$ divide the interval $[0, 1]$ into n subintervals. The sample space corresponding to $(U_1, U_2, \ldots, U_{n-1})$ is the $n-1$-dimensional hypercube Γ defined by $0 < u_k < 1$ and probabilities equal to the volume. The sample space with the $U_{(k)}$ as coordinate variables is the subset of Ω_U of Γ containing all points such that $0 < u_1 \leq u_2 \leq \ldots \leq u_{n-1} < 1$. Since the set Ω_U contains the single permutation of $u_1, u_2, \ldots, u_{n-1}$ such that they are in increasing order, the volume of Ω_U is $1/(n-1)!$.

The sample space Γ contains $(n-1)!$ congruent replicas of Ω_U, each one corresponds with a fixed permutation. These congruent spaces overlap if and only if $U_j = U_k$ for some $j \neq k$. The latter event is an zero-probability event. It follows that for any subset $A \subseteq \Omega_U$ the probability that $(U_{(1)}, U_{(2)}, \ldots, U_{(n-1)})$ lies in A equals the probability that $(U_1, U_2, \ldots, U_{n-1})$ lies in one of the $(n-1)!$ replicas of A, this probability in turn equals $(n-1)!$ times the volume of A. Thus $\mathbb{P}\{(U_{(1)}, U_{(2)}, \ldots, U_{(n-1)}) \in A\}$ equals the ratio of the volumes of A and Ω_U. Therefore, the tuple $(U_{(1)}, U_{(2)}, \ldots, U_{(n-1)})$ is distributed uniformly over the set Ω_U of points such that $0 < u_1 \leq u_2 \leq \ldots \leq u_{n-1} < 1$.

Let Ω_W denote the sample space of the vector $(W_1, \ldots, W_n)$ and note that (7.9) describes a one-to-one linear transformation L from Ω_U to Ω_W. The linear transformation L can be described by a matrix A. Note that A is a lower triangular matrix with all entries on the diagonal equal to 1 except for the entry in the lower right corner which is equal to -1. Hence, the absolute value of the determinant of the Jacobian of L is equal to one and we conclude that $(W_1, \ldots, W_n)$ is uniformly distributed over Ω_W.

The final step is to perform the linear transformation (7.10) to obtain a point X_i in the simplex S. Every point x in the simplex S has unique coefficients $w_1, w_2, \ldots, w_n$ such that $w_i \geq 0$,

$$x = \sum_{i=1}^{n} w_i \cdot v_i \quad \text{and} \quad \sum_{i=1}^{n} w_i = 1,$$

otherwise it would contradict with the affine independence of the points $v_1, v_2, \ldots, v_n$. The linear transformation from $(W_1, \ldots, W_n)$ to X_i is thus a one-to-one transformation. Moreover, the Jacobian of this linear transformation is 1. By Theorem 7.5, it follows that X_i is uniformly distributed over the simplex S because $(W_1, \ldots, W_n)$ is uniformly distributed. This completes the proof. □

Algorithm 3 Uniformly Sample Simplex

```
Input: $\{v_1, \ldots, v_n\}$                    ▷ $n$ affinely independent points

    Draw $n-1$ uniform samples $U_1, U_2, \ldots, U_{n-1}$
    Sort them in increasing order $U_{(1)}, U_{(2)}, \ldots, U_{(n-1)}$
    Compute differences
    $W_1, \ldots, W_n = U_{(1)}, U_{(2)} - U_{(1)}, \ldots, U_{(n-1)} - U_{(n-2)}, 1 - U_{(n-1)}$

Returns: $\sum_i W_i \cdot v_i$
```

7.3.4 Some additional notes

In the previous section we have introduced the beneath-and-beyond algorithm to find a triangulation of the convex polytope $P_\gamma(n)$. In this section we shall comment how to minimize the running time of the beneath-and-beyond algorithm. We will also implement Algorithm 1 to see how it improves the running time of the simulation program for the quadruple cell model.

Observe that the method to sample from (7.1) only requires basic operations to draw a sample from $U^{(n)}$, compute the vertices of the sample space and to draw a uniform sample from some simplex S. The running time is most affected by the computation of the volumes of the simplices in the triangulation and the computation of the of determinants in the beneath-and-beyond algorithm.

The beneath-and-beyond algorithm only requires knowledge about the sign of the determinant rather than the exact value of the determinant. In the literature much attention is devoted to the computation of the signs of determinants [10, 22]. These papers present methods which can be used to reduce the time to determine the sign of a determinant. However, the most crucial problem is due to Lemma 7.1. In the best case scenario the running time of the algorithm would be linear in the number of vertices. By Lemma 7.1, the number of vertices is a exponential function of n. Hence, even in the best case scenario the algorithm has an exponential running time.

Algorithm 2 can be naively implemented by running it separately for each snapshot. However, by Lemma 7.1, it is unnecessary to run the algorithm for each value of γ separately. The description of the vertices of the sample space $P_\gamma(n)$ only changes when $\lfloor \gamma \rfloor$ changes. Hence, for all $n \in \mathbb{N}$ it is only necessary to run the algorithm for some sequence $(\gamma_i)_{i=1}^n$ so that $i-1 < \gamma_i < i$. The triangulations of $P_\gamma(n)$ for the n distinct values γ_i can be saved in a file so that the file can be invoked when necessary.

7.4 Conclusions and recommendations

The algorithm presented in this chapter complements the results obtained in Chapter 6. The algorithm makes it possible to, with high probability, generate snapshots from any group of desire. Moreover, as we argue in Section 7.3.4, a clever implementation of the algorithms introduced in Section 7.3 is unlikely to cause a severe increase of the running time of the simulation program.

The algorithm depends on generating uniform samples from polytopes. We solved this issue by finding a triangulation of the polytope which reduces the problem to draw uniform samples from simplices. Drawing uniform samples from simplices is a standard exercise. The subject of uniformly drawing samples from polytopes has not yet attracted much attention in the literature. The solution which we present in this paper is, to our knowledge, not to be found in the literature. However, Dr. Pendavingh made us aware of the existence of so-called hit-and-run algorithms [23, 24, 47]. These algorithms attempt to generate an approximately random sample from a convex polytope by executing a random walk in the polytope. It is not yet clear whether random walks will decrease the running time to generate samples from the density function (7.1).

Future research would need to focus on methods to reduce the running time of the algorithm to generate snapshots from specific groups. The algorithm should be compared to the hit-and-run algorithm to decide which of these methods shows best performance. We also note that the algorithm crucially depends on the fact that the squared distance of a user is a uniformly distributed random variable which in turn is a consequence of the use of a homogeneous Poisson process. More research is required to investigate whether the algorithm can be extended to other distributions.

The sampling algorithm, which enables us to draw samples from any group of desire, has the benefit that it can be used to assess whether a change of the setting of the parameters will improve the performance of the network for a certain group. Suppose that a certain setting of the parameters improves the performance for group g. Then one would like to generate samples from other groups for group g and combine these with the already generated snapshots to have a representative sample of missed traffic for the network. This requires estimation of the prior probability to fall into the allocation regions, we illustrate this by an example.

Consider the allocations regions displayed in Table 6.12. We would need to compute the probability for X_3 to fall into each of the four regions, this leaves us with probabilities p_1, p_2, p_3 and p_4. We would generate a sample from the uniform distribution to appoint the allocation region from which a sample value has to be taken. Then we uniformly choose a snapshot which was assigned to this allocation region. We repeat previous steps until there are no more snapshots left for one of the groups.

Chapter 8

A pixel based snapshot

In the present model, users are modeled by a homogeneous spatial Poisson process Π, i.e. a spatial Poisson process with constant intensity λ. Consequently, users are homogeneously distributed over the network. A more realistic model is to divide the area which is covered by the network in smal squares called pixels. These pixels represent different environments, for example a pixel can represent 'street', 'highway' or 'meadow'. In this way the environment can be modeled, see [67]. On each of the pixels a homogeneous spatial Poisson process is defined. The combination of all these Poisson processes result in an inhomogeneous Poisson process Π with intensity function $\lambda(x)$.

The use of the inhomogeneous Poisson process would have a devastating effect on the tool which is developed Chapter 7. The tool explicitly depends on the fact that users which are connected to the basestation of interest are uniformly distributed. In this chapter we show how to extend the tool to generate snapshots with certain characteristics to a larger class of Poisson processes. The extension is based on a common used tool in Monte Carlo simulation: importance sampling. In Section 8.3 we give an introduction to importance sampling. In Section 8.2 we show how importance sampling can be used to extend the tool of Chapter 7 to a larger class of Poisson processes.

8.1 Importance sampling

Monte Carlo simulation is the use of experiments with random numbers to evaluate a mathematical expression. The simplest form, to which all instances of Monte Carlo simulation can be reduced, is the evaluation of an integral

$$\theta = \int f(x)dx.$$

By identifying a random variable X with density $p(x)$ and a function $g(x)$ it is possible to write θ as the expected value of $g(X)$

$$\mathbb{E}_p[g(X)] = \int g(x)p(x)dx = \int f(x)dx = \theta.$$

The problem of evaluating the integral becomes the familiar problem of estimating a mean. The natural estimate for a mean is the sample mean of a random sample of size n

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} g(x_i).$$

The sample mean is an unbiased estimator since

$$\mathbb{E}[\hat{\theta}(p)] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_p[g(X_i)] = \theta.$$

The main idea of importance sampling is to increase the design frequency of "important" events to make it possible to accurately estimate the frequency or expected values of those events using fewer Monte Carlo replications. However, originally importance sampling was used to reduce the variance of the desired quantity of interest. Obviously, these two objectives are closely related since by reducing the variance of the estimator it suffices to have fewer replications to obtain a certain level of accuracy for the estimator.

In what follows we will introduce importance sampling, for a mathematically more complete derivation see [32]. The idea of importance sampling is that multiplying and dividing by another density function $\widetilde{p}(x)$ does not change the value of θ since

$$\theta = \mathbb{E}_p[g(X)] = \int g(x)p(x)dx = \int g(x)\frac{p(x)}{\widetilde{p}(x)}\widetilde{p}(x)dx = \mathbb{E}_{\widetilde{p}}[g(\widetilde{X})\frac{p(\widetilde{X})}{\widetilde{p}(\widetilde{X})}], \tag{8.1}$$

provided that $\widetilde{p}(x) > 0$ if $p(x) > 0$. The random variable $\widetilde{X}$ obeys the so-called sampling distribution $\widetilde{p}(x)$. Equation (8.1) suggest the following scheme to estimate θ

i. Draw n samples $x_1, x_2, \ldots, x_n$ using the density $\widetilde{p}(x)$.

ii. Compute the weights w_i

$$w_i = \frac{p(x_i)}{\widetilde{p}(x_i)} \qquad \text{for } i = 1, 2, \ldots, n$$

(iii) Estimate θ by

$$\widetilde{\theta} = \frac{1}{n}\sum_{i=1}^{n} g(x_i)\cdot w_i. \tag{8.2}$$

The first observation is that $\widetilde{\theta}$ is an unbiased estimator since,

$$\mathbb{E}[\widetilde{\theta}] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\widetilde{p}}[g(\widetilde{X}_i)\cdot\frac{p(\widetilde{X}_i)}{\widetilde{p}(\widetilde{X}_i)}] = \theta.$$

One key question that remains is if there is a choice of $\widetilde{p}(x)$ which gives better results than the original density $p(x)$. Consider the variance of the estimator $\widetilde{\theta}$.

$$\begin{aligned} n^2 Var(\widetilde{\theta}) &= \int \left(g(x)\frac{p(x)}{\widetilde{p}(x)} - \theta\right)^2 \widetilde{p}(x)dx = \int \left(\frac{g(x)^2p(x)^2}{\widetilde{p}(x)} - 2g(x)p(x)\theta + \theta^2\widetilde{p}(x)\right)dx \\ &= \int \frac{g(x)^2p(x)^2}{\widetilde{p}(x)}dx - \theta^2 = I_{\widetilde{p}} - \theta^2. \end{aligned}$$

Next, try to choose $\widetilde{p}$ to minimize the above expression

$$I_{\widetilde{p}} = \int \frac{g(x)^2 p(x)^2}{\widetilde{p}(x)^2} \widetilde{p}(x) dx = \mathbb{E}[\frac{g(\widetilde{X})^2 p(\widetilde{X})^2}{\widetilde{p}(\widetilde{X})^2}].$$

Assuming that $g(x)$ is a non-negative function, the above expression can be lower bounded by Jensen's inequality.

$$I_{\widetilde{p}} \geq \left(\mathbb{E}[\frac{g(\widetilde{X}) p(\widetilde{X})}{\widetilde{p}(\widetilde{X})}] \right)^2 = \theta^2$$

Jensen's inequality holds for some random variable Z with equality if and only if Z is almost surely a constant [43, p. 46]. Hence, the optimal choice for $\widetilde{p}$ would be

$$\widetilde{p}_{opt} = \frac{g(x)p(x)}{\theta}.$$

Note that the variance of $\hat{\theta}(\widetilde{p})$ is zero for all k. Unfortunately, this is not a very useful result. In the first place, the optimal density explicitly depends on θ, the unknown quantity which is to be estimated. In fact, if θ were to be known, then there would be no need to run the simulation experiments. Secondly, the density $p(x)$ does not need to be known. It may be the case that it is possible to draw samples from it, but explicit expressions for it are generally not known. Hence, it is necessary to find other methods or criteria to choose a proper sampling distribution. More theory on the choice of $\widetilde{p}$ and examples of the use importance sampling to reduce variation of the estimator can be found in [11, 34].

8.2 Change from pixel based to uniform snapshot

Let C denote the circle of radius R centered around the basestation of interest. Furthermore, let $MT(\Gamma)$ denote the estimator of missed traffic when using Poisson process Γ, e.g. (3.9) or (3.11). In this section we first treat the general case when using a different Poisson process than the pixel based Poisson process. Then we focus on the specific case when users are modeled by a homogeneous Poisson process.

Suppose that the users in the wireless network model are described by a Poisson process Π with intensity function $\lambda : C \to \mathbb{R}$. The theory of importance sampling states that to eliminate the bias caused by the use of a different Poisson process $\widetilde{\Pi}$ with intensity function $\widetilde{\lambda} : C \to \mathbb{R}$, one should multiply the estimator of missed traffic by the fraction of the density function of Π over the density function of $\widetilde{\Pi}$. Observe that the use of $\widetilde{\Pi}$ brings two changes. Firstly, the distribution of the number of users in the network is changed but also the distribution of the positions of the users in the cell differ. Since the number and positions of points in a Poisson process are independent, one expects that the causes for bias are eliminated by two different multiplicators say M_N and M_D. The multiplicator M_N corrects for the changed number of users while M_D corrects the bias caused by the different distribution of the positions of the users.

With N, respectively $\widetilde{N}$, denote the number of users in the cell of interest when using Poisson process Π, respectively $\widetilde{\Pi}$. Similarly, X_j and $\widetilde{X}_j$ denote the position of the j^{th} user connected to the basestation of interest for both processes. Finally, $\Lambda(A) = \int_A \lambda(x)dx$ while $\widetilde{\Lambda}(A) = \int_A \widetilde{\lambda}(x)dx$. The number of users is Poissonian distributed with mean $\Lambda(C)$ for the process Π while the number of users is Poissonian distributed with mean $\widetilde{\Lambda}(C)$ for $\widetilde{\Pi}$. Hence, to eliminate the bias caused by the different distribution of the number of users, the estimator of missed traffic should be multiplied by

$$M_N = e^{\widetilde{\Lambda}(C)-\Lambda(C)} \left(\frac{\Lambda(C)}{\widetilde{\Lambda}(C)} \right)^{\widetilde{N}} .$$

The second multiplicator eliminates the bias caused by the different distribution for the positions of the users in the cell. The density for X_j is, by Theorem B.2, $\lambda(x)/\Lambda(C)$. Hence, the second multiplicator should be

$$M_D = \prod_{j=1}^{\widetilde{N}} \left(\frac{\lambda(\widetilde{X}_j)}{\widetilde{\lambda}(\widetilde{X}_j)} \cdot \frac{\widetilde{\Lambda}(C)}{\Lambda(C)} \right) = \left(\frac{\widetilde{\Lambda}(C)}{\Lambda(C)} \right)^{\widetilde{N}} \prod_{j=1}^{\widetilde{N}} \frac{\lambda(\widetilde{X}_j)}{\widetilde{\lambda}(\widetilde{X}_j)}$$

Therefore, a logical candidate for the multiplication factor M which eliminates the bias of the estimator of missed traffic would be

$$M = M_N \cdot M_D = e^{\widetilde{\Lambda}(C)-\Lambda(C)} \cdot \prod_{j=1}^{\widetilde{N}} \frac{\lambda(\widetilde{X}_j)}{\widetilde{\lambda}(\widetilde{X}_j)}. \tag{8.3}$$

The fact that M eliminates the bias of the estimator of missed traffic is stated in the following lemma. The proof of this lemma can be found in Appendix A.2.

Lemma 8.1. *Let Π be some (non)-homogeneous spatial Poisson process with intensity function $\lambda(x)$ as is $\widetilde{\Pi}$ described by intensity function $\widetilde{\lambda}(x)$ such that $\widetilde{\lambda}(x) > 0$ whenever $\lambda(x) > 0$ and both intensity functions are Lesbesgue integrable. If M is defined as* (8.3)*, then the estimator of missed traffic given by*

$$MT(\widetilde{\Pi}) \cdot M$$

is unbiased in the sense that

$$\mathbb{E}[MT(\widetilde{\Pi}) \cdot M] = \mathbb{E}[MT(\Pi)].$$

Now we return to our initial plan to use a homogeneous Poisson process $\widetilde{\Pi}$ with constant intensity $\widetilde{\lambda}$ instead of the inhomogeneous pixel based Poisson process Π. The inhomogeneous Poisson process splits the cell C into pixels $P_1, P_2, \ldots, P_L$. The estimator of missed traffic remains to be unbiased by multiplication with an appropriate factor as is stated in the following corollary. Recall that $\widetilde{N}$ is now defined to be the number of users connected to the basestation of interest when using Poisson process $\widetilde{\Pi}$. Similarly, $\widetilde{X}_j$ denotes the position of the j^{th} user in C when using $\widetilde{\Pi}$.

Corollary 8.1. *When $L < \infty$,*

$$MT \cdot M = MT \cdot e^{\widetilde{\lambda}|C| - \sum_\ell \lambda_\ell |P_\ell|} \cdot \prod_{j=1}^{\widetilde{N}} \frac{\sum_\ell \lambda_\ell \mathbb{1}\{\widetilde{X}_j \in P_\ell\}}{\widetilde{\lambda}}$$

is unbiased in the sense that

$$\mathbb{E}[MT(\widetilde{\Pi}) \cdot M] = \mathbb{E}[MT(\Pi)],$$

provided that $\widetilde{\lambda} > 0$.

Proof. The proof is by direct application of Lemma 8.1. □

We want to stress that, although the notation can be intimidating, the implementation of Corollary 8.1 into an existing simulation program only requires a small modification of the program. The users should be generated by sampling from the pixel based Poisson process Π with intensity function $\lambda(x)$ but suppose that the simulation program uses a process with constant intensity $\widetilde{\lambda}$. Then a snapshot is generated in which a total of $\widetilde{N}$ users are positioned at $\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_{\widetilde{N}}$ in the cell C of the basestation of interest. The simulation program takes the same steps as it would have done when using the process Π and computes an estimator of missed traffic based upon the snapshot. Only the final step of the program undergoes a minor modification. Instead of returning the computed estimator of missed traffic it is multiplied by M defined in Corollary 8.1.

8.3 Conclusions and recommendations

Importance sampling can be a powerful tool to extend the algorithm to inhomogeneous Poisson processes. Suppose users are modeled by an inhomogeneous Poisson process with intensity function $\lambda : C \to \mathbb{R}$, then Lemma 8.1 states that instead we can use a homogeneous Poisson process with constant intensity $\widetilde{\lambda}$. The bias caused by the use of the homogeneous Poisson process is removed by multiplying the estimates of missed traffic by a factor M.

One must be careful when applying importance sampling. Firstly, consider the values of missed traffic which are returned by the simulation program. The choice of $\widetilde{\lambda}$ causes that the individual estimates of missed traffic can no longer be interpreted. For example, multiplication by M can cause the estimate of missed traffic to exceed 1. A second problem of importance sampling is that it can cause an increase of the variance of the estimate so that it makes simulation impracticable. The increase/decrease of variance is discussed in Section .

The latter observation also implies the importance of the choice for the intensity $\widetilde{\lambda}$ of the homogeneous Poisson process. An intuitive choice would be to take

$$\widetilde{\lambda} = \frac{\int_C \lambda(x)dx}{\int_C 1dx}. \tag{8.4}$$

However, this choice enjoys no optimal properties. We distinct two possibilities to choose $\widetilde{\lambda}$, static importance sampling and dynamic importance sampling, respectively.

With static importance sampling we base the choice of the intensity $\widetilde{\lambda}$ of the homogeneous Poisson process on a training set. The training is used to choose $\widetilde{\lambda}$ so that the variance of the estimator of missed traffic of the training set is minimized. Adaptive importance sampling is initialized by setting $\widetilde{\lambda}$, for example choose $\widetilde{\lambda}$ as (8.4), and fixing some $m \in \mathbb{N}$. After each batch of m snapshots, the optimal value for $\widetilde{\lambda}$ is determined so that it minimizes the variance of the estimator.

The benefits and disadvantages of static and dynamic importance sampling are obvious. If the optimal choice for $\widetilde{\lambda}$ does not suffer from strong fluctuations or there is a wide range of $\widetilde{\lambda}$ for which importance sampling works similarly, then we prefer to use static importance sampling. Otherwise we would choose to use dynamic importance sampling. For more information on static and dynamic importance sampling, and for examples, we refer to [1, 40, 54, 62].

Chapter 9

Conclusions and recommendations

In this paper we have laid out the ground rules to construct a static model of the UMTS network. We studied two specific models in detail, the single cell and the quadruple cell model. For the single cell model we give a complete probabilistic analysis. For the quadruple cell model we present two novel means to reduce the running time of the simulation program. In this chapter we shall review the methods we used to analyze these models and give recommendations for future research. The conclusions and recommendations in this chapter are a summary of the conclusions and recommendations presented at the end of Chapter 4 to 8.

In Chapter 4 we study the single cell model. The advantage of the single cell model is that, conditionally on the number of users, the transmit power of the users are independent. This allows for a complete specification of the distribution, mean and variance of missed traffic when the blocked users are not removed from the network. If blocked users are removed from the network, then we use Lemma 4.1 to compute the expectation of missed traffic.

Future research for the single cell model should focus on finding the distribution of missed traffic when users are removed from the network. Furthermore, a distribution for the factor F, by which the MAI of a user is modeled, should be found. A first attempt to find the distribution of F is described in Section 5.3. Furthermore, more research on numerical integration methods is necessary for Lemma 4.1 to have practical value.

In Chapter 5 we study the quadruple cell model. The quadruple cell model is bound to be more accurate than the single cell model because it models more basestations. We find that the best fit for missed traffic is the normal distribution. In Section 5.3 we also introduce the adapted single cell model which uses factors F_1 and F_2 to model MAI. We then show that we can link the quadruple cell model to the (adapted) single cell model by fitting (a) distribution(s) to the factor(s) F (F_1 and F_2). The reduction to the adapted single cell model shows great potential since simulations do not show reason to doubt the fact that the distribution of missed traffic is the same as for the quadruple cell model. Moreover, simulations show that use of the adapted single cell model can reduce the running time by a factor 5.

For the quadruple cell model more research is needed on the modeling of the network. We propose to model seven basestations instead of four. This proposal is due to the fact that the quadruple cell model puts a distinct structure on the network which should be avoided. Analogue to the derivations in Chapter 4 we suggest to give a complete probabilistic analysis of the adapted single cell model. Furthermore, more research is needed to explain the difference between the number of users which are not blocked in the quadruple cell model and the single cell model.

In Chapter 6 we introduce the general concept of classifying data into G groups and study two classification rules, LDA and QDA, in detail when $G = 4$. Results show that snapshots can be best assigned to the groups based on LDA with as feature variable the sum of the squares of the distances of the users. In this case, roughly 60% of the observations are assigned to the correct group. Moreover, the results indicate that running time can be reduced by a factor 2 to 5. Further research is needed to find out whether sampling from the allocation region of group g provides a representative sample of snapshots belonging to group g. Furthermore, the performance of LDA can be even further improved when only the users which are not blocked are considered.

We only discussed two classification rules in detail. The fact that data can be classified by use of a single feature variable allows the use of more complex classification rules. More research is needed to find whether there exists a classification rule other than LDA, which has better performance. The results also indicate that more knowledge on the process of removing users from the network could improve performance of the classification rule.

In Chapter 7 we provide a sampling algorithm to generate samples from the distribution described by the density function (7.1). The method depends on uniformly drawing samples from a convex polytope. We used convex geometry to find a decomposition of the polytope into simplices and then draw uniform samples from the simplices.

We argue in Section 7.3.4 that it is unlikely that the running time of the simulation program for wireless network models will severely increase due to the use of the sampling algorithm. However, more research is needed to study the effect on the running time. Hit-and-run algorithms can also be used to uniformly sample from convex polytopes, these algorithms should be compared to the sampling algorithm to see which performs best. The sampling algorithm is restricted to homogeneous Poisson process, future research should focus on methods to extend the algorithm to work for other distributions.

In Chapter 8 we attempt to extend the sampling algorithm by use of importance sampling. Importance sampling allows the use of a inhomogeneous Poisson process to model users. A homogenous Poisson process with constant intensity $\widetilde{\lambda}$ is used to model the users connected to the basestation of interest. Then the bias caused by the use of an homogeneous Poisson process instead of an inhomogeneous Poisson process is removed by multiplying the estimates of missed traffic by an appropriate factor, see Lemma 8.1.

One should be careful with importance sampling because it can cause an increase in the variance of the estimator of missed traffic so that it makes simulation impracticable. More research is needed to find for which type of inhomogeneous Poisson processes importance sampling works well. Also more research is needed to find a proper setting for $\widetilde{\lambda}$ which minimizes the variance of the estimator of missed traffic.

The methods introduced in this paper significantly reduce the running time of the static simulation program. However, the results presented in Chapter 5 and 6 are only valid for a specific choice of the parameters of the model, see Appendix E. It should be validated if the findings still hold for different settings of the parameters. Ideally we would like to link the results of these chapters to the choice of parameters. We further encourage a study of the set of linear equations (3.4) and the process of removing blocked users from the network. More knowledge could help to explain the results of the reduction of the quadruple cell model to the (adapted) single cell model and could upgrade the performance of the sampling algorithm in Chapter 7. In this paper we studied the model of the 3G UMTS network while currently the HSDPA network is implemented. Therefore, the model needs to be updated so that it represents a HSDPA network.

References

[1] T. P. I. Ahamed, V. S. Borkar, and S. Juneja. Adaptive importance sampling technique for Markov chains using stochastic approximation. *Oper. Res.*, **54**:489–504, 2006.

[2] T. Anderson and D. Darling. A Test of Goodness of Fit. *J. American. Stat. Ass.*, **49**:765–769.

[3] T. W. Anderson. *An introduction to multivariate statistical analysis.* Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Inc., second edition, 1984.

[4] K. E. Atkinson. *An introduction to numerical analysis.* John Wiley & Sons, New York-Chichester-Brisbane, 1978.

[5] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Computational Geometry*, **7**:265–301, (1997).

[6] L. J. Bain and M. Engelhardt. *Introduction to probability and mathematical statistics.* Duxbury, 1992.

[7] J. O. Berger. *Statistical decision theory and Bayesian analysis.* Springer Series in Statistics. Springer-Verlag, second edition, 1985.

[8] R. P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *Comput. J.*, **14**:422–425, (1971).

[9] R. P. Brent. *Algorithms for minimization without derivatives.* Prentice-Hall Inc., 1973. Prentice-Hall Series in Automatic Computation.

[10] H. Brönnimann and M. Yvinec. Efficient exact evaluation of signs of determinants. *Algorithmica*, **27**:21–56, 2000.

[11] J. A. Bucklew. *Introduction to rare event simulation.* Springer, 2004.

[12] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey. *Graphical methods for data analysis.* Wadsworth, Duxbury Press, 1983.

[13] J. Cohen and T. Hickey. Two algorithms for determining volumes of convex polyhedra. *J. Assoc. Comput. Mach.*, **26**:401–414, (1979).

[14] A. C. Cohen, Jr. On estimating the mean and standard deviation of truncated normal distributions. *J. Amer. Statist. Assoc.*, 44:518–525, 1949.

[15] A. C. Cohen, Jr. Estimating the mean and variance of normal populations from singly truncated and doubly truncated samples. *Ann. Math. Statistics*, 21:557–569, 1950.

[16] A. C. Cohen, Jr. On the solution of estimating equations for truncated and censored samples from normal populations. *Biometrika*, **44**:225–236, 1957.

[17] A. C. Cohen, Jr. Simplified estimators for the normal distribution when samples are singly censored or truncated. *Technometrics*, **1**:217–237, 1959.

[18] A. C. Cohen, Jr. Tables for maximum likelihood estimates: singly truncated and singly censored samples. *Technometrics*, **3**:535–541, 1961.

[19] A. C. Cohen, Jr. and J. Woodward. Tables of Pearson-Lee-Fisher functions of singly truncated normal distributions. *Biometrics*, **9**:489–497, 1953.

[20] W. A. Coppel. *Foundations of convex geometry*, Vol. 12 *Australian Mathematical Society Lecture Series.* Cambridge University Press, Cambridge, 1998.

[21] H. Cramér. *Mathematical methods of statistics.* Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1999. Reprint of the 1946 original.

[22] T. Culver, J. Keyser, D. Manocha, and S. Krishnan. A hybrid approach for determinant signs of moderate-sized matrices. *Internat. J. Comput. Geom. Appl.*, **13**:399–417, 2003.

[23] M. Dyer and A. Frieze. Computing the volume of convex bodies: a case where randomness provably helps. In *Probabilistic combinatorics and its applications (San Francisco, CA, 1991)*, Vol. **44** *Proc. Sympos. Appl. Math.*, pages 123–169. Amer. Math. Soc., 1991.

[24] M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. Assoc. Comput. Mach.*, **38**:1–17, 1991.

[25] L. P. Fatti, D. M. Hawkins, and E. L. Raath. *Discriminant Analysis*, pages 1–71. Cambridge University Press, 1982.

[26] W. Feller. *An introduction to probability theory and its applications*, Vol. II *Second edition.* John Wiley & Sons Inc., New York, 1971.

[27] J. H. Friedman. Regularized discriminant analysis. *J. Amer. Statist. Assoc.*, **84**:165–175, 1989.

[28] G. Grimmett and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, Oxford, 2001.

[29] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity. II. Volume and mixed volumes. In *Polytopes: abstract, convex and computational (Scarborough, ON, 1993)*, Vol. **440** *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 373–466. Kluwer Acad. Publ., Dordrecht.

[30] P. M. Gruber and J. M. Wills. *Handbook of convex geometry*, Vol. A. North-Holland, Amsterdam, 1993.

[31] B. Grünbaum. *Convex polytopes*, Vol. 221 *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2003. Prepared and with a preface by Volker Kaibel, Victor Klee and Günter M. Ziegler.

[32] J. M. Hammersley and D. C. Handscomb. *Monte Carlo methods*. Wiley, New York, 1964.

[33] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, **29**:317–325, (1980).

[34] P. Heidelberger. Fast Simulation of Rare Events in Queueing and Reliability Models. In *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93*, pages 165–202, London, UK, 1993. Springer-Verlag.

[35] J. L. Hintze and R. D. Nelson. Violin Plots: A Box Plot-Density Trace Synergism. *The American Statistician*, pages 181–184, 1998.

[36] D. Hwang, W. A. Schmitt, G. Stephanopoulos, and G. Stephanopoulos. Determination of minimum sample size and discriminatory expression patterns in microarray data. *Bioinformatics*, **18**:1184–1193, 2002.

[37] M. James. *Classification Algorithms*. William Collins Sons & Co., 1985.

[38] M. Joswig. Beneath-and-Beyond Revisited, (2002). Available on `http://arxiv.org/abs/math.MG/0210133`.

[39] M. J. Klok. *Performance Analysis of Advanced Third Generation Receivers*. PhD thesis, Technische Universiteit Delft, 2002.

[40] H. J. Kushner and J. Yang. Analysis of Adaptive Step-Size SA Algorithms for Parameter Tracking. *IEEE Trans. Automatic Control*, **40**:1403–1410, 1995.

[41] P. Lachenbruch. *Discrimant Analysis*. Hafner Press, 1975.

[42] J. Laiho, A. Wacker, and T. Novosad. *Radio network planning and optimisation for UMTS*. Wiley, 2002.

[43] E. L. Lehmann and G. C. Casella. *Theory of point estimation*. Springer, New York, 1998.

[44] P. A. W. Lewis. Distribution of the Anderson-Darling statistic. *Ann. Math. Statist.*, **32**:1118–1124, 1961.

[45] J. C. Liberti Jr. and T. S. Rappaport. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*. Prentice Hall PTR, Upper Saddle River, 1999.

[46] J. K. Lindsey. *Parametric statistical inference*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1996.

[47] L. Lovász. Hit-and-run mixes fast. *Math. Program.*, **86**:443–461, 1999.

[48] G. G. Magaril-Il′yaev and V. M. Tikhomirov. *Convex analysis: theory and applications*, Vol. 222 *Translations of Mathematical Monographs.* American Mathematical Society, Providence, RI, 2003. Translated from the 2000 Russian edition by Dmitry Chibisov and revised by the authors.

[49] K. Mardia. Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies. *Sankhya.*

[50] G. J. McLachlan. *Discriminant analysis and statistical pattern recognition.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York, 1992.

[51] D. C. Montgomery and E. A. Peck. *Introduction to linear regression analysis.* Wiley-Interscience, Chichester, 1992.

[52] D. B. Owen. *Handbook of statistical tables.* Addison-Wesley Publishing Company, 1962.

[53] R. Prasad. *CDMA for Wireless Personal Communications.* Artech House, London, 1996.

[54] R. S. Randhawa and S. Juneja. Combining importance sampling and temporal difference control variates to simulate markov chains. *ACM Trans. Model. Comput. Simul.*, **14**:1–30, 2004.

[55] S. R. Saunders. *Antennas and Propagation for Wireless Communication Systems.* Wiley, New York, 1999.

[56] A. Schrijver. *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., 1986.

[57] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, **6**:423–434, (1991).

[58] D. J. Sheskin. *Handbook of parametric and nonparametric statistical procedures.* Boca Raton, 3rd edition, 2004.

[59] J. M. Steele. *Stochastic calculus and financial applications*, Vol. **45** *Applications of Mathematics (New York).* Springer-Verlag, 2001.

[60] M. Stephens. Use of the Kolmogorov-Smirnov, Cramer-Von Mises and Related Statistics Without Extensive Tables. *J. Royal Stat. Soc.*, **32**:115–122, (1970).

[61] H. Tverberg. How to cut a convex polytope into simplices. *Geometriae Dedicata*, **3**:239–240, (1974).

[62] F. J. Vazquez-Abad and I. Baltcheva. Intelligent Simulation for the Estimation of the Uplink Outage Probabilities in CDMA networks. Technical report, Les Cahiers du GERAD, December 2002.

[63] A. J. Viterbi. *CDMA: principles of spread spectrum communication.* Addison-Wesley Wireless Communication Series. Addison-Wesley Publishing Company, 1995.

[64] J. Voigt. A novel approach to processing speed enhancement for dynamic CDMA network simulations. In *Vehicular Technology Conference, 2004. VTC2004-Fall]. 2004 IEEE 60th, YEAR = 2004, volume =* **6***, pages = 4301–4305,*.

[65] J. Voigt, J. Deissner, J. Hubner, D. Hunold, and S. Mobius. Optimizing HSDPA performance in the UMTS network planning process. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, Vol. **4**, pages 2384–2388, 2005.

[66] P. W. Wahl and R. A. Kronmal. Discriminant Functions when Covariances are Unequal and Sample Sizes are Moderate. *Biometrics*, **33**:479–484, 1977.

[67] T. Winter, U. Türke, E. Lamers, R. Perera, A. Serrador, and L. Correia. Advanced simulation approach for integrated static and short-term dynamic UMTS performance evaluation. Technical Report D2.7, IST-2000-28088 MOMENTUM, 2003.

Appendix A

Mathematical derivations

In this chapter we shall provide the mathematical derivations which were omitted in the previous chapters.

A.1 Expression for decoded bit

In this section we derive that the statistic Z_{m1} defined by (2.3) equals

$$Z_{m1} = \sqrt{\frac{P_m}{2}}\, b_{m1} + \frac{1}{G} \sum_{\substack{k=0 \\ k \neq m}}^{N-1} b_{k1} \sum_{j=1}^{G} a_{kj} a_{mj} \sqrt{\frac{P_k}{2}} + \frac{1}{G} \sum_{j=1}^{G} a_{mj} \frac{N_j}{\sqrt{2T_c}},$$

where $(N_j)_{j=1}^{G}$ are independent and identically Gaussian distributed random variables with zero mean and variance 1.

The contribution of user m to Z_{m1} is given by

$$\begin{aligned} I_{m1} &= \frac{\sqrt{2P_m}}{T_b}\, b_{m1} \int_0^{T_b} a_m(t)^2 \cos^2(2\pi f_{ca} t) dt = \frac{\sqrt{2P_m}}{T_b}\, b_{m1} \int_0^{T_b} \cos^2(2\pi f_{ca} t) dt \\ &= \frac{\sqrt{2P_m}}{T_b}\, b_{m1} \left(\frac{T_b}{2} + \frac{\sin(2\pi f_{ca} T_b)}{8\pi f_{ca}} \right), \end{aligned}$$

and provided that the carrier frequency f_{ca} is relatively much larger than the reciprocal of the bit period

$$I_{m1} = \sqrt{\frac{P_m}{2}}\, b_{m1}. \tag{A.1}$$

In a similar way ζ is computed.

$$\zeta = \sum_{\substack{k=0 \\ k \neq m}}^{N-1} \frac{\sqrt{2P_0}}{T_b} \int_0^{T_b} b_k(t) a_k(t) a_m(t) \cos^2(2\pi f_{ca} t) dt$$

Recall that in the time to transmit a single bit we can transmit G chips.

$$\zeta = \frac{1}{G} \sum_{\substack{k=0 \\ k \neq m}}^{N-1} b_{k1} \sum_{j=1}^{G} a_{kj} a_{mj} \frac{\sqrt{2P_k}}{T_c} \int_{(j-1)T_c}^{jT_c} \cos^2(2\pi f_{ca} t) dt.$$

Provided that the carrier frequency f_{ca} is relatively much larger than the reciprocal of the chip period, it follows that

$$\zeta = \frac{1}{G} \sum_{\substack{k=0 \\ k \neq m}}^{N-1} b_{k1} \sum_{j=1}^{G} a_{kj} a_{mj} \sqrt{\frac{P_k}{2}}. \tag{A.2}$$

The last term to compute is the noise contribution

$$\eta = \frac{1}{T_b} \int_0^{T_b} a_m(t) \cos(2\pi f_{ca} t) dB(t) = \frac{1}{G} \sum_{j=1}^{G} \frac{a_{mj}}{T_c} \int_{(j-1)T_c}^{jT_c} \cos(2\pi f_{ca} t) dB(t).$$

It is a standard exercise in stochastic integration that the above integrals are mean zero Gaussian distributed with variance

$$\int_{iT_c}^{(i+1)T_c} \cos^2(2\pi f_{ca} t) dt,$$

for example see [59, p. 101]. By the familiar assumption that the carrier frequency f_{ca} is relatively much larger than the reciprocal of the chip period, it follows that the variance equals $T_c/2$. Therefore, the noise contribution becomes

$$\eta = \frac{1}{G} \sum_{j=1}^{G} a_{mj} \frac{N_j}{\sqrt{2T_c}}, \tag{A.3}$$

where $(N_j)_{j=1}^{G}$ are independent and identically Gaussian distributed random variables with zero mean and variance 1. Putting together (A.1) to (A.3) completes the proof.

A.2 Proof of Lemma 8.1

To prove Lemma 8.1, it suffices to show that the factor M corrects the errors which are made due to the use of $\widetilde{\Pi}$ instead of Π. Hence, it suffices to show that

$$\mathbb{E}[M \mathbb{1}\{\widetilde{N}(A) = 0\}] = e^{-\Lambda(A)}, \tag{A.4}$$

for all bounded A. since by Theorem B.1 the process is then characterized by a Poisson process with intensity function $\lambda(x)$.

Consider the Poisson process Π and let $A^c = C \setminus A$. By Rényi's theorem if follows

$$e^{-\Lambda(A)} = \mathbb{E}[\mathbb{1}\{N(A) = 0\}] = \mathbb{E}[\mathbb{1}\{N(A) = 0\}] \sum_{n=0}^{\infty} \mathbb{E}[\mathbb{1}\{N(A^c) = n\}].$$

We can also include the positions of the $N(A^c)$ users

$$\begin{aligned} e^{-\Lambda(A)} &= \mathbb{E}[\mathbb{1}\{N(A) = 0\}] \cdot \\ & \sum_{n=0}^{\infty} \left(\mathbb{E}[\mathbb{1}\{N(A^c) = n\}] \cdot \int_{A_c} \cdots \int_{A_c} \mathbb{E}[\mathbb{1}\{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\}] dx_1 \cdots dx_n \right). \end{aligned}$$

The random variables X_j are mutually independent, thus

$$
\begin{aligned}
e^{-\Lambda(A)} &= \mathbb{E}[\mathbb{1}\{N(A)=0\}]\cdot \\
&\sum_{n=0}^{\infty}\left(\mathbb{E}[\mathbb{1}\{N(A^c)=n\}]\cdot\int_{A_c}\cdots\int_{A_c}\prod_{i=1}^{n}\mathbb{E}[\mathbb{1}\{X_i=x_i\}]dx_1\cdots dx_n\right).
\end{aligned}
$$

The theory of importance sampling can be used to switch from the random variables belonging to Π to those of $\widetilde{\Pi}$. Carefully study each term in the following expression and recall that $N(A)\sim \mathrm{Poi}(\Lambda(A))$ while the density function of X_i restricted to a subset A is $\lambda(x)/\Lambda(A)$ as consequence of Theorem B.2.

$$
\begin{aligned}
e^{-\Lambda(A)} &= \mathbb{E}[\mathbb{1}\{\widetilde{N}(A)=0\}]\; e^{\widetilde{\Lambda}(A)-\Lambda(A)}\cdot \\
&\sum_{n=0}^{\infty}\left(\mathbb{E}[\mathbb{1}\{\widetilde{N}(A^c)=n\}]\; e^{\widetilde{\Lambda}(A^c)-\Lambda(A^c)}\left(\frac{\Lambda(A^c)}{\widetilde{\Lambda}(A^c)}\right)^n\cdot\right. \\
&\left.\left(\frac{\widetilde{\Lambda}(A^c)}{\Lambda(A^c)}\right)^n\int_{A_c}\cdots\int_{A_c}\prod_{i=1}^{n}\mathbb{E}[\mathbb{1}\{\widetilde{X}_i=x_i\}\frac{\lambda(\widetilde{X}_i)}{\widetilde{\lambda}(\widetilde{X}_i)}]\;dx_1\cdots dx_n\right)
\end{aligned}
$$

Terms cancel or complete each other and all that remains is

$$
\begin{aligned}
e^{-\Lambda(A)} &= e^{\widetilde{\Lambda}(C)-\Lambda(C)}\mathbb{E}[\mathbb{1}\{\widetilde{N}(A)=0\}]\cdot \\
&\sum_{n=0}^{\infty}\left(\mathbb{E}[\mathbb{1}\{\widetilde{N}(A^c)=n\}]\cdot\int_{A_c}\cdots\int_{A_c}\prod_{i=1}^{n}\mathbb{E}[\mathbb{1}\{\widetilde{X}_i=x_i\}]\frac{\lambda(x_i)}{\widetilde{\lambda}(x_i)}\quad dx_1\cdots dx_n\right).
\end{aligned}
$$

In the first steps of the proof we used the formulas for expectation and obtained an expression with sums and integrals, now follow the reverse path.

$$
\begin{aligned}
e^{-\Lambda(A)} &= e^{\widetilde{\Lambda}(C)-\Lambda(C)}\mathbb{E}[\mathbb{1}\{\widetilde{N}(A)=0\}]\cdot \\
&\sum_{n=0}^{\infty}\mathbb{E}[\mathbb{1}\{\widetilde{N}(A^c)=n\}]\;\mathbb{E}[\prod_{i=1}^{\widetilde{N}(A^c)}\frac{\lambda(\widetilde{X}_i)}{\widetilde{\lambda}(\widetilde{X}_i)}\mid \widetilde{N}(A^c)=n] \\
&= e^{\widetilde{\Lambda}(C)-\Lambda(C)}\mathbb{E}[\mathbb{1}\{\widetilde{N}(A)=0\}]\cdot\mathbb{E}[\prod_{i=1}^{\widetilde{N}(A^c)}\frac{\lambda(\widetilde{X}_i)}{\widetilde{\lambda}(\widetilde{X}_i)}]
\end{aligned}
$$

Since $\widetilde{\Pi}$ is a Poisson process it follows that the random variables $\widetilde{N}(A)$, $\widetilde{N}(A^c)$ and $\widetilde{X}_i,\ i=1,2,\ldots,\widetilde{N}(A^c)$ are mutually independent. Hence,

$$
e^{-\Lambda(A)} = e^{\widetilde{\Lambda}(C)-\Lambda(C)}\mathbb{E}[\mathbb{1}\{\widetilde{N}(A)=0\}\cdot\prod_{i=1}^{\widetilde{N}(A^c)}\frac{\lambda(\widetilde{X}_i)}{\widetilde{\lambda}(\widetilde{X}_i)}],
$$

this completes the proof since the last equation precisely is (A.4).

Appendix B

Mathematical Theory

B.1 Spatial Poisson processes in $\mathbb{R}^d$

In this section we shall give a short introduction on spatial Poisson processes which are used to model the users in the network. This overview of Poisson processes is a recap of the discussion in [28].

A spatial Poisson process Π in $\mathbb{R}^d$ is described by a spatial dependent intensity function $\lambda : \mathbb{R}^d \to \mathbb{R}$. The process Π is called homogeneous for $A \subset \mathbb{R}^d$ if $\lambda(x) = \lambda$ for all $x \in \mathbb{R}^d$ and $\lambda(x) = 0$ for all $x \in \mathbb{R}^d \setminus A$. In all other cases Π is called inhomogeneous. A random variable N is Poissonian distributed with mean λ if

$$\mathbb{P}(N = n) = e^{-\lambda}\frac{\lambda^n}{n!}, \tag{B.1}$$

which is denoted by $X \sim \text{Poi}(\lambda)$. The intensity of a bounded set $A \subset \mathbb{R}^d$ is defined to be

$$\Lambda(A) = \int_A \lambda(x)dx.$$

Definition B.1. *Let $d \geq 1$ and let $\lambda : \mathbb{R}^n \to \mathbb{R}$ be a non-negative measurable function such that $\Lambda(A) < \infty$ for all bounded A. The random countable subset Π of $\mathbb{R}^d$ is called Poisson process with intensity function λ if, for all $A \in \mathcal{B}^d$, the Borel σ-field of $\mathbb{R}^d$, the random variables $N(A) = |\Pi \cap A|$ satisfy*

i. $N(A)$ has the Poisson distribution with parameter $\Lambda(A)$, and

ii. if $A_1, A_2, \ldots, A_n$ are disjoint sets in $\mathcal{B}^d$, then $N(A_1), N(A_2), \ldots, N(A_n)$ are independent random variables.

Rényi found another useful characterization of a spatial Poisson process.

Theorem B.1 (Rényi's theorem)**.** *Let Π be a random countable subset of $\mathbb{R}^d$, and let $\lambda : \mathbb{R}^d \to \mathbb{R}$ be a non-negative Lebesgue integrable function satisfying $\Lambda(A) < \infty$ for all bounded A. If*

$$\mathbb{P}(\Pi \cap A = \emptyset) = e^{-\Lambda(A)}$$

for any finite union A of boxes, then Π is a Poisson process with intensity function λ.

An important and useful property of spatial Poisson processes is the conditional property.

Theorem B.2 (Conditional property)**.** *Let* Π *be a (non-)homogeneous Poisson process on* $\mathbb{R}^d$ *with intensity function* λ, $\lambda : \mathbb{R}^d \to \mathbb{R}$. *Let* A *be a subset of* $\mathbb{R}^d$ *such that* $0 < \Lambda(A) < \infty$. *Conditional on the event that* $|\Pi \cap A| = n$, *the* n *points of the process lying in* A *have the same distribution as* n *points chosen independently at random in* A *according to the common probability measure*

$$\mathbb{Q}(B) = \frac{\Lambda(B)}{\Lambda(A)}, \qquad \textit{for all } B \subseteq A.$$

Note that the conditional property implies that if $\lambda(x)$ is constant on some subset A of $\mathbb{R}^d$, the points of the process Π are uniformly distributed over the set A. Furthermore, the relevant density function for a point x in A is $\lambda x/\Lambda(A)$.

B.2 Test of hypotheses

The term hypothesis testing refers to the process of trying to decide the truth or falsity of hypotheses on the basis of experimental evidence. A hypothesis test consists of two hypotheses, the null hypothesis H_0 and the alternative hypothesis H_a. Based on a test statistic, a test is conducted whether we should reject H_0 in favor of H_a or that we should not reject H_a.

In this section we will first give a short introduction into the theory of hypothesis testing. After the introduction we shall present some goodness of fit tests. There are two types of goodness of fit tests, the first type is the one sample goodness of fit test. The one sample tests are used to test whether experimental data comes from some fully prescribed distribution function. The second type is the two sample goodness of fit test, these tests can be used to test whether two samples obey the same unknown distribution function.

B.2.1 Theory of hypothesis testing

In this section we will shortly discuss hypothesis testing. The material presented in this section is an overview of an extensive treatment of the subject to be found in [6].

In general, experimental measurements are subject to random errors, and thus any decision about the truth or falsity of the hypothesis, based on experimental evidence, is also subject to error. It is impossible to avoid an occasional decision error, but it is possible to construct tests that such errors occur infrequently and at some prescribed rate.

Definition B.2. *If* $X \sim f(x;\theta)$, *then a* statistical hypothesis *is a statement about the distribution of* X. *If the hypothesis completely specifies* $f(x;\theta)$, *then it is referred to as a* simple *hypothesis; otherwise it is called* composite.

Quite often the distribution in question has a known parametric form with a single unknown parameter θ, and the hypothesis consists of a statement about θ. For example, if $X \sim \mathcal{N}(\mu, \sigma)$ with σ known, hypothesis testing can be used to test whether μ is greater than μ_0. The *null hypothesis* H_0 is set to be $\mu < \mu_0$ and the *alternative hypothesis* is set to be $\mu > \mu_0$ or vice versa.

We must decide on the basis of the sample data whether we have sufficient statistical evidence to reject H_0 in favor of H_a, or whether we do not have sufficient evidence. The philosophy will be to divide the sample space into two regions, the *critical* or *rejection* region C, and the *nonrejection* region $S \setminus C$. If the observed sample data falls into the critical region C we reject H_0 in favor of H_a otherwise we do not reject H_0.

There are two possible errors in the procedure of hypothesis testing. These errors are

i. Type I error: reject a true H_0,

ii. Type II error: fail to reject a false H_0.

We hope to choose a test statistic and a critical region so that we would have a small probability of making these two errors. These errors are usually denoted as

i. $\mathbb{P}(\text{Type I error}) = \alpha$,

ii. $\mathbb{P}(\text{Type II error}) = \beta$.

The probability of rejecting a true H_0 is referred to as the significance level of the test.

The standard approach for testing is to specify or select some acceptable level of error such as $\alpha = 0.05$ or $\alpha = 0.01$ for the significance level of the test, and then to determine a critical region that will achieve this α. Among all critical regions of size α select the one that has the smallest type II error.

If the hypothesis is set up as described above, it is important to stress the difference between rejecting H_0 and not rejecting H_0. If H_0 is rejected and H_a is adopted, then the error rate α of being wrong is controlled. However, if H_0 is not rejected, the H_0 may be true or a type II error is made which has an uncontrolled error rate β. Most experiments have some goal or research hypothesis that one hopes to support with statistical evidence, it is clear that, if possible, this hypothesis should be taken as the alternative hypothesis.

There is not always general agreement about how small α should be for rejection of H_0 to constitute strong evidence in support of H_a. While one experimenter considers $\alpha = 0.05$ to be sufficient small, another experimenter may insist on using $\alpha = 0.01$. Hence, it would be possible for both experimenters to draw different conclusions based on the same data. If both experimenters agree to use the same test statistic, then this problem may be overcome by reporting the results of the tests in terms of the *p-value* of the test. The p-value is defined to be the smallest α at which H_0 can be rejected, based on the observed value of the test statistic. Therefore, H_0 is reject for $\alpha \geq p$ while it is not rejected for $\alpha < p$.

B.2.2 Goodness of fit tests

Suppose that $x_1 \leq x_2 \leq \ldots \leq x_n$ is an ordered sample of n independent observations from a distribution with the distribution function $F(x)$. When studying the proposed test statistics, bear in mind that it is assumed that sample values are ordered. One sample goodness of fit tests are used to test whether the experimental data obeys some prescribed distribution function $F_0(x)$. The hypotheses for a goodness of fit test are

$$\begin{aligned} H_0 &: \quad F(x) = F_0(x) \\ H_a &: \quad F(x) \neq F_0(x). \end{aligned}$$

We will treat three distinct goodness of fit tests, the Kolmogorov-Smirnov test, the Cramér-von-Mises test and the Anderson-Darling test.

The test statistics for all three the test are based on comparing the empirical distribution function $F_n(x)$ to the distribution function $F_0(x)$. The empirical distribution function $F_n(x)$ is defined as

$$F_n(x) = \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}\{x_i \leq x\}.$$

It should be clear that if the sample $x_1, x_2, \ldots, x_n$ is a sample from the distribution function $F_0(x)$, then the empirical distribution function should be close to F_0. Each of the tests has a different way to measure whether $F_n(x)$ is "close enough" to $F_0(x)$.

In [60], approximation formulae are given which can be used to compute p-values for each of the three tests. These approximation formulae are useful to generate p-values for data sets which are large enough such that the error is negligible small. Stephens [60] also suggests that the approximation formulae are already useful for data sets with as few as 8 sample values.

Kolmogorov-Smirnov test The Kolmogorov-Smirnov test, often called the K-S test, is based on the test statistic

$$D_n = \sup_{-\infty \leq x \leq \infty} |F_n(x) - F_0(x)| = \max_{1 \leq i \leq n} \left(F_0(x_i) - \frac{i-1}{n}, \frac{i}{n} - F_0(x_i) \right).$$

This particular test is known as the two-sided test since it tests both the hypotheses $F_n(x) \geq F_0(x)$ and $F_n(x) \leq F_0(x)$. The test is called on-sided if only one of these hypotheses is tested.

The null hypothesis is rejected in favor of the alternative hypothesis if D_n is greater than the critical value. Critical values for different values of α and for $1 \leq n \leq 100$ are tabulated in [52]. For data sets which contain more than 100 sample values we use the approximation formula

$$\mathbb{P}(D^* > z) = 2e^{-2z^2},$$

where D^*, the modified version of D_n, is

$$D^* = D_n(\sqrt{n} + 0.12 + 0.11/\sqrt{n}).$$

Cramér-von-Mises test Another test which can be used to test goodness of fit is the Cramér-von-Mises test or the C-M test. The test statistic is

$$W^2 = n \int_{-\infty}^{\infty} \left(F_0(x) - F_n(x)\right)^2 dF_0(x).$$

If $F_0(x)$ is differentiable, then the test statistic becomes

$$W^2 = \sum_{i=1}^{n} \left(F(x_i) - \frac{2i-1}{2n} \right)^2 + \frac{1}{12n} .$$

If the the data set is large enough we use the approximation formula

$$\mathbb{P}(W^* > z) = 0.05\ e^{2.79-6z},$$

where W^*, the modified version of W^2, is

$$W^* = (W_n^2 - 0.4/n + 0.6/n^2)(1.0 + 1.0/n).$$

Anderson-Darling test The A-D test, due to Anderson and Darling [2], uses the test statistic

$$A_n^2 = n\int_{-\infty}^{\infty} \frac{(F_n(x) - F_0(x))^2}{F_0(x)(1 - F_0(x))} dF_0(x) = -n - \sum_{i=1}^{n} \frac{2i-1}{n} (\log F_0(x_i) + \log(1 - F_0(x_{n+1-i})).$$

The Anderson-Darling test makes use of the specific distribution in calculating critical values. This has the advantage of allowing a more sensitive test and the disadvantage that critical values must be calculated for each distribution. However, the test statistic A^2 converges so rapidly that no modification is required for any $n \geq 5$. Anderson and Darling [2] found the following expression for the cdf of the test statistic A^2

$$\mathbb{P}(A^2 \leq z) = \sum_{j=0}^{\infty} \frac{(-1)^j e^{-\frac{(4j+1)^2\pi^2}{8z}} (4j+1)\Gamma\left(j + \frac{1}{2}\right)}{j!} \int_0^{\infty} e^{\frac{z}{8(y^2+1)} - \frac{(4j+1)^2\pi^2 y^2}{8z}} dy. \quad \text{(B.2)}$$

By Monte-Carlo studies Lewis [44] showed that the first two terms of the sum on the right-hand side of (B.2) obtain estimates with a precision of five significant numbers. The integral in (B.2) can only be evaluated numerically. However, by using a numerical integration formula with error control the estimate will still remain to be precise with five significant numbers.

Two-sample Kolmogorov-Smirnov test The two sample Kolmogorov-Smirnov test tries to determine if two samples differ significantly. The techniques used are similar to the one sample Kolmogorov-Smirnov test. The two sample Kolmogorov-Smirnov test compares the empirical distribution functions for both samples. The test statistic is the maximal distance between the two empirical distribution functions. This test is more complicated than the one sample goodness of fit tests but is standardly delivered with most statistical software packages. For example `Statgraphics` can be used to find asymptotic p-values for the test.

Appendix C

Maximum Likelihood Estimators

In this section we first give a short introduction into maximum likelihood estimators. We then compute the maximum likelihood estimators for exponential alike distributions and for the truncated normal distribution.

The method of maximum likelihood is based on the idea to use a parameter value in the parameter space that corresponds to the largest "likelihood" for the observed data as an estimate of an unknown parameter. Let $X_1, \ldots, X_n$ represent a random sample from $f(x; \theta)$, then

$$L(\theta) = \prod_{i=1}^{n} f(X_i; \theta),$$

is referred to as the likelihood function. The maximum likelihood estimator (MLE) $\hat{\theta}$ of θ satisfies

$$L(\hat{\theta}) = \max_{\theta \in \Omega} L(\theta),$$

where Ω is the set of all allowable values for θ.

If Ω is an open interval, and if $L(\theta)$ is differentiable and assumes a maximum on Ω, then the MLE will be a solution of the maximum likelihood equation

$$\frac{d}{d\theta} L(\theta) = 0.$$

For computational convenience the alternate form of the maximum likelihood equation

$$\frac{d}{d\theta} \log L(\theta) = 0$$

often will be used.

C.1 Exponential alike distributions

First consider the case when the $(X_i)_{i=1}^n$ are i.i.d. exponential distributed with scale parameter λ. The likelihood function is

$$\log\left(\lambda^n \prod_{i=1}^n e^{-\lambda X_i}\right) = n\log\lambda - \lambda\sum_{i=1}^n X_i.$$

Taking the derivative with respect to λ gives

$$\frac{n}{\lambda} - \sum_{i=1}^n X_i = 0.$$

The MLE of λ is

$$\hat{\lambda} = n\left(\sum_{i=1}^n X_i\right)^{-1}.$$

Let $(X_i)_{i=1}^n$ are i.i.d. two-parameter exponential distributed with scale parameter λ and location parameter γ. We need to maximize

$$\log\left(\lambda^n \prod_{i=1}^n e^{-\lambda(X_i-\gamma)}\right) = n\log\lambda - \lambda\sum_{i=1}^n (X_i-\gamma).$$

Note that the likelihood function is monotone increasing in γ. In order to maximize the likelihood we take γ as large as possible under the constraint that $\gamma \leq X_i$ for all $i = 1, \ldots, n$. Therefore, we choose $\hat{\gamma} = X_{(1)}$, the minimum of $X_1, \ldots, X_n$. We then proceed in the same way as for the exponential distribution and derive that

$$\hat{\lambda} = n\left(\sum_{i=1}^n (X_i - \hat{\gamma})\right)^{-1}.$$

C.2 Truncated normal distribution

In this section we describe the maximum likelihood estimators of the parameters for the truncated normal distribution. We distinct a few possibilities, namely a normal random variable can be left truncated, right truncated and double truncated.

Recall that the density function of a normally distributed random variable is

$$\varphi(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right],$$

while the cumulative distribution function is

$$\Phi(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x}\exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right].$$

A random X variable is double truncated normally distributed if its density function is

$$f_X(x;\mu,\sigma,x_L,x_R) = \begin{cases} 0 & \text{if } x < x_L \\ (\Phi(x_R;\mu,\sigma) - \Phi(x_L;\mu,\sigma))^{-1}\varphi(x;\mu,\sigma) & \text{if } x_L \leq x \leq x_R \\ 0 & \text{if } x > x_R \end{cases}$$

The left and right truncated normal density functions are defined similarly by replacing the denominator by $\Phi(x_L;\mu,\sigma)$, $1-\Phi(x_L;\mu,\sigma)$ respectively.

The maximum likelihood estimators of the truncation points x_L and x_R are quite simple. Let $X_1, X_2, \dots, X_n$ denote a sample from the double truncated normal distribution and $X_{(1)}, X_{(2)}, \dots, X_{(n)}$ the sample rearranged in increasing order. In order to maximize the likelihood function we need to minimize the denominator $\Phi(x_R;\mu,\sigma)-\Phi(x_L;\mu,\sigma)$ subjected to the condition that $x_L \le X_i \le x_R$. Since Φ is strictly increasing between x_L and x_R, it follows that the likelihood function is maximized by setting $\hat{x}_L = X_{(1)}$ and $\hat{x}_R = X_{(n)}$. For the left (right) truncated normal distribution the maximum likelihood estimator for x_L (x_R) is $X_{(1)}$ ($X_{(n)}$).

If the truncation points are known, or estimated when necessary, the parameters μ and σ can be estimated. Cohen devoted a series of papers [14, 15, 16, 17, 18, 19] to the subject of the maximum likelihood estimators for the parameters μ and σ. In these papers the following notation is introduced. Let $Z_L = Z_L(u_L, u_R)$ and $Z_R = Z_R(u_L, u_R)$ be defined as

$$\begin{aligned}
Z_L &= \varphi(u_L;0,1)/(\Phi(u_R;0,1)-\Phi(u_L;0,1)) \\
Z_R &= \varphi(u_R;0,1)/(\Phi(u_R;0,1)-\Phi(u_L;0,1))
\end{aligned}$$

Furthermore, let $\overline{X}$ and S represent the sample mean and standard deviation.

In case of a double truncated normal distribution, Cohen [16] derived the following system of equations for u_L and u_R

$$\begin{aligned}
\frac{Z_R - Z_L + u_R}{u_R - u_L} &= \frac{x_R - \overline{X}}{x_R - x_L} \ , \\
\frac{1 - Z_R u_R + Z_L u_L - (Z_R - Z_L)^2}{(u_R - u_L)^2} &= \frac{S^2}{(x_R - x_L)^2} \ .
\end{aligned}$$

The Newton-Raphson method might be used to find a solution to the latter equations. The likelihood estimators are then

$$\begin{aligned}
\hat{\mu} &= x_R - \frac{u_R(x_R - x_L)}{u_R - u_L}, \\
\hat{\sigma} &= \frac{u_R(x_R - x_L)}{u_R - u_L} \ .
\end{aligned}$$

Because of the symmetry of the normal distribution a right truncated normal random variable does not have to be considered separately and we focus on the left truncated normal distribution. Cohen [17] introduces the functions $Z = Z(\xi)$ and $\varphi = \varphi(\xi)$

$$\begin{aligned}
Z(\xi) &= \frac{\varphi(\xi;0,1)}{1 - \Phi(\xi;0,1)} \ , \\
\theta(\xi) &= \frac{Z(\xi)}{Z(\xi) - \xi} \ .
\end{aligned}$$

The equation of ξ which must be solved in order to estimate μ and σ is

$$\frac{1 - Z(Z-\xi)}{(Z-\xi)^2} = \frac{S^2}{(\overline{X} - x_L)^2} \ .$$

A root-finding algorithm needs to be used to find the solution $\hat{\xi}$ and the desired maximum likelihood estimators become

$$\begin{aligned}\hat{\sigma}^2 &= S^2 + \theta(\hat{\xi})(\overline{X} - x_L)^2,\\ \hat{\mu} &= \overline{X} - \theta(\hat{\xi})(\overline{X} - x_L).\end{aligned}$$

Appendix D

Results discriminant analysis

In this chapter we tabulate the results of linear and quadratic discriminant analysis. Recall that the following feature variables are used

$X_1 = N_1$, the number of users which are connected to basestation 1

$X_2 = \sum_j D_j/N_1$, the mean distance,

$X_3 = \sum_j D_j^2/N_1$, the mean squared distance.

Furthermore, M_1 is defined to the percentage of observations that were assigned to the correct group. The measure M_2 represents the percentage of the observations that were not assigned to the correct group but to a direct neighbor of the correct group. The third measure M_3 is defined to be the sum of M_1 and M_2.

Tables D.1 to D.4 show the groupwise results of quadratic discriminant analysis for every possible set of feature variables.

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	0.0	0.0	0.0
0	1	0	67.5	27.7	95.2
0	0	1	69.9	28.9	98.8
1	1	0	58.4	35.6	94
1	0	1	62.7	28.9	91.6
0	1	1	80.1	13.3	93.4
1	1	1	2.4	5.3	7.8

Table D.1: Performance QDA for group 1

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	0.0	0.0	0.0
0	1	0	61.5	38.2	99.7
0	0	1	63.1	36.1	99.2
1	1	0	67.7	42.3	100.0
1	0	1	70.6	28.9	99.5
0	1	1	57.4	42.3	99.7
1	1	1	0.0	0.0	0.0

Table D.2: Performance QDA for group 2

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	100.0	0.0	100.0
0	1	0	64.7	35.3	100.0
0	0	1	64.7	35.3	100.0
1	1	0	72.9	27.1	100.0
1	0	1	67.8	32.2	100.0
0	1	1	60.6	37.1	97.8
1	1	1	0.0	0.0	0.0

Table D.3: Performance QDA for group 3

Model			Performance		
X_1	X_2	X_3	M_1	M_2	M_3
1	0	0	0.0	0.0	0.0
0	1	0	67.1	31.5	98.6
0	0	1	71.9	28.1	100.0
1	1	0	61.0	37.6	98.6
1	0	1	67.1	32.9	100.0
0	1	1	57.5	41.1	98.6
1	1	1	100.0	0.0	100.0

Table D.4: Performance QDA for group 4

Tables D.5 and D.6 show the groupwise results for LDA, the first table shows the performance when we use the discriminant functions based on the subset of all snapshots for which $X_1 = 4$. The second table shows the performance when the discriminant functions are based on the entire training set.

Group 1		
Model	$M_1(1)$	$M_2(1)$
X_2	47.4	15.8
X_3	47.4	15.8
(X_2, X_3)	52.6	15.8

Group 2		
Model	$M_1(2)$	$M_2(2)$
X_2	60.0	0.0
X_3	56.4	0.0
(X_2, X_3)	58.2	0.0

Group 3		
Model	$M_1(3)$	$M_2(3)$
X_2	80.9	0.0
X_3	74.5	0.0
(X_2, X_3)	70.2	0.0

Group 4		
Model	$M_1(4)$	$M_2(4)$
X_2	19.0	0.0
X_3	23.8	0.0
(X_2, X_3)	28.6	0

Table D.5: LDA when $X_1 = 4$, discriminant functions based on snapshots with $X_1 = 4$

Group 1		
Model	$M_1(1)$	$M_2(1)$
X_2	57.9	10.5
X_3	52.6	15.8
(X_2, X_3)	57.9	15.8

Group 2		
Model	$M_1(1)$	$M_2(1)$
X_2	61.8	0.0
X_3	61.8	0.0
(X_2, X_3)	58.2	0.0

Group 3		
Model	$M_1(1)$	$M_2(1)$
X_2	72.3	0.0
X_3	68.1	0.0
(X_2, X_3)	68.1	0.0

Group 4		
Model	$M_1(1)$	$M_2(1)$
X_2	23.8	0.0
X_3	23.8	0.0
(X_2, X_3)	33.3	0.0

Table D.6: LDA when $X_1 = 4$, discriminant functions based on entire training set

Tables D.7 and D.8 show the groupwise results for QDA, the first table shows the performance when we use the discriminant functions based on the subset of all snapshots for which $X_1 = 4$. The second table shows the performance when the discriminant functions are based on the entire training set.

Group 1		
Model	$M_1(1)$	$M_2(1)$
X_2	36.8	15.8
X_3	42.1	15.8
(X_2, X_3)	31.6	15.8

Group 2		
Model	$M_1(2)$	$M_2(2)$
X_2	65.5	0.0
X_3	58.2	0.0
(X_2, X_3)	61.8	0.0

Group 3		
Model	$M_1(3)$	$M_2(3)$
X_2	74.5	0.0
X_3	78.7	0.0
(X_2, X_3)	63.8	0.0

Group 4		
Model	$M_1(4)$	$M_2(4)$
X_2	4.8	0.0
X_3	19.0	0.0
(X_2, X_3)	4.8	0.0

Table D.7: QDA when $X_1 = 4$, discriminant functions based on snapshots with $X_1 = 4$

Group 1		
Model	$M_1(1)$	$M_2(1)$
X_2	63.2	5.3
X_3	63.2	10.5
(X_2, X_3)	73.7	10.5

Group 2		
Model	$M_1(1)$	$M_2(1)$
X_2	65.5	0.0
X_3	61.8	0.0
(X_2, X_3)	65.5	0.0

Group 3		
Model	$M_1(1)$	$M_2(1)$
X_2	68.1	0.0
X_3	59.6	0.0
(X_2, X_3)	48.9	2.1

Group 4		
Model	$M_1(1)$	$M_2(1)$
X_2	38.1	0.0
X_3	42.9	0.0
(X_2, X_3)	23.8	0.0

Table D.8: QDA when $X_1 = 4$, discriminant functions based on entire training set

The observations in the training set are tested for skewness, kurtosis and homoscedasticity, see Tables D.9-D.11. The value for $\rho = 0.998$ while $\omega_2 = -2.89 \times 10^{-6}$ which allows for the use of the approximative distribution.

Group/Model	X_1	X_2	X_3	(X_1, X_2)	(X_1, X_3)	(X_2, X_3)	(X_1, X_2, X_3)
1	x	0.04	0.07	x	x	x	x
2	x	0.15	x	x	x	x	x
3	x	0.03	0.11	x	x	x	x
4	x	0.40	0.57	0.14	0.06	x	x

Table D.9: p-values for skewness test, an x means smaller than 10^{-2}

Group/Model	X_1	X_2	X_3	(X_1, X_2)	(X_1, X_3)	(X_2, X_3)	(X_1, X_2, X_3)
1	0.68	0.39	0.81	0.67	0.54	x	x
2	0.27	0.17	0.19	0.25	0.23	x	x
3	0.12	x	0.13	0.01	0.24	x	x
4	0.96	0.76	0.70	0.95	0.93	x	x

Table D.10: p-values for kurtosis test, an x means smaller than 10^{-2}

Model	X_1	X_2	X_3	(X_1, X_2)	(X_1, X_3)	(X_2, X_3)	(X_1, X_2, X_3)
M	2.99	58.21	38.56	126.37	100.03	171.73	260.78
p-value	0.39	x	x	x	x	x	x

Table D.11: p-values for homoscedasticity, an x means smaller than 10^{-2}

The observations in training set such that $X_1 = 4$ are tested for skewness, kurtosis and homoscedasticity, see Tables D.12-D.14. The value for $\rho = 0.99$ while $\omega_2 = -15 \times 10^{-5}$ which allows for the use of the approximative distribution.

Group/Model	X_2	X_3	(X_2, X_3)
1	0.47	0.74	x
2	0.42	0.18	x
3	0.64	0.21	x
4	0.89	0.56	x

Table D.12: p-values for skewness test, an x means smaller than 10^{-2}

Group/Model	X_2	X_3	(X_2, X_3)
1	0.18	0.26	0.67
2	0.18	0.67	x
3	0.17	0.19	x
4	0.62	0.69	x

Table D.13: p-values for kurtosis test, an x means smaller than 10^{-2}

Model	X_2	X_3	(X_2, X_3)
M	2.62	0.59	7.21
p-value	0.45	0.90	0.62

Table D.14: p-values for homoscedasticity, an x means smaller than 10^{-2}

Appendix E

Parameter settings

In this paper we study models of the UMTS network. The specific choice we made for the parameters of the wireless network are displayed in Table E.1 and Table E.2.

Parameter	Value
σ_S	6
f_{ca} (Mhz)	2000
h_b (m)	20
d_0 (km)	0.05
K_1	69.55
K_2	26.16
K_3	13.82
K_4	44.9
K_5	6.55

Table E.1: Path Loss

Parameter	Value
μ	$10^{0.60}$
G	60
N_f (mW)	$10^{-10.41}$
$P_{\max}$ (mW)	$10^{2.3}$
λ	4
$R_{\max}$ (km)	1

Table E.2: Other parameters

Acknowledgements

Obviously, this thesis could not have been realized without the help of some people. I want to thank some of these people specifically.

First I would like to thank Erik Fledderus for providing me with the assignment for my thesis and his guidance when I studied the diverse aspects of wireless network models. During the writing of my thesis I also needed to turn to other fields of science than my own specialization of probability. Bas Huiszoon helped me to understand the electrical engineering subtleties of CDMA protocols while Rudi Pendavingh helped me to fully understand the geometrical aspects of my thesis. The person with which I worked the most intensive is Remco van der Hofstad, I want to thank him for our numerous and often long talks on the diverse parts of my thesis. These talks helped me to shape my thesis into it's current form. Especially during the last hectic weeks of completing my thesis he was always available for advice.

The last, but certainly not the least, two persons which I want to thank are my parents. They gave me the opportunity to fully focus on my study without having to worry about tuition fees or the purchase of books. Besides the financial support they also are my most enthusiastic supporters. Despite of the fact that they are not able to fully understand this thesis they are at least as, if not more, enthusiastic and proud of it as I am.