Ein Markov-Chain-Monte-Carlo Ansatz zur Klassifikation und Überlebenszeitvorhersage von Krebspatienten auf Basis von Genexpressionsdaten

Diplomarbeit
zur Erlangung des akademischen Grades
Diplom Ingenieur (FH)

durchgeführt unter der Leitung von:

Dr. Lars Kaderali
Prof. Dr. Bernhard Haubold
Prof. Dr. Frank Leßke

eingereicht von

Bernhard Heinzel

im Fachbereich Bioinformatik
Fachhochschule Weihenstephan

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Viroquant Research Group Modeling
University of Heidelberg
Im Neuenheimer Feld 267
D-69120 Heidelberg

Fachhochschule
Weihenstephan
University of Applied Sciences

Fachhochschule Weihenstephan
Studiengang Bioinformatik
Am Hofgarten 4
D-85354 Freising

Email: bernhard.heinzel@gmail.com

Eingereicht am 14. Mai 2008.

Danksagung

Bedanken möchte ich mich besonders bei Dr. Lars Kaderali für die Bereitstellung der Diplomarbeit und die umfassende und gute Betreuung dieser Arbeit.

Außerdem möchte ich mich auch bei Nurgazy Sulaimanov für die Beantwortung meiner mathematischen Fragen, Matthias Böck für die kurzweiligen Diskussion sowie bei Daniel Ritter für die Bereitstellung der Markov-Chain-Monte-Carlo-Toolbox bedanken.

Abschließend gilt mein Dank natürlich Prof. Dr. Bernhard Haubold und Prof. Dr. Frank Leßke, die mich von Seiten der Fachhochschule Weihenstephan betreut haben.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass die vorliegende Arbeit von mir selbst und ohne fremde Hilfe verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt wurde. Es wurden keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt. Wörtliche und sinngemäße Zitate sind als solche gekennzeichnet.

Freising, den 14. Mai 2008

Bernhard Heinzel

Summary

With microarray experiments and their analysis we understand the cellular processes. cDNA screens measure the activity and the interactions of hundreds to thousands of genes. With gradual improvement of the measurement and the techniques of analysis we conceive the mechanisms of the cell regulation better. Current cancer reasearch has shown that gene expression data are strongly connected to the survival time of cancer patients.

A Markov Chain Monte Carlo approch using the Cox regression model to predict survival times from gene expression data is developed in this thesis. Afterwards a classification of the patients based on the predicted survival time should be carried out.

A difficulty in the analysis of microarray data is the large number of genes measured compared to the small number of patients typically available in clinical studies. To mitigate the problem we assume that only a few genes have a strong influence on the survival time. Due to that we apply an a-priori-distribution to ensure a sparse distribution of the regression parameters. This can be used to identify relevant genes, which later could be used as targets for drug design.

Using Markov chain Monte Carlo simulations combined with a Bayesian approach the distribution of the regression parameters is estimated. These can be used to predict the outcome of each individual patient based on this microarray data.

The main advantage of this Bayesian approach is that it analyses the entire distribution of the survival time. These survival time distributions are used to classify cancer patients into groups according to their risk and, possibly later, recommend treatments based on this classification.

In this thesis, the applied methods are presented and explained. Furthermore, the feasability of the method is tested on simulated data. Concluding this thesis the developed methods are applied on a clinical dataset.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Liste der Algorithmen

1 Einleitung

In dieser Arbeit soll ein Verfahren entwickelt werden, dass auf Grund von Genexpressionsdaten einzelner Patienten mit Hilfe des Cox-Regressions-Modells und Markov-Chain-Monte-Carlo-Simulationen Vorhersagen zu deren Überlebenszeit trifft und anschließend eine Klassifikation auf Grund ihrer Überlebenszeiten durchzuführen.

Mit Microarray Experimenten und deren Analysen lassen sich Einblicke in die zellulären Prozesse gewinnen. cDNA Screens messen die Aktivität und die Interaktionen von hunderten bis tausenden Genen. Die sukzessive Verbesserung der Messtechniken und Analysen liefern neue Einsichten in die Mechanismen der Zellregulation. So hat sich unter anderem in der Krebsforschung gezeigt, dass Genexpressionsdaten Aufschluss über die Überlebenszeit von Krebspatienten geben können.

Eine Schwierigkeit bei der Analyse von Microarray-Daten sind die enorme Anzahl von gemessenen Genen im Vergleich zu der geringen Anzahl von Datenpunkten (Patienten). Um die Probleme der Hochdimensionalität zu umgehen wird davon ausgegangen, dass es nur wenige signifikante Regressionsparameter gibt. Aus diesem Grund wird eine a-priori-Verteilung verwendet um die Regressionsparameter des Modells auszudünnen und so dafür zu sorgen, dass nur sehr wenige Gene einen großen Einfluss auf die Überlebenszeit haben. Dies kann dazu verwendet werden, um die relevanten Gene zu identifizieren. Die so ausfindig gemachten Gene könnten anschließend als Ziele für personalisierte Medikamente dienen, die an das genetische Profil des Patienten angepasst sind.

Durch Markov-Chain-Monte-Carlo-Simulationen kombiniert mit einem Bayes'schen Ansatz werden die Verteilungen der Regressionsparameter geschätzt, anschließend werden diese dazu verwendet, die Überlebenszeit des individuellen Patienten auf Basis der entsprechenden Microarray-Daten vorherzusagen.

Diese Methodik hat den Vorteil, dass sie es ermöglicht, die gesamte Verteilung über die Überlebenszeiten zu analysieren. Diese Überlebenszeitverteilungen werden anschließend dazu verwendet um Krebspatienten in Risikogruppen einzuteilen um später eventuell auf Basis dieser Klassifikation Behandlungsmethoden empfehlen zu können.

In dieser Arbeit sollen die verwandten Methoden vorgestellt und erläutert werden, anschließend wird die Funktionsweise der beschriebenen Methode an Hand eines simulierten Datensatzes getestet. Abschließend wird ein realer klinischer Datensatz evaluiert und ausgewertet.

2 Grundlagen

Die in dieser Arbeit verwendeten reellen Daten wurden aus Genexpressionsexperimenten bei Krebs-Patienten gewonnen. Um die folgenden Kapitel verständlicher zu gestalten wird in diesem Kapitel ein kurzer Überblick über die Grundlagen der verwendeten Methoden gegeben. Des weiteren werden die Themen Bayes'scher Rückschluss (Abschnitt 2.3) sowie Regressionsanalyse (Abschnitt 2.4) kurz angesprochen.

2.1 Krebs

Unter Krebs versteht man in der Medizin einen malignen (bösartigen) Tumor. Im Gegensatz zu gutartigen Tumoren wie Muttermalen und Fettgewülsten (Lipome) können maligne Tumore schwerwiegende Krankheiten verursachen, die bis zum Tod des Patienten führen können.

2.1.1 Entstehung

Krebs ist eine genetische Erkrankung. Krebs ist eine Krankheit, in der eine mutierte Zelle einen selektiven Vorteil gegenüber den normalen Zellen bekommt und sich auf Kosten ihrer Nachbarn vermehren kann.

Eigenschaften, die Zellen krebsartiges Wachstum erlauben:

- Sie missachten die externen und internen Signale zur Regulation der Zellproliferation
- Sie sind in der Lage, Selbstmord durch Apoptose zu vermeiden
- Sie umgehen programmierte Beschränkungen der Proliferation, indem sie replikative Alterung (begrenzte Anzahl an Zellteilungen) und Diffferenzierung vermeiden
- Sie sind genetisch instabil
- Sie verlassen ihr Heimatgewebe
- Sie überleben und vermehren sich an fremden Stellen (metastasieren)

Bei der Krebsentstehung laufen mehrere Zyklen von Mutation ab. Zellen, die von einer Vorläuferzelle abstammen, weichen anfangs nur leicht von der Norm ab. Nach und nach mutieren diese Zellen weiter. Diese Mutationen können ihr gegenüber ihren Nachbarzellen einen Vorteil verleihen. Ihre Nachkommen werden so zum dominanten Zellen die ihren Nachbarzellen Nährstoffe entziehen können und dadurch im Wachstum behindern.

Das Genom gesunder Zellen wird durch die Produkte verschiedener Gene vor Veränderungen geschützt. Mutationen in diesen Genen haben eine genetische Instabilität zur Folge. Diese erhöhte genetische Instabilität erlaubt es Krebszellen Selektionshürden, wie Mangel an Wachstumssiganlen oder Apoptose zu überwinden.

Die Mutation von Genen in unserem Körper sind keine Seltenheit. Durch äußere Einflüsse oder Fehlfunktionen in der Zelle mutieren ständig Zellen. Normalerweise sorgt das Reparatursystem der Zellen dafür, dass solche Mutationen behoben werden oder, wenn dies nicht mehr möglich ist, dass die Zelle sich selbst zerstört.

Onkogene und Tumorsuppressor-Gene kommen in allen gesunden Körperzellen vor. Diese zwei Gruppen von Genen regulieren dort das Zellwachstum (Proliferation) und die Zellreifung (Differenzierung) der Zellen. Onkogene kontrollieren das Zellwachstum und fördern dieses, Tumorsuppressor-Gene hingegeben unterdrücken dieses. In gesunden Zellen herrscht ein ausgeklügeltes Gleichgewicht zwischen Onkogenen und Tumorsuppressor-Genen. Gerät diese Balance zwischen Zellzyklus (Wachstum und Teilung) und Zelltod außer Kontrolle, so beginnt die Zelle unkontrolliert zu wachsen und sich zu teilen und verdrängt so gesundes Gewebe. Abbildung 2.1 zeigt den Vergleich der Zellteilung zwischen gesunden Zellen und entarteten Krebszellen.

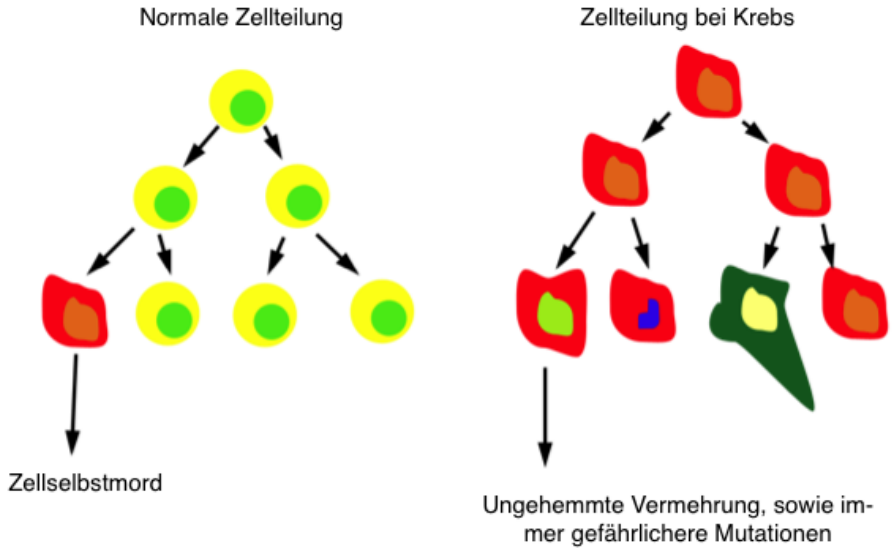


Abbildung 2.1: Vergleich des Verlaufs von normaler Zellteilung zur Zellteilung von entarteten Krebszellen

2.1.2 Ursachen genetischer Veränderung

Man unterscheidet zwischen drei Gruppen von Karzinogenen (Krebserzeugern):

1. Chemische Substanzen
2. Viren

3. Strahlung

Die am häufigsten vorkommende Gruppe von Karzinogenen sind chemische Substanzen. Asbest, polyzyklische aromatische Kohlenwasserstoffe, Benzol oder Nitrosamine (enthalten in Tabakrauch) sind nur einige Beispiele für krebserzeugende Stoffe. Aber auch Aflatoxine (Schimmelpilzgifte) gehören zu diesen Karzinogenen.

Viren sind inzwischen auch dafür bekannt, dass sie Krebs begünstigen und auslösen können. So ist bekannt, dass Hepatitis-Viren Leberzellkrebs auslösen können und das Gebärmutterhalskrebs (Cervix-Carcinom) durch bestimmte Gruppen von Papillomaviren verursacht wird.

Die häufigste Strahlung, der der Mensch ausgesetzt ist, die bekannt dafür ist, Krebs auslösen zu können, ist die Sonnenstrahlung. Diese kann im genetischen Code der Hautzellen Mutationen initiieren und somit Hauttumore hervorrufen. Radioaktive Strahlung ist bekannt dafür, ein Auslöser von Leukemie zu sein. Die Nobelpreisträgerin Marie Curie ist ein prominentes Opfer der von Strahlung hervorgerufener Leukämie (Blutkrebs).

2.1.3 Therapiemöglichkeiten

Aufgrund der Vielfalt von Krebserkrankungen gibt es keine allgemein wirksame Behandlung. Hinzu kommt, dass die möglichen Therapiemöglichkeiten stark von der Lokalisation des Karzinoms sowie von dessen Entwicklung und Ausbreitung abhängen. In den folgenden Abschnitten sollen nur vier der am häufigsten angewandten Behandlungsmethoden beschrieben werden. Weitere Behandlungsmethoden, die hier keine weitere Erwähnung finden, sind unter anderem die Immuntherapie, die Hemmung des Blutgefäßwachstums und die Behandlung mit Hormonpräperaten..

Operative Entfernung

Handelt es sich beim diagnostizierten Karzinom um einen lokal ortbaren Tumor, so kann dieser in vielen Fällen mit Hilfe eines operativen Eingriffs entfernt werden. Bei diesen Eingriffen wird der Tumor an sich, sowie umgebendes Gewebe chirurgisch entfernt. Sollte der Tumor bereits Metastasen gebildet haben, ist die vollständige Entfernung meist nicht mehr möglich. Unter Metastisierung versteht man die Ausbreitung eines Tumors über die ursprüngliche Region hinweg. So entstehen neben den Primärtumor örtlich differenzierte Sekundärtumore.

Zytostatika

Zytostatika sind Substanzen, die das Zellwachstum und die Zellteilung hemmen. Sie werden in der Chemotherapie verwendet um das Zellwachstum sowie die Zellteilung der Krebszellen zu stören und so eine weitere Ausbreitung des Tumors zu verhindern.

Zytostatika blockieren Stoffwechselvorgänge der Zellteilung und des Zellwachstums, dadurch schädigen sie vor allem Krebszellen, denn diese weisen eine höhere Geschwindigkeit bei der Zellteilung auf als

gesunde Zellen. Da Zytostatika aber nicht so selektiv sind und nicht nur Krebszellen beeinträchtigen, kommt es zu Nebenwirkungen der Behandlung. Fast immer werden Haarwurzelzellen aufgrund ihrer hohen Reproduktion geschädigt. Neben dem daraus resultierenden Haarausfall leiden die Patienten oft unter Erbrechen sowie einer Verminderung der weißen und/oder roten Blutkörperchen im Blut (Knochenmarksdepression). Zytostatika werden vor allem bei der Behandlung von Tumoren verwendet die sich über mehrere Körperregionen ausgebreitet haben, Beispiele hierfür sind Leukämien, Lymphome sowie Tumore die bereits Metastasen gebildet haben.

Strahlentherapie

Bei der Strahlentherapie werden vorwiegend hochenergetische Strahlen (Gammastrahlung und Röntgenstrahlung) verwendet um Tumore zu verkleinern oder komplett zu zerstören. Hierbei wird mit Hilfe der Strahlung das genetische Material der bestrahlten Zellen zerstört oder soweit geschädigt, das sich die Zelle nicht weiter teilen kann. Hierbei wird ausgenutzt, dass Tumorzellen empfindlicher auf die Strahlung reagiert als normale Zellen, da die Reperaturmechanismen schlechter arbeiten als in gesunden Zellen. Hierbei gilt ebenso wie bei Zytostatika, dass eine Schädigung der umliegenden normalen Zellen nicht ausgeschlossen werden kann. Deshalb behandelt man den Patienten in einer Strahlentherapie in einer Reihe von Sitzungen, wodurch gesundes Zellgewebe die Möglichkeit hat sich zu regenerieren. Eine Strahlentherapie kann besonders gut bei soliden Tumoren sowie in Kombination mit der Verabreichung von Zytostatika eingesetzt werden.

Targets für Drug Design

Eine weiter noch in den Kinderschuhen steckende Methode zur Krebsbehandlung ist das *silencing* von bestimmten Genen. Hierunter versteht man das gezielte ausschalten der Expression von bestimmten Genen. Die Publikation von Kim et al. [Kim03] beschreibt, wie der RNA Interferenz (RNAi) genannte Mechanismus in Pflanzen und Tierzellen dazu verwendet wird, um bestimmte Gene von der Expression auszuschließen.

Chi et al. zeigen in ihrer Publikation [CCW+03], dass die für das Ausschalten verantwortlichen doppelsträngigen *small interfering RNA* (siRNA) äußerst genspezifisch sind und keine unbeabsichtigte Nebeneffekte in den Zelllinien hervorrufen.

So könnte diese Methode des *gene silencing* dazu verwendet werden, gezielt Gene auszuschalten die für die Entwicklung von Krebszellen verantwortlich sind oder deren Vermehrung unterstützen. Experimente in dieser Richtung wurden bereits erfolgreich durchgeführt, so haben Liu et al. [LYS+04] durch siRNA gezielt Brustkrebstumorzellen an ihrer Proliferation (Zellteilung) gehindert und die Apoptose (Zelltod) induziert.

Dies zeigt, dass es für die praktische Entwicklung von entscheidender Wichtigkeit ist, Onkogene sowie Tumorsupressor-Gene zu identifizieren.

2.2 Genregulation

Der molekularbiologische Prozess durch den in der Zelle bestimmt wird, wann und wie stark Gene aktiviert werden, nennt man Genregulation. Hierbei interagieren DNA, RNA, Proteine und andere Substanzen, um Gene in bestimmten Mengen zu exprimieren oder deren Expression zu unterdrücken. Die hinter der Genregulation steckenden Mechanismen sind äußerst komplex und ihre Darstellung würde den Rahmen dieser Diplomarbeit überschreiten. Deswegen soll hier nur kurz der grundlegende Ablauf der Genexpression beschrieben werden. Anschließend wird die experimentelle Methode beschrieben, mit der die Daten der Genexpression gewonnen werden können.

2.2.1 Genexpression

Genexpression bezeichnet die Ausprägung der, in Genen codierten, genetischen Information. So wird im allgemeinen die Genexpression in drei Schritten beschrieben:

1. Transkription
2. Translation
3. Postranslationale Modifikation

Neben den aufgezählten Schritten hat der Genlocus ebenso einen Einfluss auf die Ausprägung der genetischen Information. Der Genlocus ist die physikalische Lage eines Gens. Da die DNA im Zellkern nicht linear sondern stark gefaltet ist, können bestimmte Bereiche des Chromosoms nicht exprimiert werden wenn die DNA nicht in einer bestimmten Konformation vorliegt.

Soll die Information, die sich auf einem bestimmten Bereich des Chromosoms (Gen) befindet, exprimiert werden, so wird zu erst mit Hilfe der RNA (Ribonukleinsäure) die gewünschte DNA Sequenz enzymatisch in eine komplementäre mRNA (messenger RNA) kopiert.

Anschließend wird die erhaltene mRNA benutzt, um die darin erhaltenen genetischen Informationen in ein entsprechendes Protein zu synthetisieren. Nachdem das Protein vollständig synthetisiert ist, muss es noch in seine aktive räumliche Konformation gebracht werden. Hierzu besitzt die Zelle spezielle Proteine, die Chaperone, die dem synthetisierten Protein bei der Faltung helfen.

Für eine detaillierte Beschreibung der Genregulation empfehle ich [Kni06].

2.2.2 DNA-Microarrays

Die DNA-Microarray-Technolgie wird verwendet, um ein quantitatives Maß für die Expression von Genen zu erhalten. Die in dieser Diplomarbeit verwendeten realen Genexpressionsdaten wurden mit Hilfe von Oligonukleotid-Arrays gewonnen. Deswegen soll hier nur auf dieses Verfahren eingegangen werden. Weitere Information zu dem verwendeten Datensatz befinden sich in Abschnitt 5.2.1.

Um die Expression der Gene einer Zelle festzustellen, misst man die mRNA-Menge (messenger RNA). der einzelnen Gene. Hierbei geht man wie folgt vor. Zu erst extrahiert man aus den Zellen die mRNA. Anschließend wird die mRNA in cRNA (complementary RNA) umgeschrieben, die beispielsweise mit Floureszenzfarbstoffen markiert wird.

Im zweiten Schritt gibt man dann die cRNA auf die Oligonukleotid-Arrays. Diese Arrays bestehen aus einer Oberfläche, auf die Oligonukleotide angebracht sind, so genannte Spots. Gibt man nun die mit Floureszenzfarbstoffen markierten cRNA auf diese Oberfläche, so hybridisieren die cRNAs mit ihren komplementären Oligonukleotiden. Diese Hybridisierung findet unterschiedlich häufig statt.

Nachdem die unhybridisierten cRNA Stücke entfernt wurden, scannt man, üblicherweise mit einem Laser, das Array und detektiert hierbei die Itensitäten der Floureszenzfarbstoffe. Dadurch erhält man die an den Oliogonukleotid-Spots hybridisierten cDNA-Mengen. Anschließend werden die Signale noch normalisiert. Abbildung 2.2 zeigt ein Bild eines solchen Arrayexperiments. Hier sieht man deutlich die unterschiedlichen Intensitäten der Oliogonukleotid-Spots.

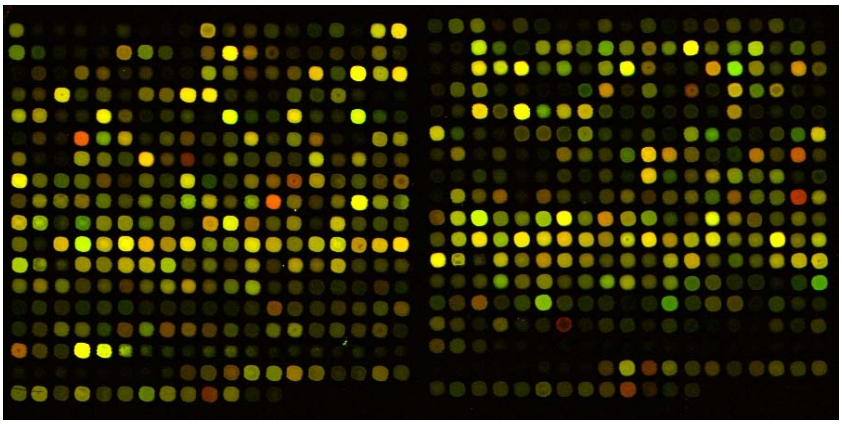


Abbildung 2.2: Beispiel eines cDNA Microarrays. Umso intensiver der Spot, desto mehr cDNA hybridisierte.

2.3 Bayes'scher Rückschluss

Unter Bayes'schen Rüchschluss versteht man eine statistische Methode mit Hilfe dieser, von Stichproben oder Beobachtungen auf die zugrunde liegende Wahrscheinlichkeitsverteilung geschlossen werden kann.

2.3.1 Bayes' Theorem

Das *Bayes' Theorem* (Gleichung (2.1)) erlaubt die Umkehrung von Schlussfolgerungen. Dies ermöglicht es aus Stichproben oder Beobachtungen auf die zugrunde liegende A-Posteriori-Wahrscheinlichkeit zu schließen.

Für zwei Ereignisse A und B gilt demnach:

$$P(A|B) \;=\; \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.1}$$

Hierbei ist:

- $P(A)$ die A-Priori-Wahrscheinlichkeit für das Ereignis A und
- $P(B|A)$ die Wahrscheinlichkeit für das Ereignis B unter der Bedingung, dass A eingetreten ist und
- $P(B)$ die A-Priori-Wahrscheinlichkeit für das Ereignis B.

Dadurch wird der *Bayes'scher Rückschluss* zu einem fundamentalen Teil von simulationsbasierten Verfahren wie der Gruppe von Monte-Carlo-Simulationen. Bei diesen Verfahren wird versucht, analytisch komplexe Fragestellung durch Simulationen dieser zu beantworten. So ist es oft möglich, ein bestimmtes Experiment zu simulieren ohne die zugrunde liegenden Mechanismen zu kennen. Anschließend wird versucht, durch die Ergebnisse der Simulationen auf die zugrunde liegenden Mechanismen zu schließen.

Ein Beispiel hierfür wären die Berechnung der Gewinnwahrscheinlichkeit, mit einer bestimmten Pokerhand. Die Berechnung der Wahrscheinlichkeit mit einer gegebenen Hand zu gewinnen, ist ein kombinatorisch äußerst schwer zu lösendes Problem. Es ist aber durchaus möglich eine Pokerpartie am Computer zu simulieren. Um nun mit Hilfe eines Monte-Carlo-Ansatzes die Gewinnwahrscheinlichkeit zu ermitteln, simuliert man einfach eine sehr hohe Nummer von Pokerpartien und kann an Hand der resultierenden Ergebnisse auf die gesuchte Gewinnwahrscheinlichkeit zurück schließen.

Für eine detaillierte Beschreibung des in dieser Diplomarbeit verwendeten Markov-Chain-Monte-Carlo-Verfahrens siehe Kapitel 3.

2.4 Regressionanalyse

Die Regressionsanalyse ist ein statistisches Analyseverfahren. Es wird benutzt, um die Beziehung einer abhängigen Variable Y und einer oder mehreren unabhängigen Variablen $\overrightarrow{x} = \{x_0, \cdots, x_n\}$ zu finden.

Ist ein Satz von Merkmalen $\overrightarrow{x} = \{x_0, \cdots, x_n\}$ gegeben, so ist das Ziel, eine Funktion $f(\overrightarrow{x})$ zu finden, die den Fehler der Abbildung von $\overrightarrow{x}$ auf Y minimiert.

Kombiniert man nun die Regressionsanalyse mit statistischen Modellen, kann man dadurch Vorhersagen über den zukünftigen Verlauf oder Rückschlüsse auf die Beziehungen der Variablen Y und x machen. Die Verwendbarkeit hängt sehr stark von dem verwendeten Modell ab.

Diese Regressionsmodelle bestehen normalerweise aus einer Gleichung (Gleichung (2.2)) welche die gegebenen Merkmale $\overrightarrow{x} = \{x_0, \cdots, x_n\}$ mit den Regressionsparametern $\overrightarrow{\theta} = \{\theta_0, \cdots, \theta_n\}$ verknüpft.

Die Qualität solch eines Regressionmodells, das heißt wie gut das Modell die gegebenen Daten abbildet, hängt sehr stark von der Wahl des Regressionsmodels ab:

$$f(\overrightarrow{x}) = \overrightarrow{x} \times \overrightarrow{\theta} \tag{2.2}$$

Die grundlegende Problemstellung ist nun die Regressionsparameter zu schätzen. Um die Regressionsparameter möglichst gut zu schätzen, bedient man sich den unterschiedlichsten Methoden. Beispiele hierfür sind das Minimieren des quadratischen Fehlers, der Maximum-Likelihood-Schätzer oder die Methode des Bayes'schen Rückschlusses.

Eine detaillierte Beschreibung des verwendeten Cox-Regressionsmodells in Kombination mit dem Bayes'schen Rückschlusses befindet sich in Abschnitt 4.1.

2.5 Details zur Implementierung

Da Markov-Chain-Monte-Carlo-Simulationen im allgemeinen und der Hybrid-Monte-Carlo-Algorithmus im speziellen sehr rechenintensive stochastische Methoden sind, musste bei der Wahl der Programmiersprache vor allem auf die Geschwindigkeit Rücksicht genommen werden. Aus diesem Grund fiel die Wahl auf die prozessornahe Programmiersprache C++. Ein weiterer wichtiger Aspekt der für C++ sprach, war die Möglichkeit zur Objektorientierung. Als Compiler wurde der freie Compiler *gcc* benutzt.

Der Sourcecode der zugrunde liegenden Markov-Chain-Monte-Carlo-Algorithmen wurde der Markov-Chain-Monte-Carlo-Toolbox von Daniel Ritter [Rit07] entnommen. Die Implementation der Regressionsanalyse, die benötigten Module für Dateneingabe und Datenausgabe, sowie die mathematischen Verfahren wurden während der Diplomarbeit implementiert. Bei der Implementierung wurde darauf geachtet, dass die Funktionen möglichst effektiv und performant implementiert werden. Vor allem bei den Zielfunktionen des Hybrid-Monte-Carlo-Algorithmus musste sehr darauf geachtet werden, keine unnötigen Aufrufe und Deklarationen zu benutzten, da diese Funktion mehrere hundert Millionen mal aufgerufen wird und maßgeblich die Performanz der Anwendung beeinflusst.

Die Berechnungen wurden auf einem 8 Prozessor Linux Cluster mit 32 GB RAM durchgeführt.

3 Markov Chain Monte Carlo

Wie bereits erwähnt, können *Monte Carlo Simulationen* benutzt werden, um analytisch schwer lösbare Probleme numerisch zu lösen. Um solch kombinatorisch schwer zu lösende Probleme zu analysieren, wiederholt man das Experiment dessen Lösung ermittelt werden soll, um durch die beobachteten a-posteriori-Wahrscheinlichkeiten mit Hilfe des *Satzes von Bayes* die erwarteten a-priori-Wahrscheinlichketen zu erhalten. Für eine weiterführende Einführung siehe [AdFDJ03]. In den folgenden Abschnitten werde ich auf die zwei grundlegenden Bestandteile einer Markov ChainMonte Carlo Simulation eingehen, dieMonte Carlo Integration (Abschnitt 3.1) und die Markovketten (Abschnitt 3.2).

3.1 Monte Carlo Simulation

Um den Erwartungswert $E[f(\theta)]$ berechnen zu können, generiert man Stichproben (Samples) $\theta^{(t)}$, $t = 1, ..., n$, von der zu untersuchenden Verteilung $p(.)$. Der Erwartungswert kann anschließend mit

$$E[f(\theta)] \approx \frac{1}{n} \sum_{t=1}^{n} f(\theta^{(t)}), \tag{3.1}$$

berechnet werden.

Dies bedeutet, dass der Erwartungswert von $f(\theta)$ durch den Erwartungswert der Stichproben $\theta^{(t)}$ geschätzt wird. Das *Gesetz der großen Zahlen* besagt, dass bei Unabhängigkeit der Stichproben von der Verteilung $p(.)$ mit Erhöhung der Stichprobenanzahl n auch die Genauigkeit des geschätzten Erwartungswertes $E[f(\theta)]$ steigt. Die Prämisse der Stichprobenunabhängigkeit kann vernachlässigt werden, wenn die Stichproben unter bestimmten Bedingungen gezogen werden. Eine Möglichkeit, solche Stichproben zu generieren, ist eine Markovkette mit $p(.)$ als stationärere Verteilung.

Eine genaue Darstellung der Sampling Algorithmen befindet sich in Kapitel 3.3.

3.2 Markovketten

Die Darstellung und Erläuterung von Markovketten in diesem Abschnitt ist weit davon entfernt, vollständig und umfassend zu sein. Für eine weiterführende und detailierte Einführung in das Thema Markovketten empfehle ich Kapitel 4, [Gam97].

Markovketten sind eine Klasse von stochastischen Prozessen. Die wichtigste Eigenschaft von Markovketten ist, dass bei einer Markovkette n-ter Ordnung die Wahrscheinlichkeit für das Eintreffen eines bestimmten Ereignisses nur von den n vorhergegangenen Zuständen abhängt. Im Falle einer Markovkette 1. Ordnung heißt das, dass das zukünftige Ereignis x^{t+1} nur von dem gegenwärtigen Zustand x^t abhängt. Dabei ist es für die Vorhersage von x^{t+1} unwichtig, ob die gesamte Vorgeschichte $\left\{x^{t-n}, ..., x^{t-1}, x^t\right\}$, $n \in N$ oder nur das vorhergehende Ereignis x^t bekannt ist.

Eine zeitlich diskrete ($t \in \{0, 1, 2, ..., n\}$) Markovkette mit einem endlichen Zustandsraum ($S = \{\theta_1, ..., \theta_m\}$) n-ter Ordnung wird wie folgt definiert:

$$P(\theta^{(t+1)}|\theta^{(t)}, \theta^{(t-1)}, ..., \theta^{(0)} = P(\theta^{(t+1)}|\theta^{(t)}, \theta^{(t-1)}, ..., \theta^{(t-n+1)}) \tag{3.2}$$

Die Gleichung beschreibt, dass die Wahrscheinlichkeit, dass das Ereignis θ_j eintritt, nur von den n vorhergegangenen Ereignissen abhängt.

Im Falle einer Markovkette 1. Ordnung reduziert sich die Gleichung (3.2) auf

$$P(\theta^{(t+1)}|\theta^{(t)}, \theta^{(t)}, ..., \theta^{(0)} = P(\theta^{(t+1)}|\theta^{(t)}). \tag{3.3}$$

Im folgenden Absatz werden Zustände und Stichproben als Synonyme verwendet. Solch eine Markovkette besitzt eine Übergangswahrscheinlichkeit $\pi(.)$ mit der sie zum nächsten Zustand übergeht. In unserem Fall gehen wir von einer homogenen Markovkette aus, das heißt, dass die Übergangswahrscheinlichkeit $\pi(.)$ unabhängig von der Zeit t ist. Wie von Gilks et al [GRS96] beschrieben, *vergisst* solch eine Markovkette ihren Startzustand und konvergiert zu einer einzigartigen stationären Verteilung $k(.)$. Die Zustände der Markovkette sind unabhängig von dem Initalzustand $\theta^{(t)}$ und der Zeit t. Mit zunehmender Zeit t verhalten sich die Zustände $\theta^{(t)}$ wie Stichproben der stationären Verteilung $k(.)$.

Das bedeutet, dass die *generierten* Zustände der Markovkette benutzt werden können, um den Erwartungswert $E[f(\theta)]$ zu schätzen, da θ sich nach der Verteilung $k(.)$ verhält.

3.3 Sampling Algorithmen

In Abschnitt 3.2 wurde gezeigt, dass sich, mit Hilfe einer Markovkette, die eine stationäre Verteilung $k(.)$ angenommen hat, welche der gewünschten Verteilung $p(.)$ entspricht, der Erwartungswert $E[f(\theta)]$ berechnen lässt. Um einen Richtwert zu erhalten, nach wie vielen Iterationen eine Markovkette konvergiert ist, werden mehrere Markovketten mit unterschiedlichen Initalwerten gesatertet. Anschließend wird verglichen, nach wie vielen Iterationen sich die erzeugten Werte annähern.

In den nächsten Abschnitten beschreibe ich den Metropolis-Hastings-Algorithmus (3.3.1) und den Hybrid-Monte-Carlo-Algorithmus (3.3.2) die benutzt werden können, um eine Markovkette mit den gewünschten Attributen zu erstellen.

3.3.1 Metropolis-Hastings-Algorithmus

Um Stichproben von der gewünschten Verteilung $p(\theta)$ zu ziehen, wird zuerst ein Stichprobenkandidat von einer Vorschlagsverteilung (Proposal Distribution) mit der Dichte $\pi(\theta^{(t+1)}|\theta^{(t)})$ gezogen.

Der Kandidat wird dann mit der Wahrscheinlichkeit

$$\alpha(\overline{\theta}|\theta^{(t)}) := min\left\{1, \frac{p(\overline{\theta})\pi(\theta^{(t)}|\overline{\theta})}{p(\theta^{(t)})\pi(\overline{\theta}|\theta^{(t)})}\right\} \tag{3.4}$$

akzeptiert. Wenn der Kandidat akzeptiert wird, geht die Markovkette in den nächsten Zustand $\theta^{(t+1)}$ über. Wird der Kandidat abgelehnt, bleibt die Markovkette im Zustand $\theta^{(t)}$. τ ist die Anzahl der Zeitschritte über die die Markovkette iteriert.

Algorithmus 1 : Metropolis-Hastings vgl. Neal [Nea96], Seite 26

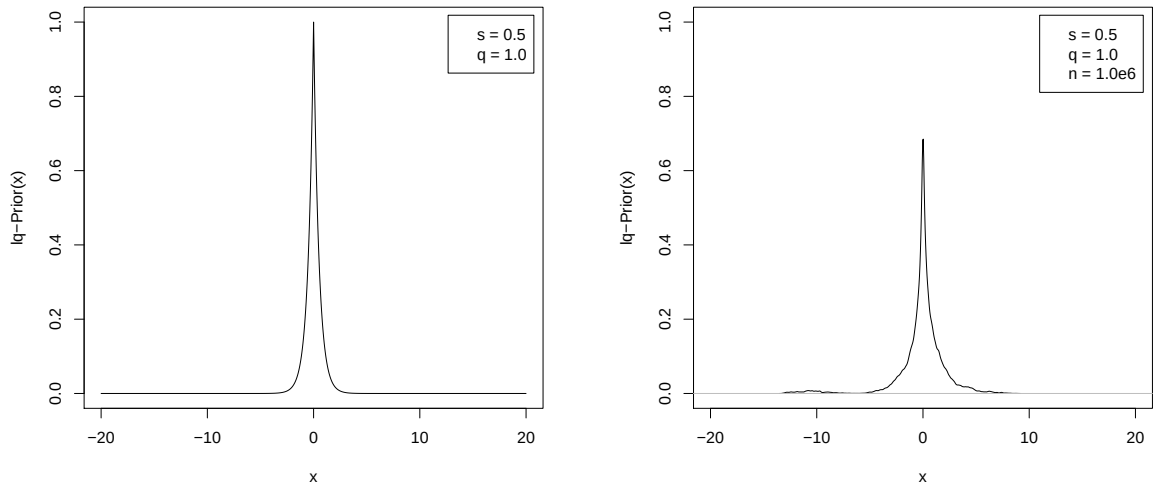
Initialize $\theta^{(0)}$
$t \leftarrow 0$
while $t < \tau$ **do**
  Sample $\overline{\theta}$ from $\pi(.|\theta^{(t)})$
  Sample uniform (0,1) random variable U
  $\alpha \leftarrow min(1, \frac{p(\overline{\theta})\pi(\theta^{(t)}|\overline{\theta})}{p(\theta^{(t)})\pi(\overline{\theta}|\theta^{(t)})})$
  if $U \leq \alpha$ **then**
    $\theta^{(t+1)} \leftarrow \overline{\theta}$`/* new point` $\overline{\theta}$ `accepted */`
  else
    $\theta^{(t+1)} \leftarrow \theta^{(t)}$`/* new point` $\overline{\theta}$ `rejected */`
  Output : $\theta^{(t+1)}$
  $t \leftarrow t + 1$

Die Vorschlagsdichte $\pi(.|\theta^{(t)})$ kann jeder arbiträren Form entsprechen und dennoch ist $p(\theta)$ die stationäre Verteilung der Markovkette. Für die mathematische Herleitung siehe [Kad06].

Wie in Abschnitt 3.2 angenommen, wird die Markovkette unabhängig ihrer Initialwerte zu ihrer stationären Verteilung $p(\theta)$ konvergieren. Ist dies der Fall, spricht man von Irreduzibilität der Markovkette.

Beispiel LQ-Prior

Der hier beschriebene Metropolis-Hastings-Algorithmus, wurde angewedet um Samples von dem sogenannten LQ-Prior (3.5) zu generieren. Abbildung 3.1(a) zeigt einen Plot des LQ-Priors mit den



(a) (b)

Abbildung 3.1: Plot der originalen a-priori-Verteilung die als Zielfunktion für den den Metropolis-Hastings-Algorithmus (a) dient und der Dichteplot der erstellten Stichproben (b).

Parametern $s = 0.5$ und $q = 1.0$.

$$p(\theta) = \exp\left[-\frac{1}{q \cdot s^q} \cdot |x|^q\right] \tag{3.5}$$

Es wurden mit Hilfe des Metroplolis-Hastings-Algorithmus 10^6 Stichproben gezogen. Die Akzeptanzrate der Markovkette belief sich auf $\sim 78\%$. Ein Plot der Dichte der gezogenen Stichproben ist in Abbildung 3.1(b) abgebildet.

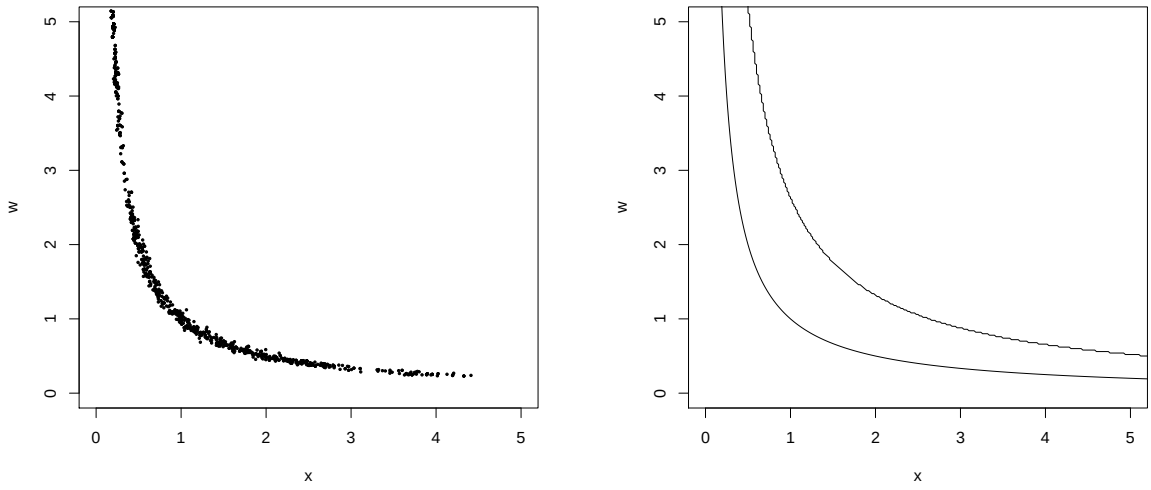
Es zeigt sich hier sehr deutlich der für die Zielfunktion charakteristische Peak um den Ursprung von x. Es fällt auf, dass die Höhe des Peaks der Stichprobenverteilung etwas kleiner ausfällt als die des Peaks der Zielfunktion. Das kann darauf zurückgeführt werden, dass nicht genug Stichproben gezogen wurden. Erhöht man die Anzahl der gezogenen Stichproben auf etwa $10^7 - 10^8$, so gleicht sich die Höhe des Peaks immer mehr der der Originalfunktion an. Werden nur wenige hundert Stichproben gezogen, so ist der Peak deutlich zu erkennen, die Höhe des Peaks hingegen ist deutlich unterhalb der Höhe der Originalfunktion.

Komplexere mehrdimensionale Funktionen

Ein großer Nachteil des Metropolis-Hastings-Algorithmus ist das *random walk behavior*. Der Ausdruck *zufälliger Lauf* bezieht sich hier auf die Generierung des Markovkette. In unserem Fall heißt das, dass der Funktionsraum um den aktuellen Wert untersucht wird. Das bedeutet, dass der gesamte Funktionsraum durchlaufen wird, dafür aber eine sehr hohe Anzahl von Iteration notwendig ist.

Wie bereits erwähnt, sind sogar für einfache, 1-dimensionale Funktionen wie den LQ-Prior mehrere Millionen Iterationen (Stichproben) notwendig, um den Funktionsraum angemessen zu durchqueren.

Bei komplexeren Funktionen wie der bananenförmigen Funktion $N(y|wx, \sigma)N(x|w, 1)$ zeigt sich dieses Verhalten noch deutlicher. Abbildung 3.2 zeigt, wie die gewünschte Funktion aussehen würde nach 10000 Iterationen des Metropolis-Hastings-Algorithmus. Hier kann die charakteristische Bananenform deutlich erkannt werden. Reduziert man hingegen die Stichprobenanzahl auf wenige Hundert, so kommt die Bananenform nicht deutlich zum vorschein.



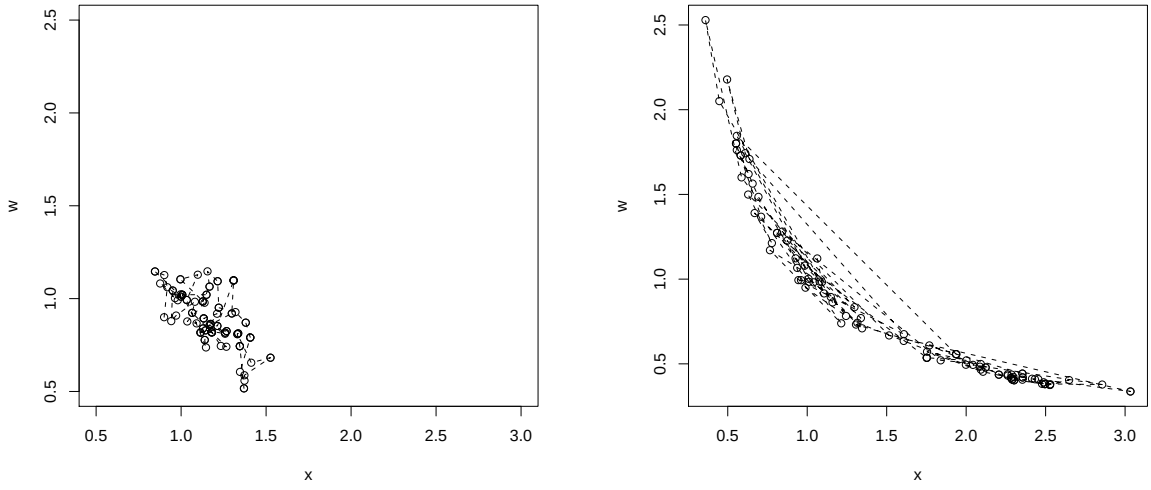
(a) 1000 Iterationen durch den Funktionsraum der Bananenfunktion

(b) Contour-Plot der Bananenfunktion

Abbildung 3.2: 10000 Iterationen durch den Funktionsraum der Bananenfunktion (a) sowie ein Contour-Plot der Bananenfunktion (b).

Abbildung 3.3(a) zeigt nur 100 Iteration durch den Funktionsraum der Bananenfunktion. Man sieht hier deutlich das *random walk behavior*, so dass die Markovkette nach 100 Iteration noch nicht genügend zur Zielfunktion (Bananenfunktion) konvergiert ist. Dieses Verhalten macht den Algorithmus für viele Anwendungen zu einer äußerst schlechten Wahl (für eine genaue Analyse siehe Chen et al. [CQL]). Des weiteren bewegt sich die Akzeptanzrate nur noch im Bereich $30 - 35\%$.

Um dieses *random walk behavior* zu vermeiden, wurden der Hybrid-Monte-Carlo-Algorithmus entwickelt, der dieses Verhalten vermeidet, indem er Ideen aus den Hamilton-Formalismus verwendet und die Hamilton-Energiefunktion benutzt. Im folgenden Abschnitt werde ich die zugrundeliegenden Ideen erklären und auf den daraus entstandenen Hybrid-Monte-Carlo-Algorithmus eingehen.



(a) (b)

Abbildung 3.3: Jeweils 100 Iteration mit Metropolis-Hastings (a) und dem Hybrid-Monte-Carlo-Algorithmus (b).

3.3.2 Hybrid-Monte-Carlo-Algorithmus

Der Hybrid-Monte-Carlo-Algorithmus wurde 1987 von Duane et al. entwickelt [DKPR87]. Er bedient sich Ideen der Molekülbewegungstheorie, um das *random walk behavior* des Metropolis-Hastings-Algorithmus einzuschränken.

Man nehme an, dass wir von einer Wahrscheinlichkeitsverteilung $p(\theta) \propto \exp(-E(\theta)), \theta = (\theta_1, \cdots, \theta_n) \in \mathbb{R}$ Stichproben ziehen wollen und $E(\theta)$ eine beliebige *Energiefunktion* ist. Man kann jede Verteilung $\pi(\theta)$ in solch eine Energiefunktion durch

$$E(\theta) = -\log p(\theta) \tag{3.6}$$

überführen. Es sei erwähnt, dass die Zielverteilung niemals 0 sein darf, da $-\log(0) = \infty$.

Neben der *Energiefunktion* führten Dueane et al. noch die sogenannten Impulsvariablen $\rho = (\rho_1, \cdots, \rho_n) \in \mathbb{R}$ und die *kinetische Energie* $K(\rho)$ ein:

$$K(\rho) = \sum_{i=1}^{n} \frac{1}{2} \frac{\rho_i^2}{m_i}, \tag{3.7}$$

wobei m_i die Masse, die mit der Impulsvariablenkomponenten ρ_i assoziiert ist. In dieser Diplomarbeit sei der Einfachheit halber angenommen, dass immer $m_i = 1$ gilt. Nun erhalten wir mit der *Hamiltonischen Funktion*

$$H(\theta, \rho) = E(\theta) + K(\rho), \tag{3.8}$$

die Gesamtenergie unseres Systems.

Die Idee des Hybrid-Monte-Carlo-Algorithmus ist es, den Sampling-Vorgang in zwei Schritte zu unterteilen. Im ersten Schritt werden gleichverteilte Stichproben von θ und ρ gezogen, wobei die Gesamtenergie $H(\theta, \rho)$ konstant bleibt. Dieser Schritt variiert dann die Gesamtenergie $H(\theta, \rho)$, während für θ und ρ Stichproben gezogen werden. Der erste Schritt ermöglicht es, große, Schritte durch den Funktionsraum zu machen, während der zweite Schritt es ermöglicht, jeden Ort am Funktionsraum zu erreichen.

Der erste Schritt wird ausgeführt durch Simulation der *Hamiltonschen Mechanik* mit Variation der Zeit τ.

$$\frac{d\theta_i}{d\tau} = \frac{\partial H}{\partial \rho_i} = \frac{\rho_i}{m_i} \tag{3.9}$$

$$\frac{d\rho_i}{d\tau} = -\frac{\partial H}{\partial \theta_i} = -\frac{E}{\theta_i} \tag{3.10}$$

Die Gleichungen (3.9) und (3.10) beschreiben, wie sich der Zustand unseres Systems verändert.

Im zweiten Schritt werden mit Hilfe des Gibbs-Sampling [GGA$^+$93] Stichproben von den Impulsvariablen ρ mit $p(\rho) \propto \exp(-K(\rho))$ gezogen. Ich verzichte hier auf eine genaue Beschreibung des Gibbs-Sampling-Algorithmus, da dieser für das Verständnis nicht unbedingt von Nöten ist. Für eine ausführliche Beschreibung des Gibbs-Sampling-Algorithmus siehe die Veröffentlichungen von Gelfand et al. [GS90, GHRPS90].

Theoretisch sind die Hamiltonschen Dyamiken vollständig stetig, praktisch kann man ihnen aber nicht exakt folgen. Deshalb betrachten wir die Dynamiken immer nach diskreten Zeitschritten $\Delta\tau$ Hier kommt die *Leapfrog-Diskretisierung* zum Einsatz. Bei der Leapfrog-Diskretisierung wird zunächst ein diskreter Zeitschritt der Größe τ als Halbschritt für ρ_i vollzogen. Anschließend wird ein ganzer Schritt für θ_i durchgeführt, gefolgt von einem weiteren Halbschritt für ρ_i:

1. $\rho_i^{(\tau+\frac{\epsilon}{2})} = \rho_i^{(\tau)} - \frac{\epsilon}{2}\frac{\partial E(\theta)}{\partial \theta^{(\tau)}}$
2. $\theta_i^{\tau+\epsilon} = \theta_i^{\tau} + \epsilon\frac{\rho_i^{(\tau+\frac{\epsilon}{2})}}{m_i}$
3. $\rho_i^{(\tau+\epsilon)} = \rho_i^{(\tau+\frac{\epsilon}{2})} - \frac{\epsilon}{2}\frac{\partial E(\theta)}{\partial \theta^{(\tau+\epsilon)}}$

Algorithmus 2 zeigt die Leapfrog-Diskretisierung für L Leapfrog-Schritte mit $\epsilon > 0$.

Algorithmus 2 : Leapfrog Diskretisierung vgl. Neal [Nea96], Seite 56

Input : Starting point (θ, ρ), stepsize $\epsilon > 0$ and number of steps $L \geq 1$.

$\rho \leftarrow \rho_i - \frac{\epsilon}{2} \frac{\partial E(\theta)}{\partial \theta_i} \ \forall i$

for $j = 0$ *to* L **do**

  $\theta_i \leftarrow \theta_i + \epsilon \rho_i \ \forall i$

  $\rho \leftarrow \rho_i - \frac{\epsilon}{2} \frac{\partial E(\theta)}{\partial \theta_i} \ \forall i$

$\rho \leftarrow \rho_i - \frac{\epsilon}{2} \frac{\partial E(\theta)}{\partial \theta_i} \ \forall i$

Der vollständige Hybrid-Monte-Carlo-Algorithmus, einschließlich des Gibbs-Sampling für die Impulsvariablen, wird in Algorithmus 3 beschrieben. Die Kombination von stochastischen Vorgängen und Grundlagen des Metropolis-Hastings-Algorithmus führt zu einer besseren Durchquerung des Zustandsraums.

Algorithmus 3 : Hybrid-Monte-Carlo vgl. Neal [Nea96], Seite 56

Initialize $\theta^{(0)}$

$t \leftarrow 0$

while $t < \tau$ **do**

  Sample $p\rho_i^{(t)}$ from $\mathcal{N}(0, m_i) \forall i$

  Carry out L leapfrog steps with stepsize ϵ starting at state $(\theta^{(t)}, \rho^{(t)})$. Store resulting candidate state in $(\overline{\theta}, \overline{\rho})$.

  $\alpha \leftarrow min(1, exp(-(H(\overline{\theta}, \overline{\rho}) - H(\theta^{(t)}, \rho^{(t)}))))$

  if $U \leq \alpha$ **then**

    $\theta^{(t+1)} \leftarrow \overline{\theta}$ `/* new point` $\overline{\theta}$ `accepted */`

  else

    $\theta^{(t+1)} \leftarrow \theta^{(t)}$ `/* new point` $\overline{\theta}$ `rejected */`

  Output : $\theta^{(t+1)}$

  $t \leftarrow t + 1$

Der folgende Abschnitt vergleicht am Beispiel der bananenförmigen Funktion $N(y|wx, \sigma)N(x|w, 1)$ den Metropolis-Hastings- mit dem Hybrid-Monte-Carlo-Algorithmus und zeigt dessen Vorteile auf.

Die bananenförmigen Funktion $\mathcal{N}(y|wx, \sigma)\mathcal{N}(x|w, 1)$

Am Beispiel der bananenförmigen Funktion $\mathcal{N}(y|wx, \sigma)\mathcal{N}(x|w, 1)$, einer zweidimensionalen Normalverteilung $\mathcal{N}$ sieht man sehr schön, welche Vorteile der Hybrid-Monte-Carlo-Algorithmus gegenüber dem Metropolis-Hastings-Algorithmus hat. So sieht man, wie sich beim Hybrid-Monte-Carlo-Algorithmus (Abbildung 3.3(a)) schon nach nur 100 Iterationen die charakteristische Bananenform der Funktion einstellt, während beim Metropolis-Hastings-Algorithmus (Abbildung 3.3(b)) noch nichts zu erkennen ist.

Das bedeutet, dass der Hybrid-Monte-Carlo-Algorithmus deutlich schneller zur gewünschten Zielfunktion konvergiert. Dies ist ein entscheidender Vorteil bei dieser Anwendung, da wie sich zeigt, die Laufzeit ein kritischer Faktor bei komplexeren Funktionen ist. Dieser Vorteil wird aber durch die benötigten Leapfrog-Schritte leider ins Gegenteil gekehrt. So wurde in diesem Beispiel mit 1000 Leapfrog-Schritten gerechnet, was eine deutliche Erhöhung der Laufzeit zu folge hatte. Aus diesen Gründen habe ich mich in dieser Arbeit für den Hybrid-Monte-Carlo-Algorithmus entschieden.

4 Relevanzbestimmung von Genen und Überlebenszeitvorhersage

Ein grundlegendes Ziel dieser Diplomarbeit ist es, Gene zu identifizieren, die einen signifikanten Einfluss auf die Überlebenszeit von Patienten mit Krebserkrankungen haben. Hierzu wird das Cox-Regressionsmodell benutzt um die Überlebenszeiten möglichst genau zu beschreiben. Ein entscheidender Schritt ist die Schätzung und Optimierung der Regressionsparameter.

Im folgenden werde ich das verwendete Regressionsmodell (Abschnitt 4.1), die Vorgehensweise bei der Schätzung der Regressionsparameter (Abschnitt 4.2) sowie die Vorhersage der Überlebenszeiten (Abschnitt 4.4) beschreiben.

4.1 Das Cox Regressionsmodell

Die Cox-Regression ist ein statistisches Regressionsmodell welches zur Modellierung von Überlebenszeiten und Ausfallzeiten benutzt wird, [Cox72]. Es basiert auf dem Prinzip der Hazardrate oder Hazardfunktion. Basis für das Regressionsmodell sind n-dimensionale Einflußvektoren die beobachtet werden können. In dieser Arbeit bestehen die Einflussvektoren aus den Genexpressionsdaten der einzelnen Patienten.

Überlebenszeitmodelle werden typischerweise auf drei verschiedene Arten beschrieben, die *Survivor-Funktion*, die *Wahrscheinlichkeitsdichtefunktion* und die *Hazardfunktion* oder *Hazardrate* (vergleiche hierzu Kalbfleisch [KR80]).

Der Vektor $x = (x_1, ..., x_n)$ beschreibt in allen drei Gleichungen die Genexpressionsdaten der Patienten. $\theta = (\theta_1, ..., \theta_n)$ ist der Vektor der Regressionsparameter, die es zu bestimmen gilt. Die Regressionsparameter θ entsprechen in unserem Fall den Gewichten der einzelnen Gene. Dies bedeutet, wie stark der Einfluss eines einzelnen Gewichtes auf die Überlebenszeit des Patienten ist. Wobei $x_i \ll 0$ *Verlängerung* und $x_i \gg 0$ *Verkürzung* der Lebenszeit bedeutet.

4.1.1 Hazardfunktion

Die Hazardfunktion (Gleichung (4.1)) beschreibt das eigentliche Cox-Modell bestehend aus dem Einflussvektor x sowie den Regressionsparametern (Gewichte) θ. Sie spezifiziert das Risiko, dass ein Patient,

der bis zum Zeitpunkt t überlebt hat, im Intervall $[t; t + \Delta t]$ stirbt. Wobei $\Delta t \to 0$ gilt. Man spricht hier auch von spontanem Ausfallrisiko.

$$\lambda(t|x, \theta) \;=\; \lambda_0(t)\; e^{\langle \theta, x \rangle} \tag{4.1}$$

Für die Baseline Hazardfunktion (4.2) gilt $x = 0$, wodurch die Funktion nur noch abhängig von der Zeit t ist. Sie beschreibt das Ausfallrisiko für die gesamte Population unabhängig von den individuellen Einflussvektoren.

$$\lambda(t|0, \theta) = \lambda_0(t) e^{\langle \theta, 0 \rangle} = \lambda_0(t) \tag{4.2}$$

In unserer Anwendung gehen wir von einer Stufenfunktion mit fünf Abschnitten aus. Die Stufenbreite ist $\frac{1}{5}$ der maximalen Beobachtungszeit der Patienten die im Datensatz erfasst sind. Es wurde die Stufenfunktion gewählt, da deren notwendige Integration numerisch einfach und sehr schnell zu berechnen ist. Abbildung 4.1 zeigt ein Beispiel solch einer baseline-Hazardfunktion (grün) sowie die Survivor-Funktion (rot) und die Wahrscheinlichkeitsdichtefunktion (blau) des Cox-Modells. Die Stufenhöhen der Funktion werden während des Optimierungsprozesses angepasst und sozusagen von den verfügbaren Patientendaten gelernt. Im Kapitel 4.3 gehe ich detaillierter auf den Lernprozess der Gewichte ein.

4.1.2 Wahrscheinlichkeitsdichtefunktion

Die Wahrscheinlichkeitsdichtefunktion der Überlebenszeit (Gleichung (4.3)) gibt die Wahrscheinlichkeit an, mit der ein Ereignis (z.B. Tod/Rückfall eines Patienten) zum Zeitpunkt t eintritt.

$$f(t|x, \theta) \;=\; \lambda_0(t)\; e^{\langle \theta, x \rangle}\; \exp\left[-e^{\langle \theta, x \rangle} \int_0^t \lambda_0(u)\; du \right] \tag{4.3}$$

Im Unterschied zur Hazardfunktion muss nicht bekannt sein ob das Ereignis eingetreten ist oder nicht. Die Dichtefunktion gibt somit zu jedem Zeitpunkt t die Eintrittswahrscheinlichkeit des Ereignisses an.

4.1.3 Survivor-Funktion

Die Survivor-Funktion (Gleichung (4.4)) beschreibt den Fall, dass ein Ereignis bis zum Zeitpunkt t noch nicht eingetreten ist. Auf die Überlebenszeitanalyse bezogen heißt es, dass ein Patient bis zum Zeitpunkt t überlebt hat. So kann durch Integration der Dichtefunktion (4.3) gezeigt werden, dass die Survivor-Funktion wie folgt dargestellt werden kann:

$$F(t|x, \theta) \;=\; \exp\left[-\int_0^t \lambda_0(u)\; du \right]^{\exp\langle \theta, x \rangle} \tag{4.4}$$

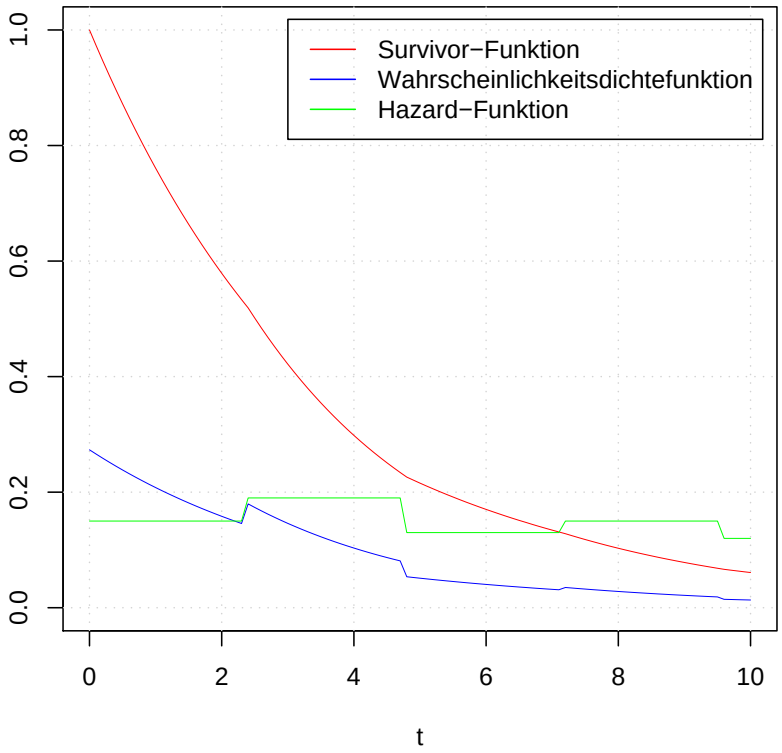


Abbildung 4.1: Ein Beispiel für die baseline-Hazardfunktion (grün) sowie die Survivor-Funktion (rot) und die Wahrscheinlichkeitsdichtefunktion (blau) des Cox-Modells.

4.2 Schätzung und Optimierung der Regressionsparameter

Um den Regressionsparametervektor θ zu schätzen und zu optimieren, geht man wie in [KR80] beschrieben vor und bildet aus den partiellen Likelihood-Funktionen der zensierten und der unzensierten Daten einen gemeinsamen Likelihood. Der Datensatz eines Patienten gilt dann als zensiert, wenn der Zeitpunkt des interessanten Ereignisses (Tod) unbekannt ist. Dies ist in klinischen Studien der Fall, wenn ein Patient frühzeitig aus der klinischen Studie austritt oder die Studie endet. Ist dies der Fall, ist bei diesen Patienten nicht der Todeszeitpunkt bekannt, sondern die Untergrenze der Überlebenszeit, bis zu der der Patient überlebt hat.

4.2.1 Die Likelihood-Funktion

Durch Kombination der Survivor-Funktion (Gleichung (4.4)) und der Wahrscheinlichkeitsdichtefunktion (Gleichung 4.3) erhält man die Likelihood-Funktion des zugrunde liegenden Datensatzes.

$$L(\theta) \propto \prod_{j=1}^{L} f(t^{(j)}|x^{(j)},\theta)^{1-\delta^{(j)}}\ F(t^{(j)}|x^{(j)},\theta)^{\delta^{(j)}} \tag{4.5}$$

Der Parameter $\delta^{(j)}$ ($\delta^{(j)} \in \{0,1\}$) dient zur Kennzeichnung, ob die Überlebenszeit von Patienten i zensiert ist oder nicht. Ist die Überlebenszeit eines Patienten zensiert, bedeutet dies, dass bei diesem Patienten nur eine Dauer bekannt ist in der er nicht gestorben ist, also seine minimale Überlebenszeit.

$$\delta^{(j)} = \begin{cases} \delta^{(j)} = 1 & \text{Daten zensiert} \\ \delta^{(j)} = 0 & \text{Daten nicht zensiert} \end{cases} \tag{4.6}$$

Dies stellt sicher, dass zensierte Patienten nur durch die Survivor-Funktion in die Likelihood-Funktion eingehen, da der Einfluss der Wahrscheinlichkeitsdichtefunktion durch den Parameter $\delta^{(j)} = 1$ neutralisiert werden. Dies ist nötig, da für die zensierten Patienten der Zeitpunkt des Todes nicht bekannt ist, jedoch die minimale Überlebenszeit. Genau umgekehrt verhält es sich mit den nichtzensierten Patienten. Diese beeinflussen die Likelihood-Funktion nur durch die Wahrscheinlichkeitsdichtefunktion.

4.2.2 Die a-priori-Verteilung

Um später mit Hilfe des Bayes' Theorem auf den Parameter rückzuschließen, verwenden wir eine a-priorie Verteilung um die Regressionsparameter auszudünnen. Da der zu optimierende Regressionsparametervektor hochdimensional ist nehmen wir an, dass nur sehr wenige Gene relevant für die Überlebenszeitvorhersage sind, d.h. dass nur sehr wenige Regressionsparameter deutlich ungleich 0 ($|\theta_n| \ll 0$) sein sollen. Um dies zu gewährleisten, benötigen wir eine a-priori-Verteilung $p(\theta)$ der Regressionsparameter, die den Fall, dass wenige Gene deutlich ungleich 0 sind, dem Fall, dass viele Gene deutlich ungleich 0, vorzieht.

Um dies zu erreichen, wurde folgende Warhscheinlichkeitsdichtefunktion verwendet:

$$p(\theta) \propto \exp\left[-\frac{1}{q \cdot s^q} \cdot |x|^q\right] \tag{4.7}$$

Für einen Wert von $q = 1$ erhält man die Laplace-Verteilung sowie die Normalverteilung für einen Wert $q = 2$. Bei dieser Anwendung wird die a-priori-Verteilung mit $q \leq 1$ parametrisiert.

Für den 2-dimensionalen Fall ($\theta = \{\theta_1, \theta_2\}$) verhält sich die Gleichung wie in Abbildung 4.2 zu sehen ist. Man sieht hier deutlich, dass sich die Wahrscheinlichkeiten 0 annähern, wenn sich mindestens einer der beiden Parameter dem Wert ± 1 nähert.

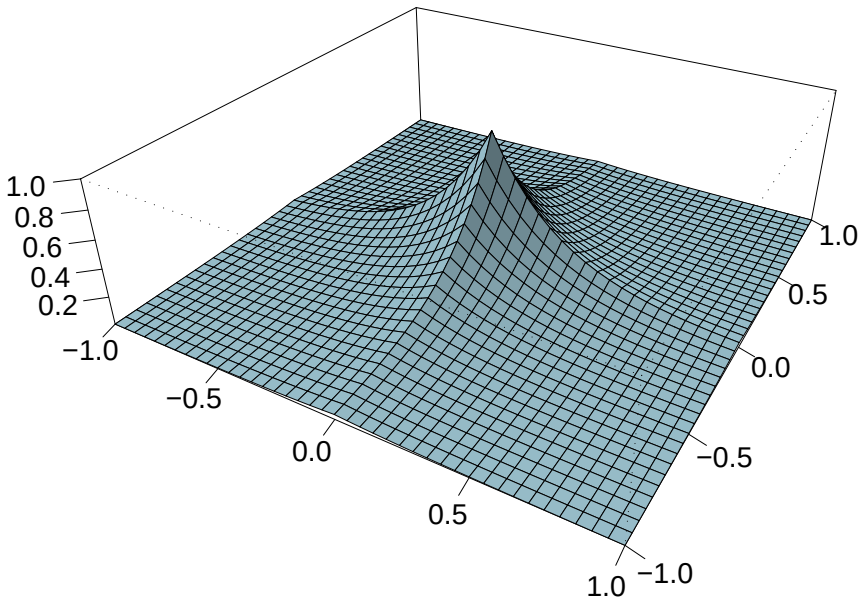


Abbildung 4.2: $p(\theta)$ für den 2-dimensionalen Fall, mit den Parametern $q = 1.0$ und $s = 0.3$

4.2.3 Schätzung des Parametervektors

Um den Parametervektor der Cox-Regression zu schätzen, wird nun die Verteilung über die Regressionsparameter betrachtet. Diese Verteilung $p(\theta|D)$ kann mit Hilfe des Bayes'schen Theorems berechnet werden.

- D_y = Überlebenszeiten,
- D_x = unklassifizierte Genexpressionsdaten und
- D = Trainingsdaten bestehende aus D_x und D_y

Wendet man nun das Bayes'sche Theorem an, erhält man:

$$p(\theta|D) = \frac{p(D_y|D_x,\theta)p(D_x|\theta)p(\theta)}{p(D)} \tag{4.8}$$

Unter der Annahme, dass die Genexpressionsdaten D_x statistisch unabhängig von den Regressionsparametern θ sind ($p(D_x|\theta) = p(D_x)$), ergibt sich im nächsten Schritt:

$$p(\theta|D) = \frac{p(D_x)}{p(D)} \cdot p(D_y|D_x,\theta) \cdot p(\theta) \tag{4.9}$$

Als a-priori-Wahrscheinlichkeitsverteilung für den Parametervektor θ verwende ich die in Abschnitt 4.2.2 beschriebene Wahrscheinlichkeitsverteilung. Da die Wahrscheinlichkeitsdichtefunktionen $p(D_x)$ und $p(D)$ unabhängig von θ sind, können diese vernachlässigt werden. Damit ergibt sich nach weiterer Vereinfachung:

$$p(\theta|D) \propto p(D_y|D_x,\theta)p(\theta). \tag{4.10}$$

Die Gleichung $p(D_y|D_x,\theta)$ entspricht unserer Likelihood-Funktion 4.5. Setzt man nun die Gleichungen (4.7) und (4.5) in die vereinfachte Form des Bayes'schen Theorems (4.10), so erhält man die gewünschte a-posteriori-Wahrscheinlichkeitsverteilung über die Regressionsparameter θ in Abhängigkeit von den Trainingsdaten D.

Nun setzt man die a-priori-Wahrscheinlichkeiten (Gleichung 4.7) für $p(\theta)$ ein und multipliziert diese über die einzelnen Regressionsparameter θ. Anschließend ersetzt man die Gleichung $p(D_y|D_x,\theta)$ durch die kumulierte Likelihood-Funktion (Gleichung (4.5)) und erhält nach dadurch Gleichung (4.11).

$$p(\theta|D) = \prod_{i=1}^{Q} p(\theta^{(i)}) \prod_{j=1}^{L} f(t^{(j)}|x^{(j)},\theta)^{1-\delta^{(j)}} F(t^{(j)}|x^{(j)},\theta)^{\delta^{(j)}} \tag{4.11}$$

Die Verteilung 4.11 wird nun benutzt, um von ihr mit Hilfe des Hybrid-Monte-Carlo-Algorithmus (Abschnitt 3.3.2) Stichproben zu ziehen, die dann einen Aufschluss über den Regressionsparametervektor θ geben.

4.3 Integration des Hybrid-Monte-Carlo-Algorithmus

Um unsere Maximum-Posterori-Funktion $p(\theta|D)$ mit dem Hybrid-Monte-Carlo-Algorithmus zu verwenden, muss sie in den negativen logarithmischen Raum überführt werden. DAs Gleichungsystem (4.12) zeigt die einzelnen Schritte, um die Form $-\log[p(\theta|D)]$ zu erhalten.

$$
\begin{aligned}
-\log\left[p(\theta|D)\right] &= -\log\left[\prod_{i=1}^{Q} p(\theta^{(i)}) \prod_{j=1}^{L} f(t^{(j)}|x^{(j)},\theta)^{1-\delta^{(j)}} F(t^{(j)}|x^{(j)},\theta)^{\delta^{(j)}}\right] \\
&= \sum_{i=1}^{Q}\left[\frac{1}{qs^q}\left|\theta_i\right|^q\right] + \sum_{j=1}^{L} -\log\left[f(t^{(j)}|x^{(j)},\theta)^{1-\delta^{(j)}}\right] - \log\left[F(t^{(j)}|x^{(j)},\theta)^{\delta^{(j)}}\right] \\
&= \sum_{i=1}^{Q}\left[\frac{1}{qs^q}\left|\theta_i\right|^q\right] + \sum_{j=1}^{L} -\log\left[f(t^{(j)}|x^{(j)},\theta)\right](1-\delta^{(j)}) - \log\left[F(t^{(j)}|x^{(j)},\theta)\right](\delta^{(j)})
\end{aligned}
\tag{4.12}
$$

Des weiteren benötigen wir die partiellen Ableitungen nach den Regressionsparametern θ_n, um sie bei der Diskretisierung des Leapfrog-Algorithmus zu verwenden. Gleichung (4.13) zeigt die partielle Ableitung nach θ_i.

$$
\frac{\partial p(\theta|D)}{\partial \theta_i} = s^{-q}|\theta_i|^q \frac{\text{signum}(|\theta_i|)}{|\theta_i|} \sum_{j=1}^{L}(\delta^{(j)})(\int_0^t \lambda_0(t)dt\,\langle\theta_j, x_j\rangle\, x_{ij}) - (1-\delta^{(j)})x_{ij} \tag{4.13}
$$

Die folgenden Abschnitte zeigen zwei Erweiterungen des Hybrid-Monte-Carlo-Algorithmus , die in dieser Anwendung sich als sehr nützlich erwiesen haben. Zum Einen ein sich veränderndes ϵ und zum Anderen die selbst lernende Schritthöhe für unsere Baseline-Stufenfunktion.

4.3.1 Gamma-Verteilung für ϵ

Um eine möglichst gute Traversion durch den Funktionsraum zu erreichen, ist es hilfreich, bei der Diskretisierung des Hybrid-Monte-Carlo-Algorithmus verschieden große Schritte zu gehen. Dies wurde realisiert, indem die Schrittgröße ϵ zufällig von einer Gammaverteilung gezogen wird. Gleichung (4.14) zeigt die Gamma-Funktion, die Teil der Gammaverteilung ist. Sie ist eine Erweiterung der Fakultätsfunktion auf alle positiven reellen Zahlen. Die Gammaverteilung 4.15 selber ist eine Verallgemeinerung der Exponential-Verteilung und damit eine kontinuierliche Wahrscheinlichkeitsverteilung.

$$
\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\mathrm{d}t \tag{4.14}
$$

$$
f(x) = \begin{cases} \frac{b^p}{\Gamma(p)}x^{p-1}e^{-bx} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{4.15}
$$

Abbildung 4.3 zeigt ein Beispiel für eine Gammaverteilung. Es wird eine Gammaverteilung benutzt, da die gezogenen Zufallswerte nur positiv sein können, was für den Hybrid-Monte-Carlo-Algorithmus eine notwendige Vorraussetzung ist.

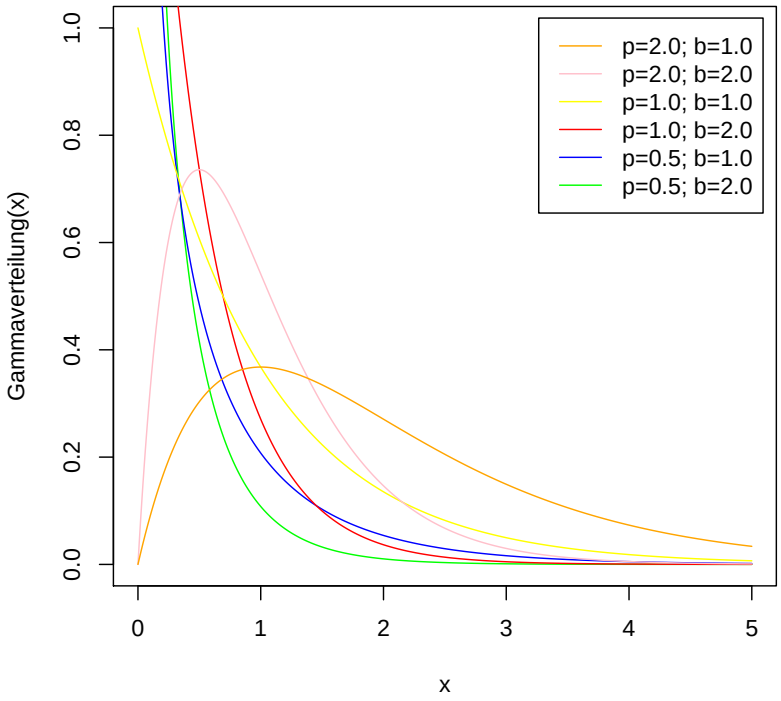


Abbildung 4.3: Beispiel für die Wahrscheinlichkeitsdichte der Gammaverteilung mit verschiedenen Parametern

4.3.2 Gamma-Verteilung für das Ziehen der baseline-Hazard-Schritthöhen

Da wir in unserem Regressionsmodell die Schritthöhen der baseline-Hazard-Funktion ebenso als Regressionsparameter ansehen, müssen sie nach dem Hybrid-Monte-Carlo-Algorithmus ebenso generiert werden. Um nur Schritthöhen von $h > 0$ zu erhalten, werden sie ebenso von einer Gammaverteilung gezogen. Abbildung 4.4 zeigt ein Beispiel der Gammaverteilung von der die Schritthöhen zufällig gezogen wurden. Hier kann man sehr gut sehen das sich die meisten gezogenen Werte im Bereich von 0.0 bis 0.2 befinden. Die sich anpassenden Schritthöhen haben den großen Vorteil, dass sie das spontane Ausfallrisiko $\lambda_0(t)$ während der Beobachtungszeit genauer modellieren und sozusagen durch den verwendeten Datensatz gelernt werden und sich durch den Hybrid-Monte-Carlo-Algorithmus selbständig anpassen.

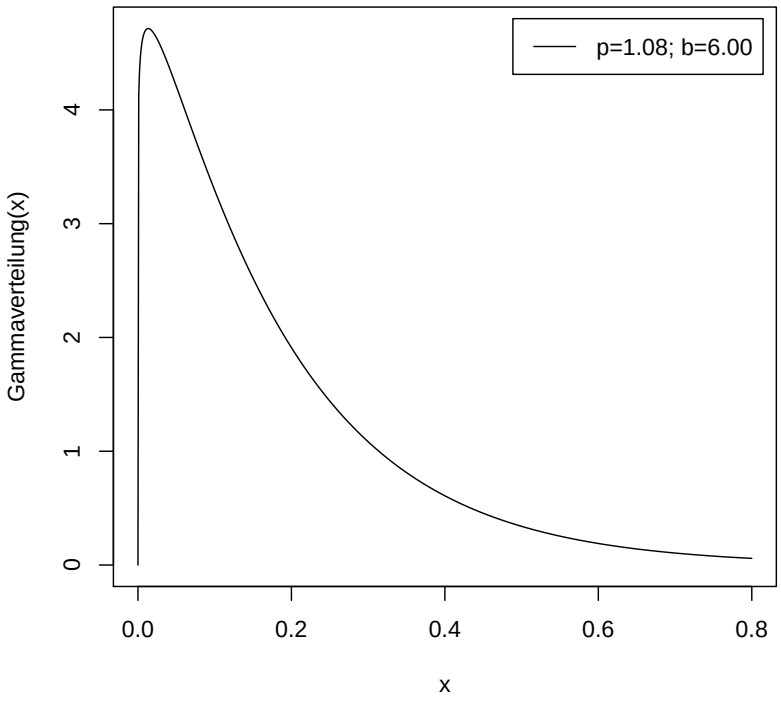


Abbildung 4.4: Ein Beispiel für die Gammaverteilung aus der die Schritthöhen des Baseline Hazard gezogen werden.

4.4 Verteilung der Überlebenszeiten

Um nun eine Verteilung der Überlebenszeiten gemäß den von der Markovkette generierten Regressionsparameter zu erhalten, benützt man die Regressionsparameter eines jeden Schrittes der Markovkette um damit einen Wert für die Überlebenszeit zu erzeugen. Die Überlebenszeiten wurden mit Hilfe des Acceptance-Rejection-Verfahren von der Wahrscheinlichkeitsdichtefunktion des Cox-Modells (Gleichung (4.16)) generiert. Das verwendete Acceptance-Rejection-Verfahren wird im Anhang A.1 näher erläutert.

$$f(t|x,\theta) \;=\; \lambda_0(t)\, e^{\langle\theta,x\rangle}\; \exp\left[-e^{\langle\theta,x\rangle} \int_0^t \lambda_0(u)\, du\right] \tag{4.16}$$

Man erhält eine Verteilung über die Überlebenszeiten eines jeden Patienten. Diese kann anschließend mit den tatsächlichen Überlebenszeiten verglichen werden und es können Parameter der Wahrscheinlichkeitsverteilung wie Standardabweichung und Varianz berechnet werden.

5 Ergebnisse

Um die grundlegende Funktionalität der vorgestellten Methode zu zeigen, wurde zunächst der simulierte Datensatz *SIMULATE* untersucht. In den folgenden Abschnitten werde ich die Attribute des Datensatzes beschreiben (Abschnitt 5.1) und anschließend die gewonnenen Ergebnisse (Abschnitt 5.1.1 & 5.1.2) zeigen. Abschnitt 5.2 beschreibt abschließend die Erkenntnisse bei der Auswertung des *NEUROBLASTOM* Datensatzes.

Da das Sampling mit Hilfe des Metropolis-Hastings-Algorithmus keine verwertbaren Ergebnisse erbrachte und die Markovketten zu keinem Zeitpunkt konvergierten, soll in dieser Arbeit nur auf die Ergebnisse des Hybrid-Monte-Carlo-Algorithmus eingegangen werden. Das Versagen des Metropolis-Hastings-Algorithmus lässt sich durch die Hochdimensionalität und Komplexität der Funktion sowie des bereits beschriebenen *random walk behaviors* des Algorithmus erklären.

5.1 Der *SIMULATE* Datensatz

Um ein proof-of-concept zu haben, wurde ein Datensatz mit vereinfachten Merkmalen erstellt. Er besteht aus ähnlich vielen Patienten sowie Genexpressionsmessungen wie sie bei einem realen Datensatz zu erwarten sind, enthält aber deutlich weniger Rauschen.

Um die Funktionalität des Analyseansatzes zu überprüfen, wurden die Genexpressionsdaten vereinfacht. Bei der Simulation der Genexpressionsdaten wird davon ausgegangen, dass nur das erste gemessene Gen einen Einfluss auf die Überlebenszeit hat. Der simulierte Genexpressionswert x_1 der den Einfluss des ersten Gens bestimmt, wurde normal verteilt mit dem Erwartungswert 0.6 sowie -0.6 und der Varianz 0.5 generiert. Die Genexpressionswerte für die Gene $2, ..., 7400$ wurden normalverteilt mit dem Erwartungswert 0 und der Varianz 0.5 generiert. Die Überlebenszeiten wurden nach der Wahrscheinlichkeitsdichteverteilung des Cox-Regressions-Modells (Gleichung (4.3)) simuliert. Als Parametervektor wurde $\theta = (1, 0, ..., 0)$ und $\lambda_{Cox} = 0.18$ angenommen. Die Patienten unterliegen einer gleich verteilten Zensur in dem Zeitintervall $[0, 10]$ Jahre. Sollte der Zeitpunkt der Zensur auftreten bevor der Patient stirbt, so wird der Patientendatensatz als *zensiert* markiert und der Zeitpunkt der Zensur gespeichert. Andernfalls wird der Zeitpunkt des Todes gespeichert. Von den 240 simulierten Patienten sind 124 Patienten der Zensur unterworfen, was einem Anteil von 51.67 % entspricht. Werte in dieser Größenordnung sind auch bei reellen Datensätzen zu beobachten.

Der Datensatz wurde dann geteilt in einen Trainingsdatensatz und einen Validierungsdatensatz. Der Trainingsdatensatz enthält 160 Patienten und der Validierungsdatensatz die verbleibenden 80 Patienten.

Bei den nachfolgenden Ergebnissen wurde mit dem Trainingsdatensatz gearbeitet, um eine spätere Validierung mit dem Validierungsdatensatz zu ermöglichen.

5.1.1 Konvergenz der Markovkette

Um festzustellen ob eine Markovkette konvergiert ist, gibt es keine allgemein gültige Vorgehensweise. Um zu ermitteln wann eine Markovkette ausreichend konvergiert ist, wurden mehrere Markovketten mit unterschiedlichen Initalwerten gestartet und verglichen, nach wie vielen Iterationen sich die erzeugten Werte annäherten. In unserem Anwendungsbeispiel hat die Erfahrung gezeigt, dass nach etwa $300 - 800$ Schritten der Markovkette davon ausgegangen werden kann, dass diese ausreichend konvergiert ist.

Es ist nicht möglich auf Grund der Akzeptanzrate auf die Güte der Markovkette zu schließen. So wurden die besten Ergebnisse mit dem *SIMULATE* Datensatz mit Ketten erreicht, die eine Akzeptanzrate zwischen 12% und 35% aufwiesen.

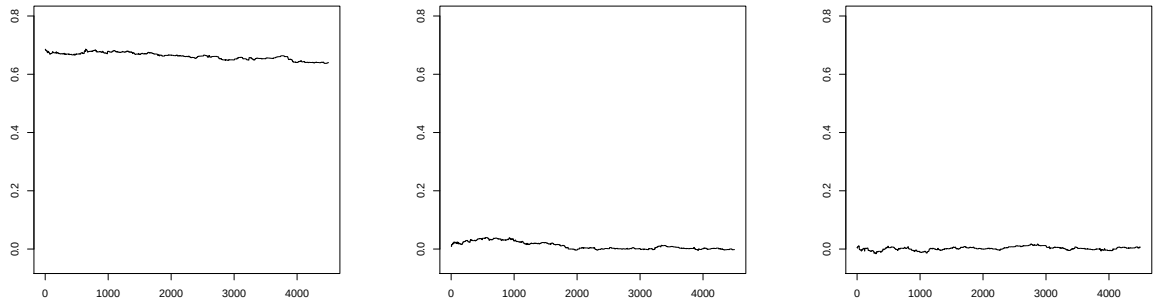
Für die a-priori-Verteilung (Gleichung (4.7)) wurden die Parameter $q = 1.0$ und $s = 0.1$ verwendet. Es wurden 500 Leapfrogschritte (L) ausgeführt und 5000 Iterationen durchgeführt, wobei ein *burn-in* von 500 Schritten verwendet wurde und somit die ersten 500 Schritten verworfen wurden. Die Werte für die Schrittgröße ϵ der Diskretisierung wurden von einer Gammaverteilung (Gleichung (4.15)) mit den Parametern $b = 2$ und $p = 2.0 \times 10^{-4}$ gezogen. Die gezogenen Werte von ϵ haben ihren Mittelwert bei $\bar{\epsilon} = 1.0 \times 10^{-4}$ mit einer Varianz $\sigma^2 = 5.0 \times 10^{-5}$. Als Initialwert wurde 0 für alle Regressionsparameter der Gene gewählt sowie 0.18 für die initialen Werte der Baseline-Hazard-Stufen.

Die Wahl der Schrittgröße ϵ hat sich als die entscheidende Schwierigkeit herausgestellt. So führt ein zu großes ϵ zu einer schnellen Konvergenz (oft in den ersten 20 Schritte), darüber hinaus werden aber annähernd alle folgenden Schritte zurückgewiesen da die Kette sozusagen aus ihrem *Optimum* heraus springt. Ein zu klein gewähltes ϵ führt dazu, dass die Kette niemals konvergiert und die gesampelten Werte für die Regressionsparameter sich nicht signifikant von ihrem Initialwert 0 entfernen. Neal et al. [Nea96] schlug in seiner Arbeit einen variables ϵ vor. Dies wurde bereits durch die zufällig gezogenen Werte nach der Gammaverteilung für ϵ berücksichtigt. In seiner Arbeit geht Neal noch einen Schritt weiter und schlägt vor, dass sich das ϵ mit der Laufzeit der Kette reduziert. Dies erlaubt dann in einem nahezu optimalen *Ort* im Funktionsraum eine Feinabstimmung. Diese Idee wurde implementiert, indem man die Parameter der Gammaverteilung für ϵ nach dem Burn-In so verändert, dass die gezogenen Werte etwa um den Faktor $5 - 20$ kleiner sind als die Werte für ϵ zu Beginn der Markovkette.

Diese Veränderung führte zu deutlich besseren Ergebnissen der Markovkette wie in den Abbildungen 5.1 zu sehen ist. Eine Aufstellung der verwendeten Parameter befindet sich im Anhang B.2.

Bei vorherigen Markovketten war die Varianz der gezogenen Werte sehr hoch. Um dieses übermäßige springen der Werte zu verringern, wurde außerdem die Massen der Hamiltonischen Energiefunktion reduziert, was ein ruhigeres und stationäreres Verhalten zur Folge hatte.

Die Abbildung 5.1 zeigt den Verlauf der Markovkette für die Regressionsparameter θ_1 bis θ_1. Hier wird deutlich, dass θ_1 sich wie erwartet nach dem Burn-In deutlich vom Ursprung entfernt hat. Hingegen verbleiben θ_2 und θ_3 nahe bei seinem zu erwartenden Wert $\theta_2 = \theta_3 = 0$ und bewegen sich um den Ursprung.



(a) Regressionsparameter θ_1 (b) Regressionsparameter θ_2 (c) Regressionsparameter θ_3

Abbildung 5.1: Verlauf der von der Markovkette gezogenen Werte für θ_1 (a), θ_2 (b) und θ_3 (c). Es ist deutlich zu sehen, dass sich der Parameter θ_1 deutlich von den anderen abhebt und seinen Mittelwert nahe $\theta_1 \sim 0.7$ hat.

5.1.2 Bestimmung der Regressionsparameter

Abbildung 5.2 zeigt den Boxplot der Regressionsparameter θ_1 bis θ_{10}. Man sieht hier klar, dass sich der Parameter θ_1 deutlich von den anderen abhebt und seinen Mittelwert nahe $\theta_1 \sim 0.7$ hat. Dass der errechnete Mittelwert nicht exakt dem Erwarteten entspricht, kann auf den Einsatz der a-priori-Verteilung zurückgeführt werden. Die a-priori-Verteilung *bestraft* Werte deutlich ungleich 0 für die Regressionsparameter und *drückt* somit den Wert in Richtung $\theta_1 \to 0$. Die gesampelten Werte der Regressionsparameter θ_2 und θ_3 befinden sich um 0, wie es zu erwarten war.

In Abbildung 5.2 sieht man sehr deutlich, dass die gezogenen Werte eine sehr kleine Standardabweichungen aufweisen. Tabelle 5.1 zeigt die berechneten Mittelwerte und Standardabweichung (σ) der Regressionsparameter θ_1 bis θ_{10}.

Für die Parameter θ_{11} bis θ_{7400} zeigt sich ein ähnliches Verhalten wie für θ_2 bis θ_{10}. So konnte gezeigt werden, dass das Verfahren auf dem simulierten Datensatz angewendet das relevante Gen zuverlässig bestimmt hat.

Für das Sampling der Baseline-Hazard wurde eine höhere Masse für die Hamiltonischen Energiefunktion gewählt. Dies führt, wie in Abbildung 5.3 zu sehen ist, zu einem etwas unruhigerem Verhalten.

Zudem fällt auf, dass sich ausnahmslos alle fünf Intervallhöhen der Baseline-Hazard Funktion verringert haben. Dies könnte ein Zeichen für die verbesserte Anpassung der gewählten Stufenfunktion

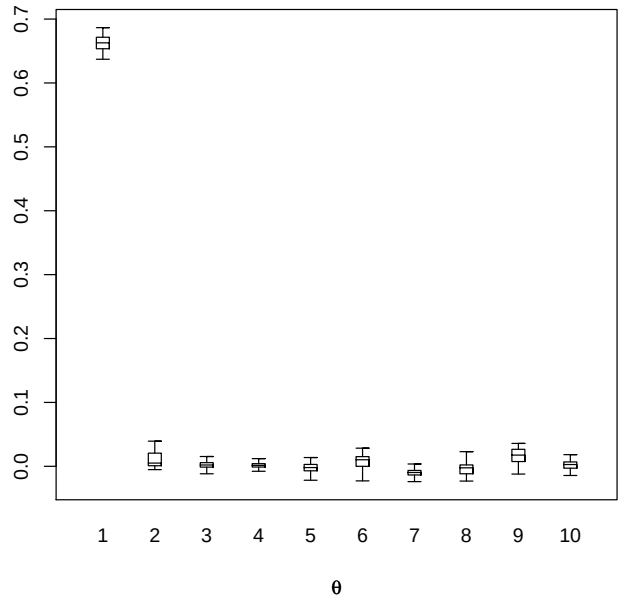


Abbildung 5.2: Boxplot der Regressionsparameter θ_1 bis θ_{10}.

Tabelle 5.1: Mediane und Standardabweichungen (σ) der Regressionsparameter θ_1 bis θ_{10}

	Median	σ
θ_1	0.6621	0.01192
θ_2	0.0113	0.01269
θ_3	0.0019	0.00581
θ_4	0.0020	0.00466
θ_5	-0.0038	0.00900
θ_6	0.0058	0.01310
θ_7	-0.0105	0.00670
θ_8	-0.0036	0.01001
θ_9	0.0163	0.01120
θ_{10}	0.0684	0.00715

an die Trainingsdaten sein. So zeigt Abbildung 5.4 das eine eine Survivor-Funktion mit einer gereingeren Baseline-Hazard (rot) besser die tatsächlich Survivor-Funktion der *SIMULATE* Daten modelliert.

Um die Güte einer Kette zu bestimmen wurden in dieser Arbeit, wie in Kapitel 4.4 beschrieben die von ihr gesampelten Überlebenszeiten herangezogen. So ist es möglich, die Fehler der vorhergesagten Überlebenszeiten zu berechnen, um so Rückschlüsse auf die Güte der Markovkette zu ziehen.

Zu Verifizierung wurden weitere Ketten mit unterschiedlichen Parametern gestartet, die alle ähnliche

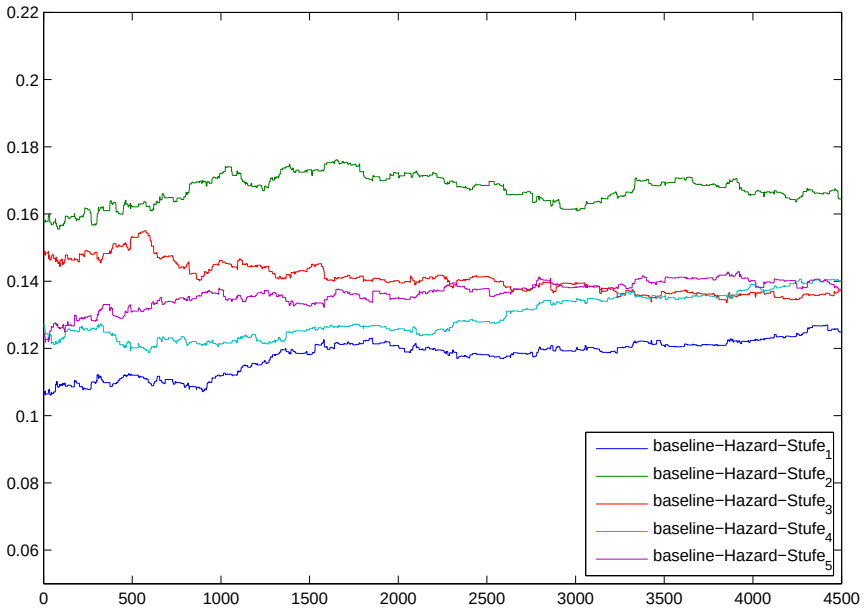


Abbildung 5.3: Verlauf der Baseline-Hazard Werte.

Ergebnisse für die Konvergenz der Markovkette, sowie das anschließende Sampling der Überlebenszeiten erzielten. Ketten mit $10000 - 20000$ Schritten zeigten, dass die Markovketten auch bei größeren Schrittzahlen stabil bleiben. Eine Verbesserung der Ergebnisse durch die Erhöhung der Schrittzahl konnte nicht nachgewiesen werden.

Im folgenden Abschnitt 5.2 wird das Verfahren auf den in Abschnitt 5.2.1 beschriebenen *NEUROBLASTOM* Datensatz angewendet um Gene zu identifizieren, die einen signifikanten Einfluss auf die Überlebenszeit des Patienten haben.

5.1.3 Validierung der Markovkette

Um die generierte Markovketten zu validieren, wird der von ihr erzeugte Regressionsparametervektor θ dazu verwendet, für die 80 Patienten des Validierungsdatensatzes Überlebenszeiten vorherzusagen.

Verteilungen über die Überlebenszeiten

Um eine Verteilung über die Überlebenszeiten der Patienten zu erhalten, wird nun für jeden Patienten i von der Überlebenszeitverteilung $p(j_i|x_i, \theta_n)$ eine Überlebenszeit für jeden Schritt der Markovkette ($k = 1 \cdots 4500$) mit Hilfe des beschriebenen Acceptance-Rejection-Verfahren erzeugt.

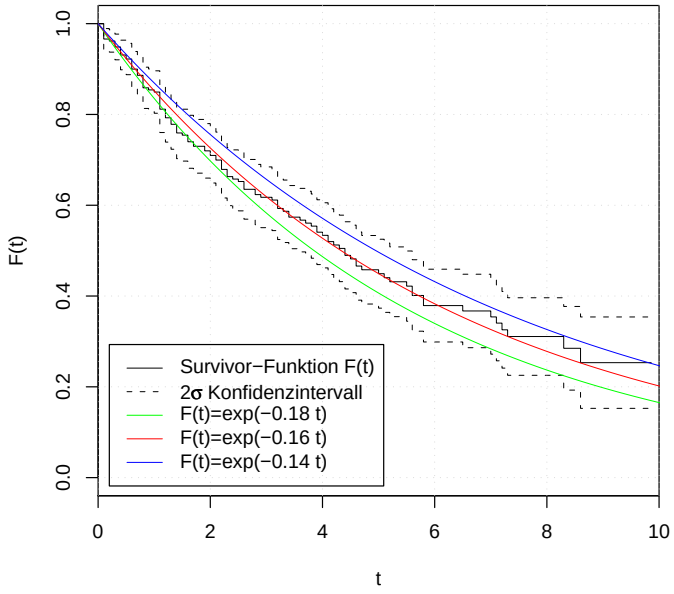


Abbildung 5.4: Darstellung der aus den *SIMULATE* Daten berechneten Survivor-Funktion sowie die Survivor-Funktionen mit geschätzter konstanter Baseline-Hazard.

Um die Verteilung über die Überlebenszeiten mit den tatsächlichen Überlebenszeiten quantitativ zu vergleichen, wird sie mit dem Median der Verteilung der Überlebenszeiten verglichen. Ein Augenmerk wird aber auch auf die Standardabweichung der Verteilung gerichtet.

Die Tabellen 5.2(a) und 5.2(b) geben einen Überblick über die vorhergesagten Überlebenszeiten. Tabelle 5.2(a) zeigt die Mediane, Standardabweichung und den Fehler der Vorhersagen. Tabelle 5.2(b) zeigt die selben Werte für Patienten, die Zensur unterworfen sind. Für zensierte Patienten kann bis auf die Patienten $5, 18, 38$ kein Fehler berechnet werden, da die tatsächliche Überlebenszeit nicht bekannt ist und der Median der vorhergesagten Überlebenszeiten immer nach dem Zeitpunkt der Zensur liegt.

Für die verbleibenden 40 Patienten wurden ähnliche Ergebnisse beobachtet. Tabelle B.2 bietet eine vollständige Übersicht der Vorhersagen des Validierungsdatensatzes.

Abbildung 5.5 stellt die vorhergesagten Überlebenszeiten gegen die tatsächlichen Überlebenszeiten bzw. die Zensurzeiten dar. Bei einer perfekten Vorhersage würden die Vorhersagen der nichtzensierten Patienten, gekennzeichnet mit einem x, den tatsächlichen Überlebenszeiten entsprechen und würden deshalb direkt auf der Diagonalen liegen.

Bei genauerer Betrachtung der Vorhersagen für unzensierte Patienten fallen insbesondere Patient 8 und 33 auf. Patient 8 fällt auf, da seine Überlebenszeitvohersage einen großen Fehler von $\Delta t_8 = 8.45$ Jahren zeigt und die Wahrscheinlichkeitsverteilung eine große Standardabweichung von $\sigma_8 = 18.5$ Jahren besitzt (Tabelle 5.2). Der Patient 33 hingegen ist ein Beispiel für den gegenteiligen Fall. Seine Vorhersage

(a) Nichtzensierte Patienten.

ID	Tatsächl.	Median	Std. Abw.	Fehler
0	3.9	2.14	3.57	1.76
1	2.2	1.28	2.01	0.92
3	8.6	5.41	7.88	3.19
6	5.5	6.36	10.74	0.86
7	1.9	2.81	5.54	0.91
8	2.3	10.75	18.5	8.45
9	5.6	10.65	16.54	5.05
10	5.8	13.21	21.2	7.41
11	0.1	4.04	6.56	3.94
14	1.4	2.8	5.1	1.4
15	5.2	6.88	14.66	1.68
19	2.2	3.36	6.32	1.16
24	1.3	4.74	6.97	3.44
26	2.2	3.07	5.14	0.87
27	2.3	2.94	4.53	0.64
30	2.1	6.1	9.48	4.0
33	2.8	2.68	5.51	0.12
35	0.8	7.19	13.34	6.39
36	1.1	2.61	4.54	1.51
40	3.9	0.82	1.55	3.08

(b) Zensierte Patienten.

ID	Tatsächl.	Median	Std. Abw.	Fehler
2	2.28	28.33	44.95	≥ 0
4	2.51	16.58	32.03	≥ 0
5	7.95	3.79	7.03	≥ 4.17
12	8.46	8.57	15.73	≥ 0
13	2.21	14.92	25.85	≥ 0
16	1.08	19.54	34.12	≥ 0
17	3.09	11.51	17.24	≥ 0
18	7.98	5.9	9.12	≥ 2.08
20	0.03	3.24	8.17	≥ 0
21	2.3	11.97	18.96	≥ 0
22	5.58	10.18	17.71	≥ 0
23	3.2	18.54	38.5	≥ 0
25	0.33	32.52	79.26	≥ 0
28	0.04	2.56	3.92	≥ 0
29	6.98	14.06	27.86	≥ 0
31	5.57	9.08	13.16	≥ 0
32	5.74	14.71	25.59	≥ 0
34	7.64	8.35	13.72	≥ 0
37	2.21	9.98	16.9	≥ 0
38	4.46	3.54	6.63	≥ 0.91

Tabelle 5.2: Zusammenfassung der ersten 20 nicht zensierten Patienten (a) und der ersten 20 zensierten Patienten (b). Die zweite Spalte zeigt. die tatsächliche Überlebenszeit bzw. bei den zensierten Patienten die Beobachtungszeit. Die Spalten drei bis fünf geben die Mediane, Standardabweichung sowie den Fehler der vorhergesagten Überlebenszeitverteilung an.

weist nur einen sehr kleinen Fehler von $\Delta t_{33} = 0.12$ Jahren auf und eine Standardabweichung von $\sigma_{33} = 5.51$ Jahren.

Stellt man nun die relativen Häufigkeiten der vorhergesagten Überlebenszeiten als Wahrscheinlichkeitsdichte dar, so ist, wie in Abbildungen 5.6(a) (Patient 8) und 5.6(b) (Patient 33) zu sehen, der Peak für den Patienten 33 deutlicher ausgeprägt als der Peak für den Patienten 8. Dies erklärt den großen Fehler bei der Vorhersage von Patient 8. So besitzt die Vorhersageverteilung offensichtlich eine große Ungenauigkeit und Streuung.

Der Vergleich der durchschnittlich vorhergesagten Überlebenszeiten zwischen zensierten und nichtzensierten Patienten zeigte das die durchschnittlich vorhergesagte Überlebenszeit der zensierten Patienten mit 10.67 Jahren deutlich über der der unzensierten Patienten mit 4.91 Jahren lag. Dies könnte auf eine Korrelation zwischen tatsächlichen Überlebenszeit und der Wahrscheinlichkeit, dass ein Patient der Zensur unterworfen ist, zurückgeführt werden. So besteht bei Patienten die sich länger in der Studie aufhalten auch eine erhöhte Wahrscheinlichkeit dafür, dass sie vor ihrem Tod aus der Studie ausscheiden.

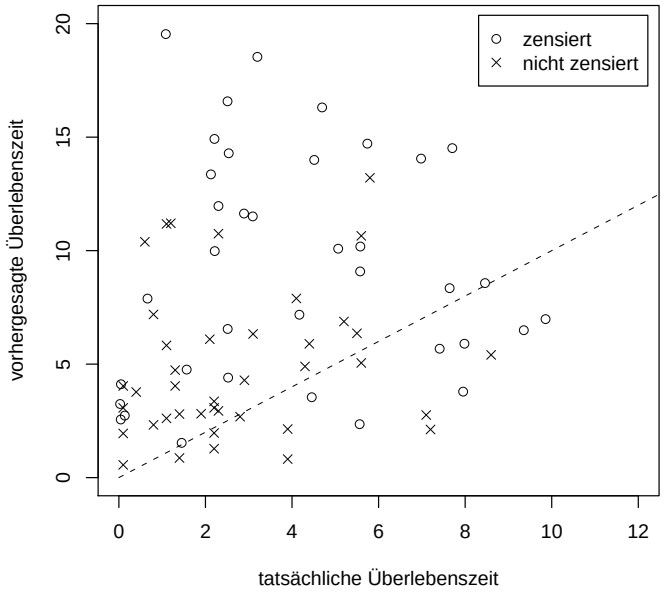


Abbildung 5.5: Darstellung vorhergesagte Überlebenszeiten im Vergleich zu den tatsächlichen Überlebenszeiten. Umso näher die Markierung der nichtzensierten Patienten "x" an der Diagonalen liegt, desto besser ist die Vorhersage. Bei zensierten Patienten kann keine Aussage über die Qualität gegeben werden. Liegt die vorhergesagte Überlebenszeit über der beobachteten Zensurzeit, so kann kein Fehler berechnet werden.

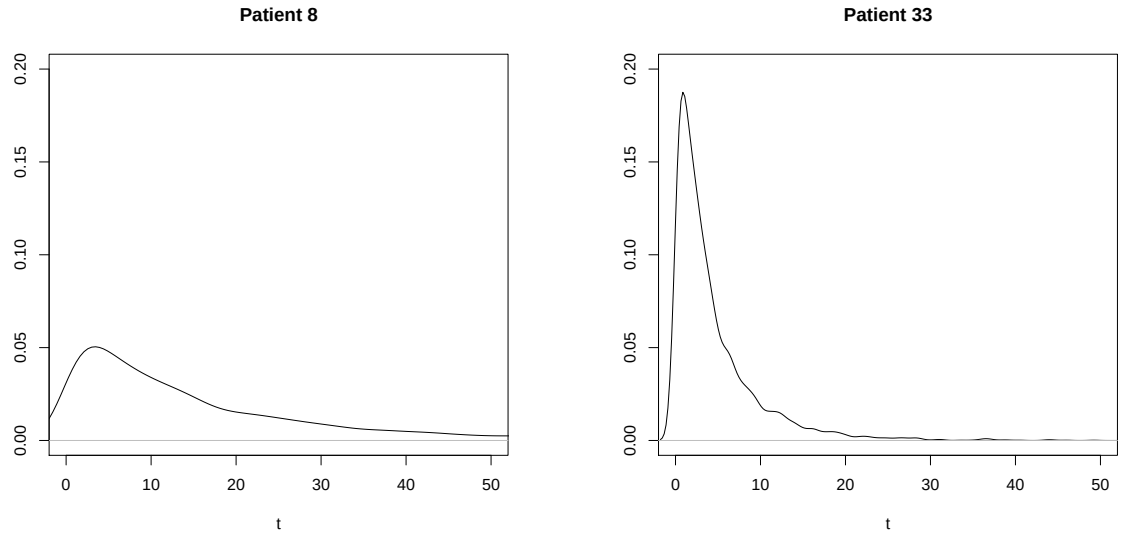
5.1.4 Klassifikation in Risikogruppen

Neben den direkten Vorhersagen von Überlebenszeiten haben die gewonnen Ergebnisse einen weiteren Nutzen. So können Patienten auf Grund der vorhergesagten Überlebenszeiten in unterschiedliche Risikogruppen eingeteilt werden. Die hier gewonnenen Erkenntisse können als qualitatives Maß für die Vorhersage der Überlebenszeiten verwendet werden.

Abbildung 5.7 zeigt die Survivor-Funktionen der Gruppen von Langzeitüberlebenden und Kurzzeitüberlebenden. Die Patienten des Validierungsdatensatzes wurden in Langzeitüberlebende, Überlebenszeiten größer 5 Jahren, und Kurzzeitüberlebende, Überlebenszeit kleiner 5 Jahren, basierend auf den getroffenen Vorhersagen eingeteilt. Die dargestellten Survivor-Funktionen wurden anschließend mit den tatsächlichen Überlebenszeiten berechnet.

Der Logrank-Test (siehe Abschnitt A.2 für weitere Informationen bezüglich des Tests) auf Ähnlichkeit der beiden Survivor-Funktionen war mit einem P-Wert von 3.59×10^{-8} erfolgreich und die Gruppen wurden sicher voneinander getrennt.

Die Abbildungen 5.8(a) und 5.8(b) zeigen die zeitabhängigen Receiver Operating Characteristic-Kurven (ROC) für $t = 1$ Jahr (a) respektive $t = 5$ Jahre (b). Die Abbildungen stellen die Sensitivität gegen $1-$Spezifität für jeden möglichen cutoff-Wert dar. Eine genau Beschreibung der ROC-Kurven befindet



(a) Wahrscheinlichkeitsdichte der Überlebenszeitvorhersagen für Patient 8

(b) Wahrscheinlichkeitsdichte der Überlebenszeitvorhersagen für Patient 33

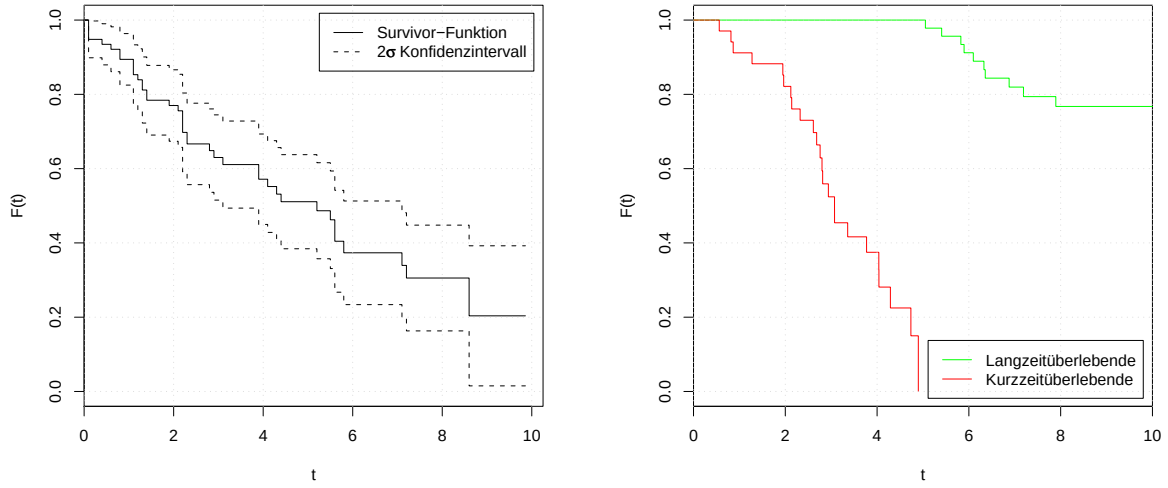
Abbildung 5.6: Wahrscheinlichkeitsdichte ausgewählter Überlebenszeitvorhersagen. Patient 33 (b) besitzt im Vergleich zu den Patient 8 (a) eine sehr geringe Standardabweichung der Überlebenszeitverteilung, wodurch der Peak stärker ausgeprägt ist.

sich im Anhang A.4. Die beiden Abbildungen 5.8(a) und 5.8(b) machen deutlich, dass eine Klassifikation in Lang- und Kurzzeitüberlebende für einen Schwellenwert von $t = 1$ Jahr (a) sowohl für einen Schwellenwert von $t = 5$ Jahren (b) erfolgreich war. So konnte auch gezeigt werden, dass die Patienten des Validierungdatensatzes nicht nur erfolgreich in Lang- und Kurzzeitüberlebende ($t = 5$ Jahre) eingeteilt wurden, sondern ebenfalls eine Klassifikation in eine Hochrisikogruppe mit einer Überlebenszeit von weniger als einem Jahre möglich war.

Abschließend beschreibt die Abbildung 5.9 die Fläche unter der Kurve (AUC) in Abhängigkeit des Schwellenwertes an dem eine Unterscheidung in Lang- und Kurzzeitüberlebende durchgeführt wurde. Es zeigte sich, dass die besten Werte für Spezifität und Sensitivität bei einem Schwellenwert von 5 Jahren erreicht wurde. Dies entspricht auch in etwa der Hälfte der maximal beobachteten Zeit.

5.1.5 Vergleich mit dem Gradientenabstiegsverfahren

Eine ähnliche Methode zur Relevanzbestimmung von Genen und zur Vorhersage von Überlebenszeiten entwickelte Kaderali [KZF$^+$06]. Der *CASPAR* (Cancer Survival Prediction using Automatic Relevance determination) getaufte Algorithmus benützt ein Gradientenabstiegsverfahren um eine Maximum-Likelihood-Funktion zu optimieren. Die dort verwendete Maximum-Likelihood-Funktion entspricht der in dieser Arbeit verwendeten Funktion. Ein Unterschied ist die verwendete a-priori-Verteilung zum Ausdünnen der Parameter.



(a) Kaplan-Meier-Plot des vollständigen *SIMULATE* Validierungsdatensatzes

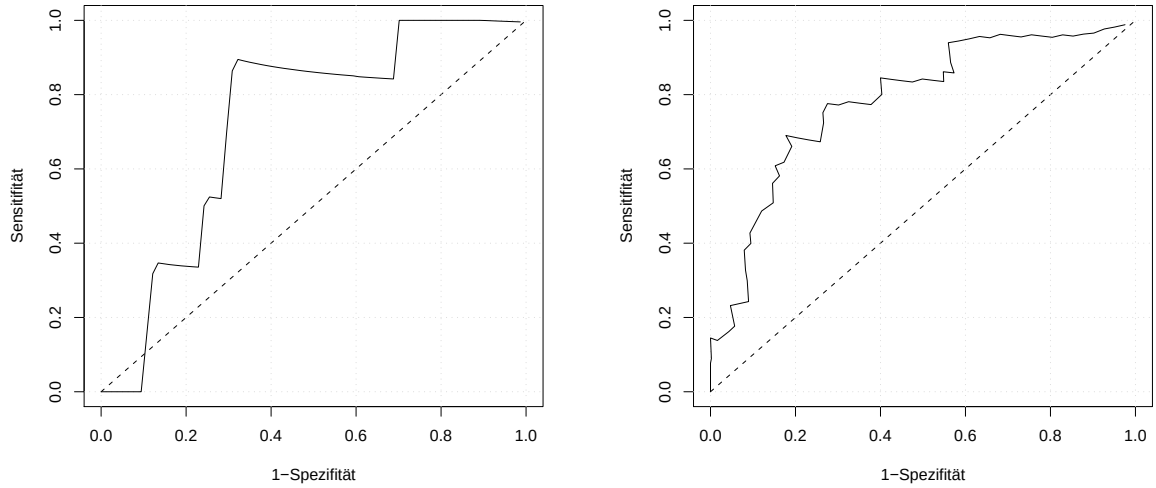
(b) Kaplan-Meier-Plot der Kurz- und Langzeitüberlebenden

Abbildung 5.7: Kaplan-Meier-Kurven des *SIMULATE* Datensatzes. Abbildung (a) zeigt die Kaplan-Meier-Kurve des gesamten Datensatzes. Die Abbildung (b) zeigt die Kaplan-Meier-Kurven der Langzeitüberlebenden ($t \geq 5$) und Kurzzeitüberlebenden ($t \leq 5$). Der Logrank Test auf die Gleichheit der Gruppen war erfolgreich (P-Wert$= 3.59 \times 10^{-8}$).

So wird in der Arbeit von Kaderali eine a-priori-Verteilung verwendet, die eine nummerische Integration benötigt. Um diese aufwändige Integration zu vermeiden, wird versucht, die in dieser Arbeit herangezogene a-priori-Verteilung zu verwenden. Es stellte sich heraus, dass es durch die Verwendung des LQ-Prior nicht möglich war, den Regressionsparameter θ_1 des relevanten ersten Gens ausreichend zu bestimmen. Der CASPAR-Algorithmus bestimmte einen Wert von $\theta_1 \sim 0.12$.

Abbildung 5.10 stellt die Maximum-Likelihood-Funktion des Gradientenabstiegsverfahrens dar. Hierbei wurde der Regressionsparameter θ_1 im Bereich von -0.5 und 1.5 variiert, während die übrigen Regressionsparameter $\theta_2, \cdots \theta_{7400}$ bei 0 belassen wurden. Wie in Abbildung 5.10 zu sehen, befindet sich im Bereich von 0.12 ein lokales Minimum, das der Gradientenabstieg nicht mehr verlassen konnte, wodurch dieser niemals das globale Optimum erreichte. Auch durch eine Anpassung der a-priori-Verteilung konnte kein besseres Ergebnis erreicht werden bei dem das globale Optimum erreicht wurde. Dies resultierte zudem in einer extrem schlechten Klassifikation der Patienten.

Eine Möglichkeit dieses Problem zu umgehen wäre es, unterschiedliche Startpunkte für den Gradientenabstieg zu verwenden. Dies würde aber auch ein gewisses Wissen um die tatsächlichen Regressionsparameter voraussetzen, was selten der Fall ist. Ebenso würde dieses Verhalten den Rechenzeitvorteil des Gradientenabstiegsverfahrens vermutlich zunichte machen.



(a) Zeitabhängige ROC-Kurve für $t = 1$ Jahr

(b) Zeitabhängige ROC-Kurve für $t = 5$ Jahre

Abbildung 5.8: ROC-Kurven für $t = 1$ (a) und $t = 5$ (b). Die Fläche unter den Kurven beträgt $AUC(t = 1) = 0.7015$(a) und $AUC(t = 5) = 0.7579$(b).

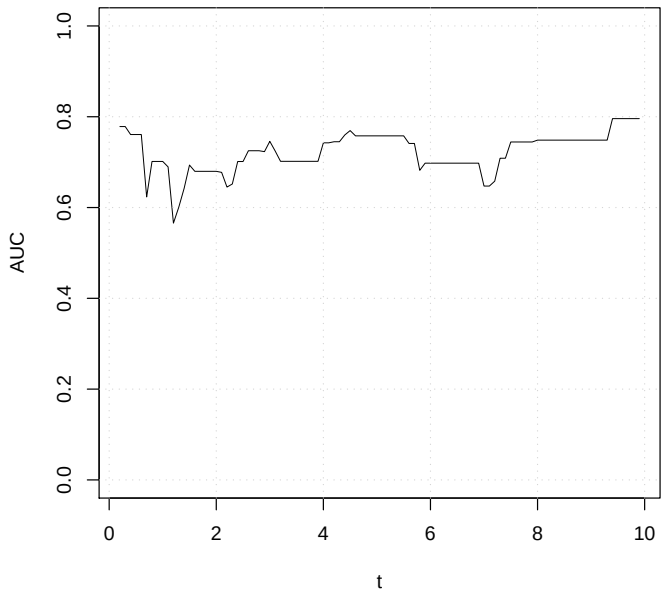


Abbildung 5.9: Darstellung der Fläche unter der ROC-Kurven in Abhängigkeit von der Zeit t.

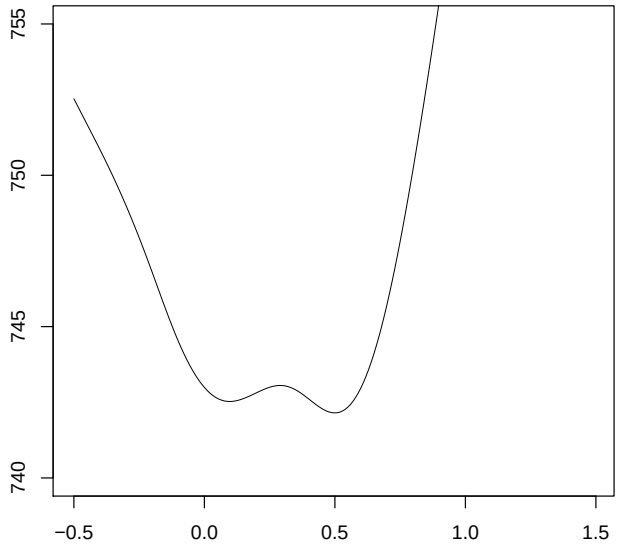


Abbildung 5.10: Darstellung der Maximum-Likelihood-Funktion des Gradientenabstiegsverfahren. Variation von θ_1 im Bereich von -0.5 und 1.5 während $\theta_2, \cdots \theta_{7400}$ bei 0 belassen werden..

5.2 Der *NEUROBLASTOM* Datensatz

5.2.1 Beschreibung des Datensatzes

Nachdem die grundlegende Funktionsfähigkeit der Methode durch den simulierten Datensatz bestätigt werden konnte, wurde das Verfahren anschließend auf einen realen Datensatz angewandt. Hierzu stand der *NEUROBLASTOM* Datensatz, publiziert von Oberthuer et al. [OBW+06], zur Verfügung.

Bei diesem Datensatz wurden auf einem Oligonukleotid-Array jeweils 10163 Genexpressionsmessungen von 256 Patienten durchgeführt. Das Alter der Patienten befindet sich zwischen 0 und 256 Monaten, der Median liegt bei 15 Monaten. Tabelle 5.3 zeigt die Zusammensetzung des Datensatzes. Es fällt auf, dass 71.48% der Patienten zensiert sind und somit der Todeszeitpunkt nicht bekannt ist. Es hat sich gezeigt, dass diese starke Zensur des Datensatzes keinen Einfluss auf die Funktionsfähigkeit der Methode hatte.

(a) vollständiger Datensatz

	Patienten	**Anteil**	**Median**	**max. Beobachtungszeit**
zensiert	183	71.48	4.89	15.56
unzensiert	73	28.52	1.50	5.52
gesamt	256	100.00	3.92	15.56

(b) Trainingsdatensatz

	Patienten	**Anteil**	**Median**	**max. Beobachtungszeit**
zensiert	137	85.625	4.96	15.56
unzensiert	23	14.375	2.21	4.99
gesamt	160	100.000	4.56	15.56

(c) Validierungsdatensatz

	Patienten	**Anteil**	**Median**	**max. Beobachtungszeit**
zensiert	81	84.375	4.63	9.51
unzensiert	15	15.625	2.66	8.84
gesamt	96	100.000	4.33	9.51

Tabelle 5.3: Zusammensetzung des NEUROBLASTOM Datensatzes. *Mediane* und *maximale Beobachtungszeit* in Jahren.

Geschätzte Baseline-Hazard-Funktion

Um geeignete Initialwerte für die Stufenhöhen der Baseline-Hazard-Funktion zu schätzen versucht man das Integral der Baseline-Hazard-Funktion innerhalb der Survivor-Funktion Gleichung (5.1) so zu wählen, dass sie ungefähr dem Kaplan-Meier-Schätzers des Datensatzes entspricht. Hierbei wird von einer

konstanten Baseline-Hazard-Funktion ausgegangen. Anschließend kann von den geschätzten Survivor-Funktionen auf die Initialwerte der Baseline-Hazard-Stufenhöhen geschlossen werden.

$$F(t|x,\theta) \;=\; \exp\left[-\int_0^t \lambda_0(u)\,du\right]^{\exp\langle\theta,x\rangle} \;=\; \exp\left[-t\;\lambda_0(u)\right] \tag{5.1}$$

Natürlich können die so erhaltenen Werte für $\lambda_0(t)$ nur als Richtwert dienen und müssen bei der späteren Parameterwahl ebenso angepasst werden.

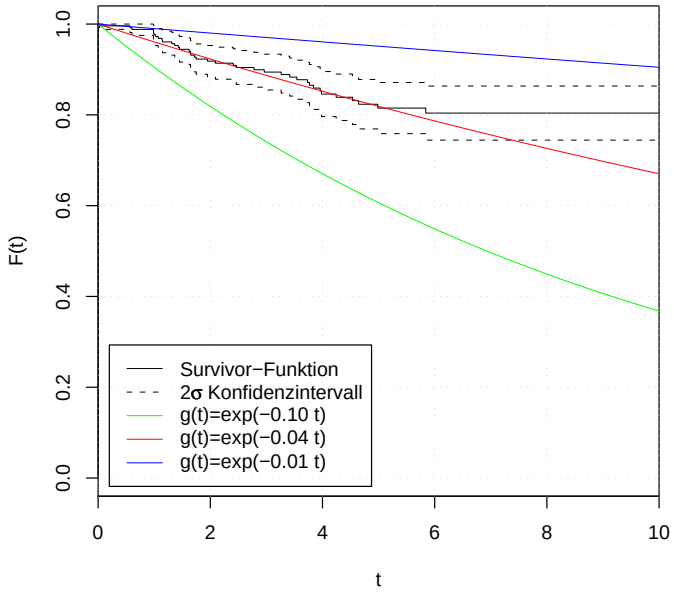


Abbildung 5.11: Kaplan Meier Plot der *NEUROBLASTOM* Daten sowie Schätzungen der Baseline-Hazard-Funktionen

Abbildung 5.11 zeigt den Kaplan-Meier-Plot der *NEUROBLASTOM* Daten sowie einige geschätzte Baseline-Hazard-Funktionen. Die Wahl der Initialwerte der Stufenhöhen hat sich als sehr einflussreich auf das Sampling der Überlebenszeiten sowie auf den gesamten Verlauf der Markovkette erwiesen.

So führt ein zu hoch (z.B. $\lambda_0(t) \geq 0.3$) oder zu niedrig (z.B. $\lambda_0(t) \leq 0.02$) gewählter Initialwert dazu, dass die Markovkette nicht in endlicher Zeit konvergiert und die von ihr erzeugten Überlebenszeiten äußerst ungenau sind. Ein zu niedrig gewählter Initialwert der Stufenhöhen führt bei zensierten Patienten zu extrem großen Vorhersagen, oftmals mehrere Hundert Jahre. Hingegen führen kleine Veränderungen im Bereich von ± 0.05 zu keinen auffälligen Veränderungen der Resultate. Dies kann damit erklärt werden, dass sich die Stufenhöhen während des Markovketten-Laufes anpassen. Somit kann man davon ausgehen, dass der *Lernvorgang* kleinere Abweichungen bei den Initialwerten zuverlässig ausgleicht.

5.2.2 Ermittelte Regressionsparameter

Bei der zur Analyse verwendete Markovkette wurden 5000 Iterationen, nach einem Burn-In von 500 Iterationen, mit jeweils 500 Leapfrog-Schritten durchgeführt. Die Schrittgröße ϵ wurde zufällig von einer Gammaverteilung mit Erwartungswert 1.8×10^{-6} und Varianz 1.658×10^{-6} gezogen. Da bei diesem Datensatz nicht zu erwarten war, dass nur ein einzelnes Gen einen sehr starken Einfluss auf die Überlebenszeit hat, wurden die Parameter der Gammaverteilung so gewählt, dass nur sehr kleine Werte für die Schrittgröße ϵ erzeugt werden.

Eine vollständige Übersicht über alle Parameter dieser Markovkette befindet sich im Anhang B.1.

Als relevant bestimmte Gene

Wie stark ein bestimmtes Gen Einfluss auf die Überlebenszeit des Patienten hat spiegelt sich in dem assoziierten Regressionsparameter (Gewicht) wieder. So gibt der Betrag dieses Regressionsparameters an, wie stark dieses Gene bei der Vorhersage der Überlebenszeit gewichtet wird. Ein hoher positiver Wert kann mit einem verkürzenden, ein hoher negativer Wert mit einem verlängernden Einfluss auf die Überlebenszeiten in Verbindung gebracht werden.

(a) Größte Gewichte

θ_i	Std. Abw.	Name	Nr. i
0.0525	6.18×10^{-5}	SLC25A13	3454
0.0468	7.06×10^{-5}	Hs356531.174	3745
0.0418	0.0007014	Hs116608.1	676
0.0400	7.19×10^{-6}	AF116661	9209
0.0337	2.58×10^{-5}	Hs151032.1	8395
0.0336	7.35×10^{-5}	CP110	7372
0.0334	9.96×10^{-5}	I_3338200	8387
0.0334	4.16×10^{-5}	SH3KBP1	9169
0.0331	3.32×10^{-5}	SPARC	5577
0.0329	1.21×10^{-5}	Hs85092.1	1170

(b) Kleinste Gewichte

θ_i	Std. Abw.	Name	Nr. i
−0.03968	3.36×10^{-5}	AKAP4	2648
−0.03854	2.19×10^{-5}	IER3	5644
−0.03621	4.09×10^{-5}	THC1487837	6982
−0.03603	3.21×10^{-5}	THC1506334	6904
−0.03436	0.0001022	UTS2	5066
−0.03186	5.48×10^{-5}	RBM6	2254
−0.03179	2.26×10^{-5}	DC13	3223
−0.03098	3.07×10^{-5}	BC005031	4303
−0.03065	0.0002207	I_3211815	5335
−0.03002	4.58×10^{-5}	A_32_BS105865	5819

Tabelle 5.4: Auflistung der Gene mit den zehn größten und zehn kleinsten ermittelten Regressionsparameter. Ein hoher positiver Wert kann mit kürzerem, ein hoher negativer Wert mit längerem Überleben in Verbindung gebracht werden.

Die Tabellen 5.3(a) und 5.3(b) zeigen die zehn größten und kleinsten Gewichte wie sie die hier analysierte Markovkette bestimmt hat.

Vergleicht man die am stärksten ausgeprägten Regressionsparameter unterschiedlicher Markovketten, so zeigte sich, dass diese von Markovkette zu Markovkette unterschiedlich sind. Dies könnte darauf hinweisen, dass es viele redundante Informationen in den Genexpressionsdaten gibt und dass mit unterschiedlichen Kombination von Regressionsparametern ähnlich gute Vorhersagen getroffen werden können.

Um zu überprüfen ob es bestimmte Gene gibt, die häufiger als andere als relevant bestimmt werden, wird im folgenden Abschnitt eine Häufigkeitsanalyse durchgeführt.

Häufigkeitsanalyse

Um festzustellen wie oft ein Gen als relevant bestimmt wird, wurden zehn Markovketten mit identischen Parametern gestartet. Als Trainingsdaten wurden jeweils 160 zufällig gewählte Patienten des *NEUROBLASTOM* Datensatzes verwendet. Mit den übrigen 96 Patienten wurde anschließend die Markovkette validiert.

Tabelle 5.5 zeigt die zehn Gewichte die am häufigsten unter den 50 größten bzw. kleinsten Gewichte der jeweiligen Markovkette gefunden wurden, sowie deren bestimmter Wertebereich.

(a) Die 20 häufigsten positiven Gewichte

	Name	$min(\theta_i)$	$max(\theta_i)$	Nr. i
5	Hs116608.1	0.022	0.025	676
5	AF233453	0.03	0.04	4954
4	AF073310	0.023	0.035	9809
4	HNOEL-iso	0.026	0.03	7901
4	PSMAL/GCP	0.025	0.036	7518
4	STK25	0.025	0.03	2966
4	I_962014	0.022	0.033	2591
4	AK026229	0.021	0.033	2554
4	Hs190368.1	0.025	0.026	1646
3	A_32_BS16	0.023	0.031	8344
3	APS	0.024	0.028	8335
3	PGPEP1	0.023	0.029	8202
3	AL832173	0.026	0.042	6201
3	COPS5	0.023	0.032	5628
3	I_932479	0.027	0.029	510
3	THC1513367	0.027	0.031	4182
3	BC019932	0.023	0.024	3983
3	KIAA0643	0.023	0.027	3726
3	PPFIA2	0.027	0.03	280
3	BC030713	0.022	0.026	1774

(b) Die 20 häufigsten negativen Gewichte

	Name	$min(\theta_i)$	$max(\theta_i)$	Nr. i
4	KLF13	-0.037	-0.024	5310
3	U61738	-0.042	-0.029	9906
3	BC038969	-0.026	-0.025	9877
3	TNFAIP1	-0.031	-0.031	9862
3	FLJ20516	-0.026	-0.022	8436
3	PER3	-0.026	-0.021	8325
3	CDC25C	-0.03	-0.021	7731
3	COG1	-0.03	-0.03	7624
3	NIFU	-0.032	-0.021	7410
3	A_32_BS10	-0.033	-0.024	5819
3	MPRP-1	-0.028	-0.021	4893
3	EEF1A1	-0.027	-0.022	305
3	APOA4	-0.024	-0.023	2415
3	ARHGDIA	-0.034	-0.022	2306
3	LOC144347	-0.028	-0.023	1157
3	AF273042	-0.035	-0.023	1100
2	TWIST1	-0.022	-0.022	9724
2	MGC40179	-0.036	-0.032	9702
2	AK095685	-0.022	-0.022	9692
2	CASP8	-0.045	-0.022	9671

Tabelle 5.5: Auflistung der 20 am häufigsten bestimmten Gene mit dem dazugehörigen Wertebereich. Die erste Spalte gibt an, in wie vielen der zehn Markovketten-Läufe das Gen unter den 50 größten bzw. kleinsten zu finden war.

Es zeigte sich, dass es die in Tabelle 5.4(a) und 5.4(b) dargestellten Genen bei bis zu 5 von 10 Markovketten-Läufen unter den 50 Genen mit dem größten bzw. kleinsten assoziierten Regressionsparameter gefunden wurden. So wurde das Gen *COPS5* (auch CSN5 oder JAB1) bereits in Verbindung mit einem Protein-Komplex, der die Progression bei Bruskrebs fördert, ([ALL+08] und [ZCS+07]).

Das am häufigsten für eine positive Prognose identifzierte Gen *KLF13* spielt, Unteruchungen zu Folge,

bei der Regulation der Autoimmunreaktion und bei der Verstärkung der Immunreaktion auf Krebszellen eine Rolle, [KA07].

Im folgenden Abschnitt diskutiere und beschreibe ich die die Funktion weiterer identifzierter Gene. Diese Gene wurden häufig durch Markovketten identifiziert.

5.2.3 Relevante Gene

Funktion von Gene mit negativer Wirkung auf die vorhergesagte Überlebenszeit.

SPARC (auch Osteonectin oder BM40) ist ein Matrix-assoziiertes Protein das Veränderungen in der Zellform beeinflusst indem es an der Synthese der extrazellulären Matrix Teil hat. Des weiteren hemmt es die Progression der Zelle (Bradshaw et al. [BGMS03]). Es wurde bei mehreren Studien ([CLW+06], [WLC+04] und [WDJB+05]) mit der negativen Einfluss auf die Prognose des Krebspatienten in Verbindung gebracht werden.

CP110 translatiert das CP110 Protein das sowohl für die Entstehung von Basalkörperchen als auch für die Entwicklung von Zentrosomen verantwortlich ist. So konnte nachgewiesen werden, dass ein Protein-Komplex bestehend aus CP110 und CEP97 die Biogenese von Zentriolen und Zilien hemmt. Dies ist ein Prozess der bei Krebs- und Lebererkrankungen beeinträchtigt ist [BDCS08].

STK25 (auch YSK1 oder SOK1) translatiert die Serine/Threonine Kinase 25 die mit den für die zerebrale, kavernöse Fehlbildung (CCM) verantwortlichen Genen (CCM1/KRIT1, CCM2, CCM3) bindet und zusammen mit dem Protein CCM2 einen Protein-Komplex bildet der essentiell für die Pathogenese der CCM ist [VSS+07].

COPS5 (auch CSN5 oder JAB1) kodiert ein Protein das Teil des COP9 Signalosoms ist. Dieser hochkonservierte Protein-Komplex dient als Regulator für viele *signaling pathways*. Es wurde bereits mehrfach in Verbindung mit einem, die Progression bei Bruskrebs fördernden, Protein-Komplex gebracht. ([ALL+08] und [ZCS+07]).

APS (auch SH2B2) kodiert ein von B-Lymphozyten exprimiertes Protein. Es fungiert als Adaptor-Molekül mit SH2 und PH Domänen. Es scheint eine Rolle bei der Signal-Transduktion des Shc/Grb2-Rezeptors zu spielen. Es wurde eine erhöhte Expression bei einigen Osteosarkom-Zelllinien nachgewiesen [YWS+].

Funktion von Gene mit positiver Wirkung auf die vorhergesagte Überlebenszeit.

KLF13 gehört zu einer Familie die vor allem Transformationsfaktoren kodiert die aus, um ein Zink-Molekül angeordneten, DNA bindenden Domänen besteht. Diese Transformationsfaktoren binden vor allem an GC-reiche Regionen wie GT- oder CACCC-Boxen. Unteruchungen zu Folge [KA07] spielen sie eine wichtige Rolle bei der Regulation der Autoimmunreaktion und bei der Verstärkung dieser auf Krebszellen.

PER3 Die Unterbrechung oder Störung des Tagesrhythmus des Gehirns spielt eine wichtige Rolle bei der Entwicklung von Krebs. Bei der Untersuchung der Regulation des Tagesrhythmus [YCL+06] hat sich herausgestellt das bei Patienten chronischer Stammzellen-Leukämie die Expression, der für die Steuerung des Tagesrhythmus verantwortlichen Gene (hPER1, hPER2, hPER3 und hCRY1), vermindert ist.

CDC25C ist ein hochkonserviertes Gen das eine Schlüsselrolle bei der Regulation der Zellteilung spielt. Das kodierte Protein ist eine Tyrosin Phosphatase und gehört zu der CDC25 Phosphatase Familie. Bei der Untersuchung von Prostatakrebszellen zeigte sich eine Einschränkung der Zellteilung nachdem die Expression von CDC25C gehemmt wurde [PNKS08].

IER3 (auch IEX-1) fungiert in Zellen als Schutz vor Tumornekrose-Faktoren und fördert die Apoptose von Tumorzellen. Untersuchungen [MRI+07] zeigten das bei der induzierten Apoptose, mit Hilfe von Gastrin-17, von Tumorzellen eine hohe Menge des IER3-Proteins ausgeschüttet wurde. Verhinderte man die Expression von IER3 durch siRNA, so konnte die Apoptose mit Gastrin-17 nicht eingeleitet werden.

5.2.4 Überlebenszeitvorhersage

Wie bereits bei der Überlebenszeitvorhersage des *SIMULATE* Datensatzes, wird hier für jeden Schritt der Markovkette mit Hilfe des Acceptance-Rejection-Verfahren eine Überlebenszeit für jeden Patienten von der Überlebenszeitverteilung $p(j_i|x_i, \theta_n)$ gezogen und anschließend wird die daraus resultierende Verteilung mit der tatsächlichen Überlebenszeit, soweit vorhanden, verglichen.

Die Tabellen 5.5(a) und 5.5(b) stellen die Mediane der vorhergesagten Überlebenszeitverteilungen der Patienten den tatsächlichen Überlebenszeiten und Zensurzeiten gegenüber.

Tabelle 5.5(a) zeigt die 15 nichtzensierten Patienten des Validierungsdatensatzes. Es wurden fast ausschließlich Überlebenszeiten vorhergesagt, die über den tatsächlichen Überlebenszeiten liegen, eine Ausnahme ist Patient 58. Die Fehler der Vorhersage für Patient 58 ist mit 0.98 Jahren relativ gering im Vergleich zu den übrigen Vorhersagefehlern.

Die in Tabelle 5.5(a) dargestellte Zusammenfassung der Überlebenszeitverteilungen der zensierten Patienten beschreibt ein ähnliches Bild wie das bei der Auswertung des *SIMULATE* Datensatzes. So kann auch hier für nur 3 der ersten 20 Patienten ein Fehler berechnet werden, da bei diesen die Vorhersagezeit kürzer ist als die Dauer nach der der Patient zensiert wurde.

Fasst man die zensierten sowie die nichtzensierten Patienten zusammen und berechnet die durchschnittlichen Werte der Mediane der Vorhersageverteilung, so zeigt sich, dass die durchschnittliche Vorhersage der zensierten Patienten mit 10.97 Jahren deutlich über der nichtzensierter Patienten mit 3.86 liegt. Dies konnte auch schon bei den *SIMULATE* Daten geobachtet werden. Ähnlich verhält es sich mit der durchschnittlichen Standardabweichung. So beträgt die durchschnittliche Standardabweichung der zensierten Patienten 6.21 Jahre und ist somit deutlich größer als die durchschnittliche Standardabweichung der nichtzensierten Patienten mit 5.73 liegt.

(a) Nichtzensierte Patienten.

ID	Tatsächl.	Median	Std. Abw.	Fehler
20	3.84	4.37	6.61	0.53
29	1.07	4.08	6.15	3.01
33	4.24	4.86	7.41	0.61
36	3.73	5.04	7.4	1.31
38	3.98	6.06	9.05	2.07
44	2.96	3.51	5.36	0.55
47	2.78	3.93	5.7	1.15
48	1.37	2.02	3.11	0.66
49	1.15	6.22	9.04	5.07
50	1.43	3.09	4.4	1.66
51	5.84	4.53	6.72	1.31
52	0.6	2.36	3.57	1.75
58	4.53	3.56	5.07	0.98
89	0.99	1.75	2.56	0.77
92	1.45	2.49	3.8	1.04

(b) Zensierte Patienten.

ID	Tatsächl.	Median	Std. Abw.	Fehler
0	5.81	8.39	12.03	≥ 0
1	6.37	12.42	17.96	≥ 0
2	6.12	4.18	6.35	≥ 1.94
3	2.1	9.55	13.27	≥ 0
4	5.5	11.24	17.19	≥ 0
5	9.51	17.07	23.7	≥ 0
6	3.76	11.73	17.79	≥ 0
7	2.92	10.3	14.9	≥ 0
8	7.33	6.01	9.07	≥ 1.32
9	6.11	12.89	19.39	≥ 0
10	2.72	8.94	14.0	≥ 0
11	8.68	14.33	21.43	≥ 0
12	6.81	14.76	21.37	≥ 0
13	3.32	10.02	14.02	≥ 0
14	9.19	15.83	23.6	≥ 0
15	4.89	10.7	16.12	≥ 0
16	1.32	8.88	13.62	≥ 0
17	7.1	11.39	16.15	≥ 0
18	2.58	5.53	8.15	≥ 0
19	8.34	9.95	14.05	≥ 0
25	4.34	2.77	4.03	≥ 1.57

Tabelle 5.6: Zusammenfassung der Überlebenszeitverteilungen der 15 nicht zensierten Patienten (a) und der ersten 20 zensierten Patienten (b). Die zweite Spalte zeigt. Die tatsächliche Überlebenszeit bzw. bei den zensierten Patienten die Beobachtungszeit. Die Spalten drei bis fünf geben die Mediane, Standardabweichung sowie den Fehler der vorhergesagten Überlebenszeitverteilung an.

Um einen Überblick über die vorhergesagten sowie die tatsächlichen Überlebenszeiten zu erhalten, stellt die Abbildung 5.12 diese gegenüber. Hier bestätigt sich, dass die Vorhersagen der nichtzensierten Patienten unter denen der zensierten liegen und somit näher am Ursprung angesiedelt sind.

Betrachtungen einzelner Patienten

Durch die Vorhersage von Überlebenszeitverteilungen erhält man mehr Informationen als von einzelnen Werten. Die Abbildung 5.13 zeigt die Wahrscheinlichkeitsdichteverteilungen von vier ausgewählten Patienten die hier näher betrachtet werden sollen.

Für die Patienten 48 (Abbildung 5.13(a)) und 38 (Abbildung 5.13(b)) ist die tatsächliche Überlebenszeit bekannt. Bei Patient 48 beträgt diese 2.78 Jahre und der Median der Vorhersage beträgt 2.02 Jahre. Die Standardabweichung der Überlebenszeitverteilung liegt mit 3.11 Jahren unter dem Durchschnitt. Man sieht hier sehr schön den ausgeprägten Peak bei ~ 2.2 Jahren, der auf eine hohe Sicherheit bei der Vorhersage schließen lässt. Patient 38 hingegen besitzt die größe Standardabweichung unter den nichtzensierten Patienten mit 9.05 Jahren. Der Überlebenszeitvohersage-Peak ist weniger

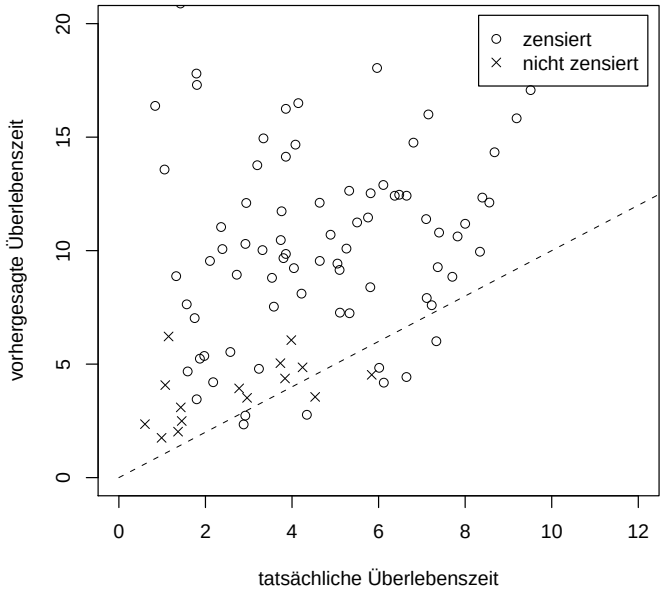


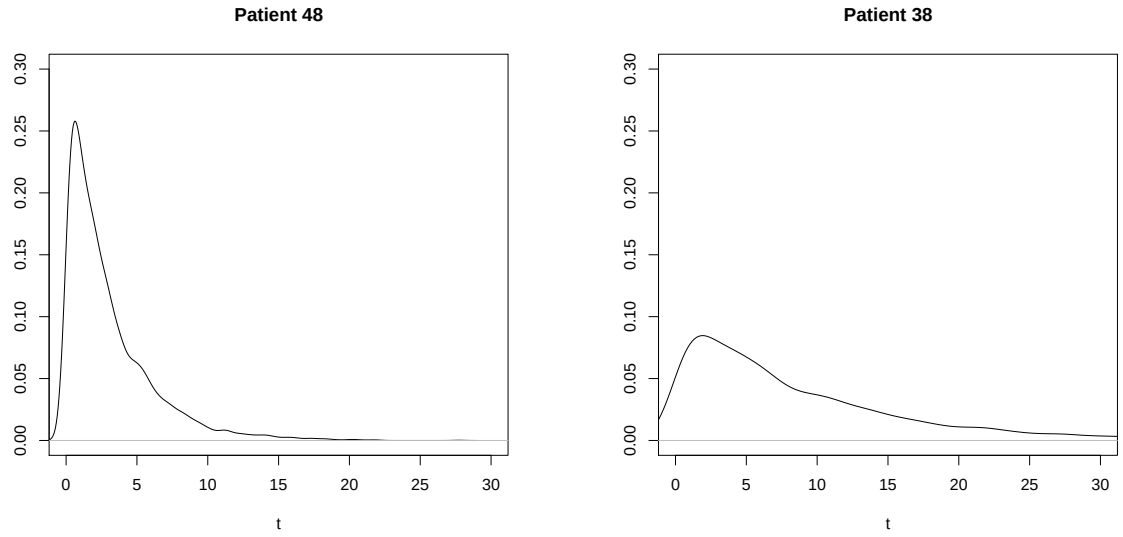
Abbildung 5.12: Darstellung vorhergesagte Überlebenszeiten im Vergleich zu den tatsächlichen Überlebenszeiten. Umso näher die Markierung der nichtzensierten Patienten "x" an der Diagonalen liegt, desto besser ist die Vorhersage. Bei zensierten Patienten kann keine Aussage über die Qualität gegeben werden. Liegt die vorhergesagte Überlebenszeit über der beobachteten Zensurzeit, so kann kein Fehler berechnet werden.

deutlich ausgeprägt, was die hohe Standardabweichung und die zugrundeliegende Unsicherheit widerspiegelt.

Die Patienten 25 (Abbildung 5.14(a)) und 6 (Abbildung 5.14(b)) sind zensiert und somit ist keine tatsächliche Überlebenszeit bekannt. Patient 6 weist eine für zensierte Patienten überdurchschnittlich hohe Standardabweichung von $\sigma_6 = 17.79$ Jahren bei einer maximalen Beobachtungszeit von nur $t_6 = 3.76$ Jahren auf. Der vorhergesagten Überlebenszeit von $t_6 = 15.2$ haftet somit eine hohe Unsicherheit an.

Patient 25 (Abbildung 5.14(a)) hingegen zeigt einen sehr deutlichen Peak in seiner Wahrscheinlichkeitsdichteverteilung. Er gehört zu der Gruppe von zensierten Patienten deren vorhergesagte Überlebenszeit unterhalb der Zensurzeit liegt und somit mit Sicherheit falsch ist.

Tabelle 5.7 zeigt nur die zensierten Patienten für die sich der Vorhersagefehler berechnen lässt. Diese 7 Patienten besitzen im Gegensatz zu den übrigen zensierten Patienten eine geringe Standardabweichung. Der Vorhersagefehler liegt bei diesen Patienten zwischen 0.19 und 2.21 Jahren.



(a) Wahrscheinlichkeitsdichte der Überlebenszeitvorhersagen für Patient 48

(b) Wahrscheinlichkeitsdichte der Überlebenszeitvorhersagen für Patient 38

Abbildung 5.13: Wahrscheinlichkeitsdichte ausgewählter Überlebenszeitvorhersagen nichtzensierter Patienten. Patient 48 (a) besitzt im Vergleich zu den Patient 38 (b) eine sehr geringe Standardabweichung der Überlebenszeitverteilung, wodurch der Peak stärker ausgeprägt ist.

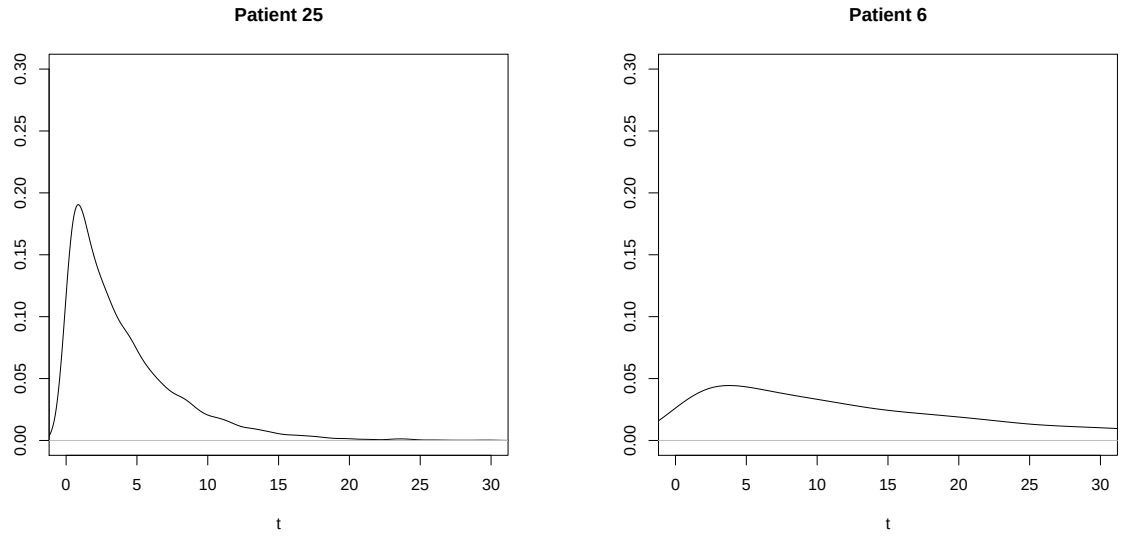
ID	Tatsächl.	Median	Std. Abw.	Fehler
2	6.12	4.18	6.35	≥ 1.94
8	7.33	6.01	9.07	≥ 1.32
23	2.88	2.34	3.79	≥ 0.54
25	4.34	2.77	4.03	≥ 1.57
31	6.01	4.84	7.11	≥ 1.18
34	6.64	4.43	6.16	≥ 2.21
39	2.92	2.73	3.85	≥ 0.19

Tabelle 5.7: Zusammenfassung der Überlebenszeitverteilungen von zensierten Patienten deren Median unterhalb der Zensurzeit liegt.

5.2.5 Klassifikation in Risikogruppen

Bei der Klassifikation stellte sich heraus, dass mit Hilfe der berechneten Überlebenszeiten die Klassifikation in Lang- und Kurzzeitüberlebende mit einer hohen Sensitivität sowie Spezifität durchgeführt werden konnte.

Die Abbildung 5.15(a) zeigt die tatsächliche Survivor-Funktion des *NEUROBLASTOM* Datensatzes in Form eines Kaplan-Meier-Plots. Die Abbildung 5.15(b) hingegen zeigt die Kaplan-Meier-Kurven für Langzeitüberlebende (Überlebenszeit ≥ 5 Jahren) und Kurzzeitüberlebende (Überlebenszeit ≤ 5 Jahren). Hierbei wurden zur Klassifikation die vorhergesagten Überlebenszeiten benutzt. Die dargestellten Survivor-Funktionen wurden anschließend aus den tatsächlichen Überlebenszeiten berech-



(a) Wahrscheinlichkeitsdichte der Überlebenszeitvorhersagen für Patient 25

(b) Wahrscheinlichkeitsdichte der Überlebenszeitvorhersagen für Patient 6

Abbildung 5.14: Wahrscheinlichkeitsdichte ausgewählter Überlebenszeitvorhersagen zensierter Patienten. Patient 25 5.14(a) besitzt im Vergleich zu Patient 6 5.14(b) eine sehr geringe Standardabweichung der Überlebenszeitverteilung, wodurch der Peak stärker ausgeprägt ist.

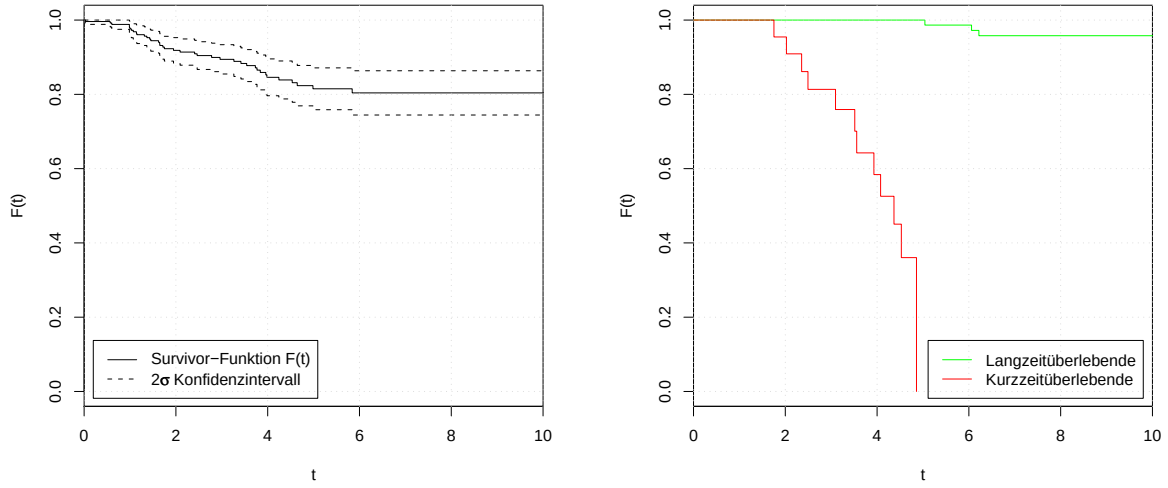
net.

Hier ist deutlich die Trennung der beiden Gruppen zu sehen. Der Logrank-Test auf Gleichheit der Survivor-Funktionen ergab mit einem P-Wert von 6.84×10^{-6}, dass die Trennung der Gruppen ähnlich erfolgreich ist wie die des simulierten Datensatzes *SIMULATE*. Die Wahl des Schwellenwertes ($t = 1$ Jahr oder $t = 3$) ergibt keinen signifikanten Unterschied bei der Qualität der Trennung.

Die Abbildungen 5.16(a) und 5.16(b) zeigen die zeitabhängigen Receiver Operating Characteristic-Kurven (ROC) für $t = 1$ Jahr (a), respektive $t = 5$ Jahre (b). Hier fallen die offensichtlich sehr guten Werte für Spezifität und Sensitivität der Klassifikation auf. Somit zeigt sich, dass die Gruppierung der Patienten erfolgreich vollzogen wurde.

Abbildung 5.17 beschreibt abschließend die Fläche unter der Kurve in Abhängigkeit des Schwellenwertes zur Unterscheidung in Lang- und Kurzzeitüberlebende. Hier zeigt sich sehr deutlich die fast ausnahmslos sehr guten AUC-Werte und somit auch die sehr guten Werte für die Sensitivität und Spezifität.

Betrachtet man Abbildung 5.17, so fällt der markante, äußerst schlechte AUC Wert ($AUC = 0.11$) bei $t \sim 0.9$ auf. Dieser lässt sich damit erklären, dass es nur einen zensierten Patienten und einen nichtzensierten Patienten mit einer Überlebenszeit bzw. Beobachtungszeit von unter 0.9 Jahren gibt. Sollten diese beiden Patienten falsch klassifiziert werden, fällt somit sehr stark die Sensitivität und somit auch der AUC-Wert.



(a) Kaplan-Meier-Plot des vollständigen *NEUROBLASTOM* Validierungsdatensatzes

(b) Kaplan-Meier-Plot der Kurz- und Langzeitüberlebenden

Abbildung 5.15: Kaplan-Meier-Kurven des *NEUROBLASTOM* Datensatzes. Abbildung (a) zeigt die Kaplan-Meier-Kurve des gesamten Datensatzes. Die Abbildung (b) zeigt die Kaplan-Meier-Kurven der Langzeitüberlebenden ($t \geq 5$) und Kurzzeitüberlebenden ($t \leq 5$). Der Logrank Test auf die Gleichheit der Gruppen war erfolgreich (P-Wert= 6.84×10^{-6}).

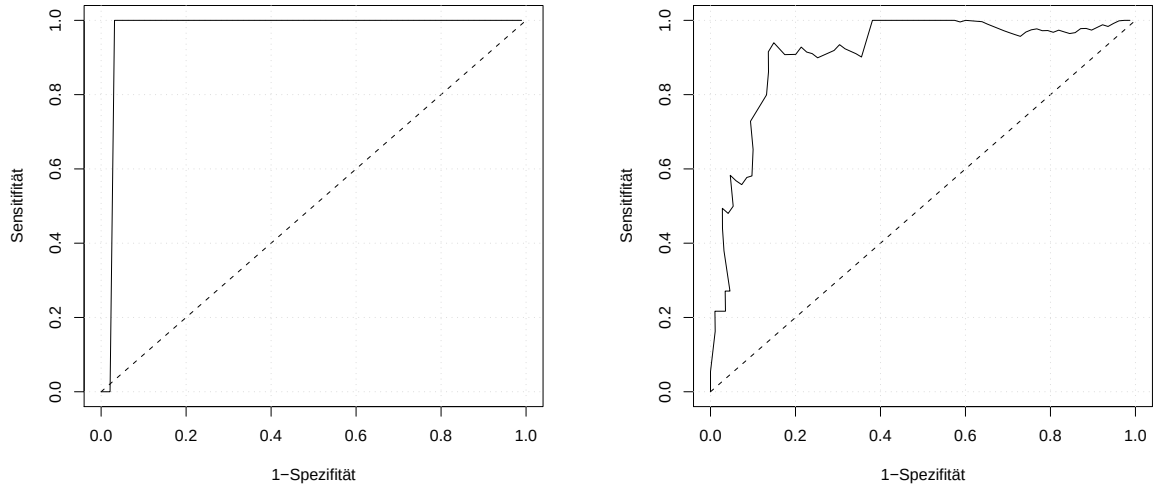
5.2.6 Vergleich mit dem Gradientenabstiegsverfahren

Um die Qualität des hier vorgestellten Markov-Chain-Monte-Carlo Ansatzes in Kontext zu bereits bestehenden Methoden zu setzen, wurde der in Abschnitt 5.1.5 kurz vorgestellte CASPAR-Algorithmus als Vergleich herangezogen.

So benutzt der CASPAR-Algorithmus, wie die in dieser Arbeit vorgestellte Methode, Genexpressionsdaten um Gene zu identifizieren, die einen signifikanten Einfluss auf die Überlebenszeit von Krebspatienten haben. Anschließend wird eine individuelle Vorhersage der Überlebenszeit für jeden Patienten vogenommen und eine Klassifikation durchzuführen. Die vollständige Analyse des *NEUROBLASTOM* Datensatzes wird beschrieben in Oberthuer et al. [OKK$^+$08].

Ähnlich der in Abschnitt 5.2.5 beschrieben Klassifikation in Riskiogruppen wurde bei der CASPAR Analyse ebenso eine Einteilung in Lang- und Kurzzeitüberlebende, mit einem Schwellenwert von 5 Jahren, durchgeführt. Hier erreichte der CASPAR-Algorithmus einen AUC-Wert von $AUC_{CASPAR}(t = 5) \sim 0.82$ im Vergleich zu $AUC_{MCMC}(t = 5) \sim 0.89$ des Markov-Chain-Monte-Carlo Ansatzes.

Vergleicht man die durchschnittlichen AUC-Werte bei Variaten des verwendeten Schwellenwertes, so ergibt sich für die Markov-Chain-Monte-Carlo-Methode ein durchschnittlicher AUC-Wert von $\overline{AUC_{MCMC}} = 0.91$. Dies bestätigt eine leichte Verbesserung der Klassifikation im Gegensatz zum CASPAR-Algorithmus mit einem durchschnittlichen AUC-Wert von $\overline{AUC_{CASPAR}} = 0.81$.



(a) Zeitabhängige ROC-Kurve für $t = 1$ Jahr

(b) Zeitabhängige ROC-Kurve für $t = 5$ Jahre

Abbildung 5.16: ROC-Kurven für $t = 1$ (a) und $t = 5$ (b). Die Fläche unter den Kurven beträgt $AUC(t = 1) = 0.9579$(a) und $AUC(t = 5) = 0.8934$(b).

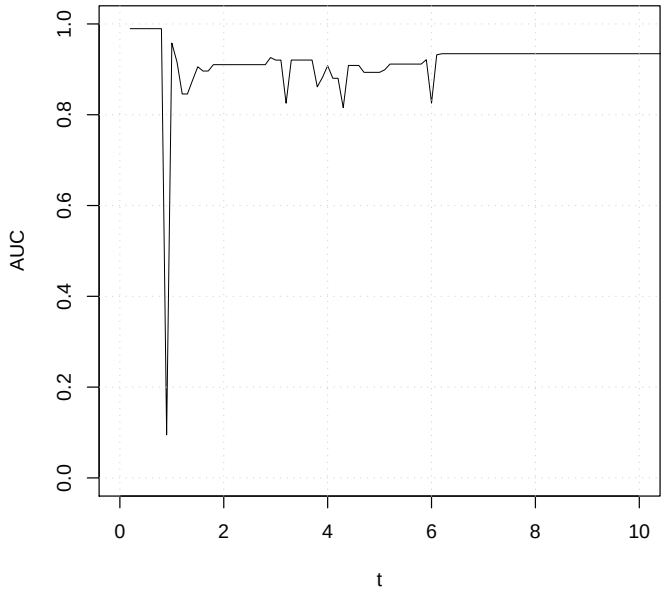


Abbildung 5.17: Darstellung der Fläche unter der ROC-Kurven in Abhängigkeit von der Zeit t.

6 Diskussion

Die beiden Hauptziele dieser Arbeit, mit Hilfe eines Markov-Chain-Monte-Carlo-Ansatzes für die Überlebenszeit relevante Gene zu identifizieren und die Überlebenszeit der Patienten vorherzusagen sowie diese zu klassifizieren, konnten bei simulierten wie auch realen Daten erreicht werden. Es konnte somit gezeigt werden, dass Datensätze mit sehr vielen Merkmalen und vergleichsweise wenigen Datenpunkten erfolgreich analysiert werden konnten, ohne zuvor eine Dimensionsreduzierung durchzuführen.

Ein Vorteil der Vohersage von Überlebenszeiten auf Basis des Markov-Chain-Monte-Carlo-Ansatzes besteht in der vollständigen Evaluierung der gesamten Verteilung über die Überlebenszeiten und nicht nur einer Vorhersage. Bei ausgewählten Patienten wurde die Möglichkeit der Überlebenszeitvorhersage auf Basis von Überlebenszeitverteilungen kurz angeschnitten.

Desweitern konnte nachgewiesen werden, dass eine Klassifikation der *NEUROBLASTOM* Patienten in Lang- und Kurzzeitüberlebende zuverlässig möglich war. Ebenso konnte gezeigt werden, dass die hier vorgestellte Methode im Vergleich mit dem CASPAR-Algorithmus ein etwas besseres Ergebnis, bei der Klassifikation auf dem *NEUROBLASTOM* Datensatz erreichte. Wobei die Verbesserung im Gegensatz zu den schon sehr guten Ergebnissen des CASPAR-Algorithmus vergleichsweise gering ausfiel.

Bei der Identifikation von einflussreichen Genen konnten bereits mit Krebs in Verbindung gebrachte Gene, wie *SPARC*, *COPS5* oder *KLF13*, auch in dieser Arbeit gefunden werden. Im Gegensatz dazu konnte nur eine Übereinstimmungen (*SPARC*) mit den vom CASPAR-Algorithmus bestimmten Genen gefunden werden.

In den folgenden Abschnitten werde ich kurz auf aufgetretene Probleme sowie offene Fragen eingehen. Abschließend gebe ich einen Ausblick auf weiterführende Arbeiten.

6.1 Probleme

6.1.1 Laufzeit und Komplexität

Die Laufzeit und Komplexität des Hybrid-Monte-Carlo-Algorithmus hängt sehr stark von den gewählten Parametern ab. Die Komplexität des Algortihmus beträgt $O(\tau L \delta^2)$. Wie zu sehen ist, hängt die Laufzeit linear von der Anzahl der Iterationen τ sowie von der Anzahl der durchgeführten Leapfrog-Schritte L ab. Die Dimensionalität hat *quadratischen* Einfluss auf die Komplexität und somit auch auf die Laufzeit der Anwendung.

Eine typische Parametrisierung des Algorithmus, für den Datensatz *SIMULATE* mit 500 Leapfrog-Schritten (L), 6000 Iterationen (τ) und einer Dimensionalität von $\delta = 7405$, benötigte für die Generierung der Markovkette ca. 17 Stunden. Für das selbe Setup unter Verwendung des *NEUROBLASTOM* Datensatzes mit $\delta = 10168$ benötigt die Anwendung bereits ~ 34.4 Stunden.

6.1.2 Parametrisierung

Als größtes Problem und Schwierigkeit der Arbeit stellte sich die Parametrisierung der Markovketten heraus. Ein Beispiel hierfür ist die Wahl der a-priori-Wahrscheinlichkeitsverteilung. So führte eine zu schwach gewählte a-priori-Wahrscheinlichkeitsverteilung dazu, dass die Regressionsparameter deutlich ungleich 0 wurden. Dies führte dazu, dass jedes Gen einen Einfluss auf die anschließend gesampelten Überlebenszeiten hatte. Daraus folgte eine Art *Overfitting* an den Trainingsdatensatz. Die Überlebenszeiten für Patienten des Validierungdatensatzes konnten äußerst schlecht vorausgesagt werden und eine Klassifikation in Lang- und Kurzzeitüberlebende konnte nicht erfolgreich durchgeführt werden. Anderseits konnte bei einer sehr starken a-priori-Wahrscheinlichkeitsverteilung beobachtet werden, dass diese alle Regressionsparameter, selbst die der relevanten Gene, so stark bestraft, dass diese nicht bestimmt werden konnten.

Weitere sehr einflussreiche Parameter sind die Schrittgröße ϵ sowie die Anzahl der Leapfrog-Schritte L. So spielt neben den gewählten Massen der Energiefunktion die Kombination aus Leapfrog-Schritten und Schrittgröße ϵ die größte Rolle wenn es um die gesamte Schrittweite, innerhalb einer Iteration der Markovkette, geht. Wählt man eine zu große Schrittgröße ϵ oder eine zu hohe Anzahl von Leapfrog-Schritten, so führt das zu extrem sprunghaften Verhalten der Markovkette, was die Wahrscheinlichkeit von *Ausreißern* erhöht. Im gegenteiligen Fall, also bei zu klein gewählten ϵ oder zu wenigen Leapfrog-Schritten, benötigt die Kette im besten Fall sehr lange, um den Funktionsraum ausreichend zu durchqueren. Im schlimmsten Fall konvergiert die Kette niemals.

Die Verwendung von großen Schrittgrößen sowie wenigen Leapfrogschritten ergab eine Markovkette die äußerst schlechte Werte generierte, da sie sozusagen dem Funktionsverlauf nicht exakt genug folgen konnte. Als eine gute Wahl haben sich Schrittgrößen im Bereich von $10^{-6} \geq \epsilon \geq 0.0001$ sowie 300 bis 500 Leapfrog-Schritte herausgestellt.

6.2 Offene Fragen

In wie weit sind die Geneexprssionsdaten redundant?

Es zeigte sich, dass bei unterschiedlichen Markovketten unterschiedliche Gruppen von Genen identifiziert wurden. Die Qualität der Überlebenszeitvorhersagen war jedoch ähnlich, dies könnte darauf zurückzuführen sein, dass es Gruppen von Genen gibt, die stark korreliert und koreguliert sind. Dies könnte darauf hinweisen, dass sie Anteil an ähnlichen Zellprozessen haben oder sich im selben Pathway befinden.

Aktuell wird in der Arbeitsgruppe, in der diese Arbeit entstanden ist, mit Hilfe von Markov-Chain-Monte-Carlo-Techniken versucht, genregulatorische Netzwerke zu rekunstruieren und somit einen tieferen Einblick in die zellulären Mechanismen zu erhalten.

Wie vergleicht man Überlebenszeitverteilungen mit einzelnen Überlebenszeiten?

In dieser Arbeit wurde für die Bewertung der Überlebenszeiten mit den vorhergesagten Überlebenszeitverteilungen deren Median verwendet, sowie ein Augenmerk auf die Standardabweichung gerichtet. Die Reduzierung der Überlebenszeitverteilungen auf den Median und die Standardabweichung hat den Nachteil, dass dadurch viele Informationen der Überlebenszeitverteilungen verloren gehen und dadurch nicht gewertet werden.

Es hat sich jedoch gezeigt, dass man mit der Hilfe der Verteilungsparameter (Median und Standardabweichung) ein besseres Verständnis der vorhergesagten Überlebenszeiten sowie deren Genauigkeit hat im Vergleich zu einer Vorhersage basierend auf einer einzelnen Überlebenszeit. Eine Möglichkeit für die Auswertung von Überlebenszeitverteilungen ist die Verechnung von Konfidenzintervallen.

6.3 Weiterführende Arbeiten

Es konnte gezeigt werden, dass die in dieser Diplomarbeit entwickelte Anwendung funktioniert, sowohl auf simulierten Daten als auch mit realen Daten. Der nächste Schritt ist nun eine Analyse von weiteren Datensätze, um die Ergebnisse mit anderen bereits existierenden Methoden besser vergleichen zu können.

Ein weiterer Ansatz zur Verbesserung der Überlebenszeitvorhersagen wäre die Aufnahme weiteres *Wissens* in Form von Einflussvektoren, um das Regressionsmodell weiter zu verfeinern. Hierzu könnte klinisches Wissen über die Patienten sowie bekannte Umwelteinflüsse verwendet werden.

Ebenso könnte das Programm um Möglichkeiten der automatischen Parametrisierung oder deren Anpassung erweitert werden, um den manuellen Aufwand zu reduzieren. Dies könnte sich aber als äußerst aufwändig herausstellen, da für die Parametrisierung ein gewisses Hintergrundwissen nötig ist und durch die Komplexität der zugrunde liegenden Mechanismen unvorhergesehene Effekte auftreten können.

Der hier vorgestellte Markov-Chain-Monte-Carlo Ansatz mit Bayes'schem Rückschluss hat gezeigt, dass selbst bei Problemen mit vielen Einflussgrößen, aber nur sehr wenigen Datenpunkten, eine Vorhersage der Überlebenszeit möglich ist. Neben der Modellierung von Überlebenszeiten könnte ebenso Ausfallzeiten von Geräten oder Maschinen vorhergesagt werden, um dadurch kritische Einflüsse zu identifizieren.

A Erläuterungen zu verwandten Tests und Verfahren

A.1 Acceptance-Rejection-Verfahren

Um Zufallsvariablen nach einer beliebigen Verteilungsfunktion zu erzeugen kann das Acceptance-Rejection-Verfahren angewandt werden. Diese Methode generiert Zufallsvariablen nicht direkt von der gewünschten Wahrscheinlichkeitsverteilung $f(x)$, sondern bedient sich einer Hilfsverteilungsfunktion $g(x)$, [PTVF].

So ist $f(x)$ eine beliebige Wahrscheinlichkeitsdichteverteilung von der Zufallsvariablen erzeugt werden sollen. $g(x)$ sei eine Hilfsfunktion von der auf einfache Weise Zufallszahlen gezogen werden können (z. B. Exponentialverteilung, Normalverteilung, etc.). Hierbei muss sichergestellt sein das es eine Konstante $k \in \mathbb{R}$ existiert, so dass gilt $f(x) \leq k \cdot g(x)$ für $x \in \mathbb{R}$.

Algorithmus 4 : Acceptance-Rejection-Verfahren nach Neumann [vN51]

1 Ziehe eine Zufallszahl x von der Wahrscheinlichkeitsverteilung $g(x)$
2 Ziehe eine normalverteilte Zufallszahl u
3 **if** $u < \frac{f(x)}{k \cdot g(x)}$ **then**
4 | Akzeptiere (accept) x als Zufallszahl nach $f(x)$.
5 **else**
6 | Weise x zurück (reject) und wiederhole den Vorgang.

Abbildung A.1 zeigt die Hilfsfunktion (blau), die Zielfunktion (rot) sowie die Differenz der beiden Funktionen. Das Verhältnis der Fläche unter der Hilfsfunktion zur Fläche unter der Zielfunktion entspricht dem Verhältnis von akzeptierten zu zurückgewiesenen Werten x. Deswegen ist es wünschenswert, dass die Fläche der Differenz der Funktionen (grün) möglichst gering ist, um die Anzahl zurückgewiesener Werte x zu minimieren.

Verwendete Hilfsfunktion

Um eine möglichst geringe Differenz in den Flächen der Hilfsfunktion und der Zielfunktion zu gewährleisten, benützt ich eine Exponentialfunktion der Form

$$p(y) = \lambda \exp(-\lambda y), \tag{A.1}$$

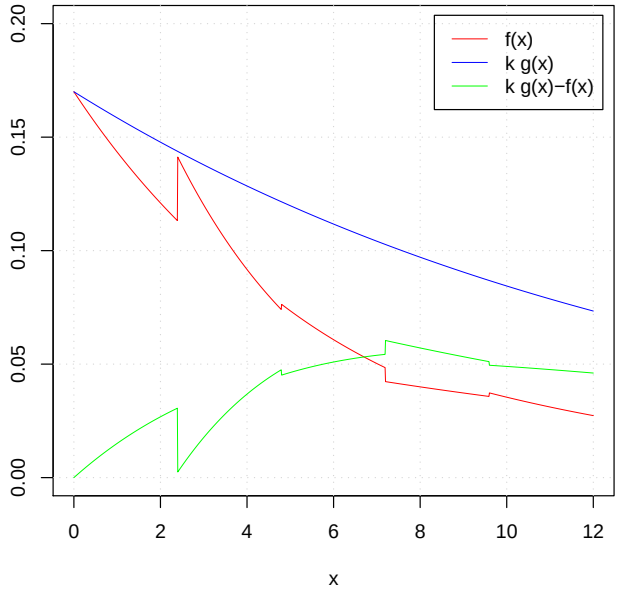


Abbildung A.1: Grafische Darstellung der Zielfunktion $f(x)$, der Hilfsfunktion $k \cdot g(x)$ sowie deren Differenz $k \cdot g(x) - (x)$

um damit Zufallszahlen zu erzeugen. Diese Zufallszahlen liegen in der Form $\frac{y}{\lambda}$ vor und müssen anschließend noch mit dem Faktor λ transformiert werden.

Als Transformationsfaktor λ habe ich einen Teil der Wahrscheinlichkeitsdichtefunktion (Gleichung 4.3) des Cox-Regressions-Modells verwendet.

$$\lambda = \min(\lambda_{Cox}) \exp(\langle \theta, x \rangle) \tag{A.2}$$

A.2 Logrank-Test

Der Logrank- oder auch Mantel-Cox-Test ist ein statistischer Hypothesentest um zwei Überlebenszeitverteilungen mit rechtszensierten Daten zu vergleichen. Er prüft die Überlebenszeitverteilungen von zwei Gruppen von Individuen sozusagen auf Ähnlichkeit.

Zur Erläuterung seien folgende Variablen definiert:

- $j = 1, \cdots, J$ seien die diskreten Zeitpunkte an denen Ereignisse (z.B. Zensur oder Tod) innerhalb der zwei zu vergleichenden Gruppen auftreten.
- N_{1j} und N_{2j} seien die Individuen der Gruppen 1 oder 2 bei denen bis zum Zeitpunkt j noch kein Ereignis eingetreten ist.
- $N_j = N_{1j} + N_{2j}$

- O_{1j} und O_{2j} seien die Anzahl von Ereignissen die zum Zeitpunkt j innerhalb der Gruppen 1 oder 2 eintreten.
- $O_j = O_{1j} + O_{2j}$

Die Verteilung der Ereignisse under der Hypothese O_{1j} wird durch eine Hypergeometrische Funktion beschrieben. Der Erwartungswert der Verteilung ist definiert durch,

$$E_j = O_j \frac{N_{1j}}{N_j}, \tag{A.3}$$

sowie die Varianz der Verteilung durch

$$V_j = \frac{O_j(N_{1j}/N_j)(1 - N_{1j}/N_j)(N_j - O_j)}{N_j - 1}. \tag{A.4}$$

Der Logrank-Test vergleicht nund jeden Wert für O_{1j} mit dem Erwartungswert E_j der Hypothese und ist defniert durch Gleichung A.5.

$$Z = \frac{\sum_{j=1}^{J}(O_{1j} - E_j)}{\sqrt{\sum_{j=1}^{J} V_j}} \tag{A.5}$$

Für eine auführliche Beschreibung der Berechnungen empfehle ich [Man66].

A.3 Kaplan-Meier-Schätzer

Kaplan-Meier-Schätzer schätzen im allgemeinen die Wahrscheinlichkeit wieder, dass ein Ereignis bis zum Zeitpunkt t noch nicht eingetreten ist. Im speziellen beschreiben sie die Wahrscheinlichkeit, dass ein Patient über den Zeitpunkt t hinaus überlebt hat [CO84].

Er ist definiert durch

$$\hat{S}(t) = \prod_{t_{(i)} \leq t} \frac{n_i - d_i}{n_i} = \prod_{t_{(i)} \leq t} \left(1 - \frac{d_i}{n_i}\right), \tag{A.6}$$

wobei d_i die Anzahl der Patienten bei denen zum Zeitpunkt $t_{(i)}$ eingetreten ist und n_i die Anzahl der sich noch unter Risiko befindlichen Patienten.

Die Abbildung A.2 zeigt einen solchen Kaplan-Meier-Plot für die gesamten *SIMULATE* Daten. In diesem Beispiel liegt die Wahrscheinlichkeit für das überleben eines Patienten von mindestens 6 Jahren hat bei ~ 0.39.

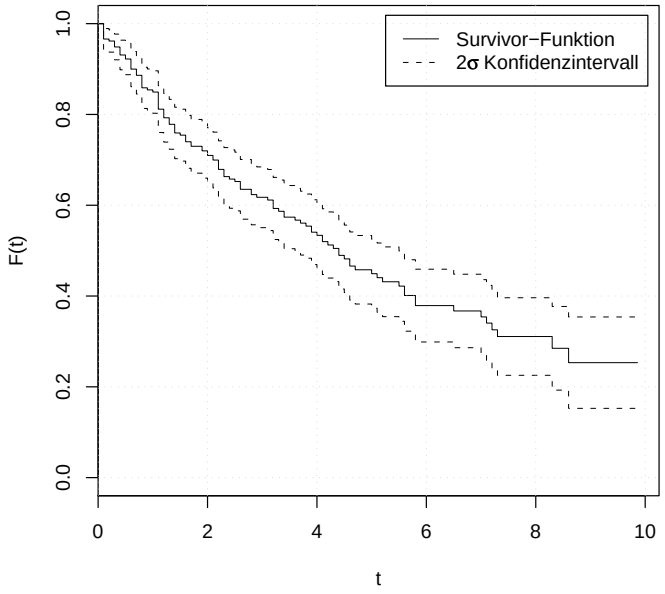


Abbildung A.2: Kaplan-Meier-Plots der gesamten *SIMULATE* Daten.

A.4 Zeitabhängige Receiver-Operator-Charakteristik

Receiver Operating Characteristic (ROC) Kurven sind eine grafische Darstellung der Sensitifität sowie der Spezifität einer zwei-Klassen Klassifikation unter Variation eines definierten *cutoff*-Wertes [Abe]. Abbildung A.3 zeigt ein Beispiel einer zeitabhängigen ROC-Kurve für $t = 5$ Jahren.

Wir gehen von zwei Klassen C_{gesund} und C_{krank} aus in die das Muster (Patient) m klassifiziert werden soll. Diese Klassifikation kann in vier Kategorien eingeteilt werden.

true positive Der Klassifikator hat den Patienten m der Klasse C_{krank} zugeordnet, und der Patient ist tatsächlich krank.

false positive Der Klassifikator hat den Patienten m der Klasse C_{gesund} zugeordnet, obwohl dieser krank ist.

true negative Der Klassifikator ordnet den Patienten der Klasse C_{gesund} zu und er ist tatsächlich gesund.

false negative Der Klassifikator hat den Patienten m der Klasse C_{gesund} zugeordnet, obwohl der Patient krank ist.

Nun variiert man für eine beobachtete Zeit $t = t_B$ den cutoff-Wert und berechnet die dadurch entstehende Sensitivität und Spezifität und trägt diese gegeneinander auf. Der gewünschte Verlauf der ROC-Kurve entspricht einer Parabel mit hoher Sensitivität und geringer Spezifität.

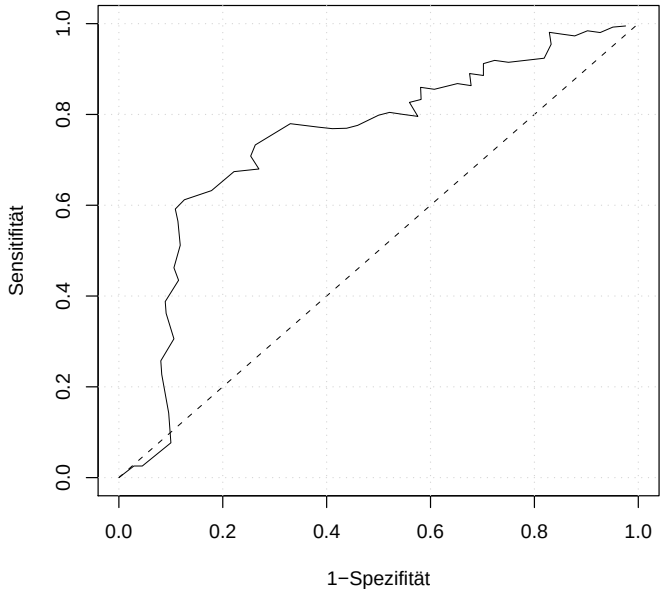


Abbildung A.3: Beispiel einer zeitabhängigen Receiver-Operator-Charakteristik

Ein weiteres Maß für Qualität einer Klassifikation ist die Fläche unter der Kurve. Durch die Variation der beobachteten Zeit $t = t_B$ entsteht ein *area under the curve*-Graph wie in Abbildung A.5. Die so erhaltene Fläche kann als Vergleichswert zu anderen Klassifikatoren herangezogen werden [DHS00]. Für eine tiefer gehende Einführung in die Analyse von Klassifikation und Überlebenszeiten empfehle ich [HZ05].

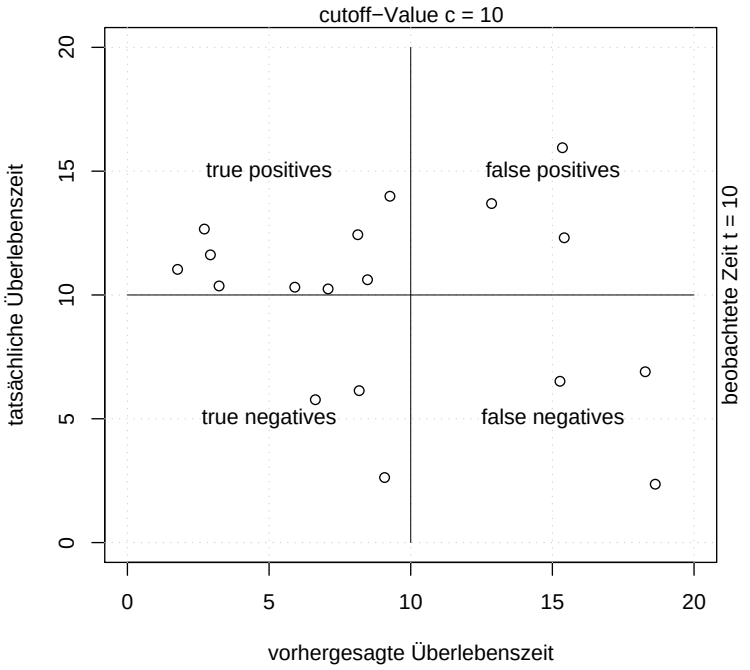


Abbildung A.4: Grafische Darstellung der ROC-Berechnungen

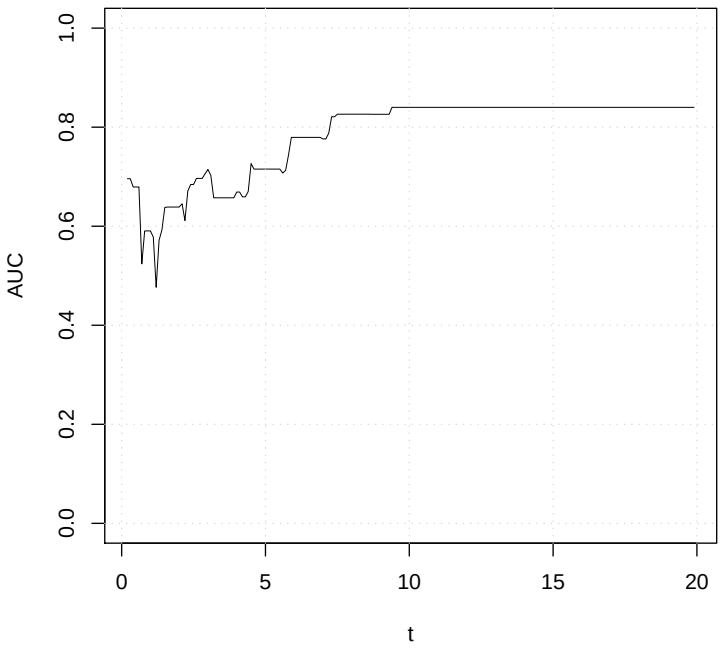


Abbildung A.5: Grafische Darstellung der *area under the curve* in Anhängigkeit der Zeit t

B Ergänzende Tabellen der ausgewerteten Markovketten

B.1 Parameter der ausgewerteten Markovketten

(a) *SIMULATE*

Parameter	**Wert**
Massen der Gewichte	0.3
Massen der Baseline Steps	0.8
Iterationen	5000
Burn-In	500
Offset	0
LQ-Prior s	0.1
LQ-Prior q	1.0
Leapfrog-Steps	500
Epsilon-A	2
Epsilon-R	0.0002
Epsilon-Reduction	10
Baseline Hazard Initialwert	0.18
Baseline Hazard Gamma-A	6.0
Baseline Hazard Gamma-R	1.08
Akzeptanzrate	0.16311
Laufzeit	$21.65h$

(b) *NEUROBLASTOM*

Parameter	**Wert**
Massen der Gewichte	0.5
Massen der Baseline Steps	0.5
Iterationen	5000
Burn-In	500
Offset	0
LQ-Prior s	0.02
LQ-Prior q	1.0
Leapfrog-Steps	500
Epsilon-A	1.1
Epsilon-R	2×10^{-6}
Epsilon-Reduction	false
Baseline Hazard Initialwert	0.15
Baseline Hazard Gamma-A	72.0
Baseline Hazard Gamma-R	10.8
Akzeptanzrate	0.4498
Laufzeit	$34.03h$

Tabelle B.1: Zusammenfassung der Parameter der zur Auswertung verwendeten Markovketten. Tabelle (a) enthält die Parameter der *SIMULATE*-Markovkette und Tabelle (b) die der *NEUROBLASTOM*-Markovkette.

B.2 Überlebenszeitvohersagen des *SIMULATE* Datensatzes

(a) Nichtzensierte Patienten.

ID	Tatsächl.	Median	Std. Abw.	Fehler
0	3.9	2.14	3.57	1.76
1	2.2	1.28	2.01	0.92
3	8.6	5.41	7.88	3.19
6	5.5	6.36	10.74	0.86
7	1.9	2.81	5.54	0.91
8	2.3	10.75	18.5	8.45
9	5.6	10.65	16.54	5.05
10	5.8	13.21	21.2	7.41
11	0.1	4.04	6.56	3.94
14	1.4	2.8	5.1	1.4
15	5.2	6.88	14.66	1.68
19	2.2	3.36	6.32	1.16
24	1.3	4.74	6.97	3.44
26	2.2	3.07	5.14	0.87
27	2.3	2.94	4.53	0.64
30	2.1	6.1	9.48	4.0
33	2.8	2.68	5.51	0.12
35	0.8	7.19	13.34	6.39
36	1.1	2.61	4.54	1.51
40	3.9	0.82	1.55	3.08
43	4.1	7.89	15.36	3.79
47	4.4	5.9	9.34	1.5
49	1.1	5.83	9.78	4.73
53	4.3	4.9	9.71	0.6
54	1.4	0.86	1.38	0.54
56	7.2	2.12	7.92	5.08
58	2.2	1.96	6.86	0.24
59	1.3	4.04	6.48	2.74
60	0.1	1.95	4.5	1.85
61	0.4	3.77	6.32	3.37
62	0.6	10.39	17.58	9.79
63	7.1	2.76	4.88	4.34
64	1.2	11.21	20.03	10.01
66	1.1	11.18	21.08	10.08
68	2.9	4.29	6.71	1.39
71	0.8	2.32	3.98	1.52
74	0.1	0.56	1.35	0.46
77	0.1	3.07	5.07	2.97

(b) Zensierte Patienten.

ID	Tatsächl.	Median	Std. Abw.	Fehler
2	2.28	28.33	44.95	≥ 0
4	2.51	16.58	32.03	≥ 0
5	7.95	3.79	7.03	≥ 4.17
12	8.46	8.57	15.73	≥ 0
13	2.21	14.92	25.85	≥ 0
16	1.08	19.54	34.12	≥ 0
17	3.09	11.51	17.24	≥ 0
18	7.98	5.9	9.12	≥ 2.08
20	0.03	3.24	8.17	≥ 0
21	2.3	11.97	18.96	≥ 0
22	5.58	10.18	17.71	≥ 0
23	3.2	18.54	38.5	≥ 0
25	0.33	32.52	79.26	≥ 0
28	0.04	2.56	3.92	≥ 0
29	6.98	14.06	27.86	≥ 0
31	5.57	9.08	13.16	≥ 0
32	5.74	14.71	25.59	≥ 0
34	7.64	8.35	13.72	≥ 0
37	2.21	9.98	16.9	≥ 0
38	4.46	3.54	6.63	≥ 0.91
39	2.54	14.29	24.14	≥ 0
41	0.05	4.11	6.9	≥ 0
42	2.89	11.63	22.78	≥ 0
44	2.13	13.36	22.24	≥ 0
45	9.36	6.49	9.96	≥ 2.86
46	4.17	7.18	12.87	≥ 0
48	0.66	7.89	12.85	≥ 0
50	2.37	23.99	37.95	≥ 0
51	1.45	1.53	2.38	≥ 0
52	0.14	2.73	5.79	≥ 0
55	4.51	14.0	25.22	≥ 0
57	7.7	14.52	21.95	≥ 0
65	5.56	2.35	5.75	≥ 3.21
67	9.86	6.98	11.15	≥ 2.87
69	7.41	5.68	9.03	≥ 1.73
70	4.69	16.31	28.41	≥ 0
72	5.07	10.09	15.53	≥ 0
73	1.57	4.76	8.58	≥ 0

Tabelle B.2: Zusammenfassung der nichtzensierten Patienten (a) und der zensierten Patienten (b).

B.3 Überlebenszeitvohersagen des *NEUROBLASTOM* Datensatzes

ID	Tatsächl.	Median	Std. Abw.	Fehler	ID	Tatsächl.	Median	Std. Abw.	Fehler
0	5.81	8.39	12.03	≥ 0	53	1.6	21.22	31.48	≥ 0
1	6.37	12.42	17.96	≥ 0	54	7.1	22.83	34.4	≥ 0
2	6.12	4.18	6.35	≥ 1.94	55	1.79	17.8	25.79	≥ 0
3	2.1	9.55	13.27	≥ 0	56	1.87	5.24	8.3	≥ 0
4	5.5	11.24	17.19	≥ 0	57	1.97	5.36	8.49	≥ 0
5	9.51	17.07	23.7	≥ 0	59	1.41	22.17	32.49	≥ 0
6	3.76	11.73	17.79	≥ 0	60	5.33	7.24	10.67	≥ 0
7	2.92	10.3	14.9	≥ 0	61	3.86	9.85	14.97	≥ 0
8	7.33	6.01	9.07	≥ 1.32	62	3.2	13.76	20.44	≥ 0
9	6.11	12.89	19.39	≥ 0	63	3.86	14.14	21.92	≥ 0
10	2.72	8.94	14.0	≥ 0	64	6.65	12.42	17.82	≥ 0
11	8.68	14.33	21.43	≥ 0	65	8.4	12.34	18.95	≥ 0
12	6.81	14.76	21.37	≥ 0	66	4.04	9.23	13.41	≥ 0
13	3.32	10.02	14.02	≥ 0	67	5.32	12.64	19.12	≥ 0
14	9.19	15.83	23.6	≥ 0	68	1.42	20.89	31.01	≥ 0
15	4.89	10.7	16.12	≥ 0	69	0.84	16.38	24.04	≥ 0
16	1.32	8.88	13.62	≥ 0	70	5.96	18.05	28.29	≥ 0
17	7.1	11.39	16.15	≥ 0	71	4.15	16.5	23.36	≥ 0
18	2.58	5.53	8.15	≥ 0	72	7.15	16.0	22.58	≥ 0
19	8.34	9.95	14.05	≥ 0	73	5.05	9.43	14.21	≥ 0
21	8.56	12.12	17.75	≥ 0	74	6.48	12.46	18.25	≥ 0
22	2.94	12.1	17.22	≥ 0	75	1.57	7.64	11.57	≥ 0
23	2.88	2.34	3.79	≥ 0.54	76	5.1	9.15	13.46	≥ 0
24	3.24	4.79	7.64	≥ 0	77	7.82	10.63	15.77	≥ 0
25	4.34	2.77	4.03	≥ 1.57	78	4.64	9.55	14.07	≥ 0
26	7.4	10.8	16.76	≥ 0	79	4.08	14.67	22.54	≥ 0
27	7.7	8.85	12.79	≥ 0	80	1.8	3.45	5.35	≥ 0
28	3.8	9.67	14.22	≥ 0	81	1.05	13.57	20.36	≥ 0
30	5.76	11.46	16.3	≥ 0	82	1.8	17.3	24.92	≥ 0
31	6.01	4.84	7.11	≥ 1.18	83	2.18	4.2	6.23	≥ 0
32	5.11	7.27	10.35	≥ 0	84	4.29	24.44	34.68	≥ 0
34	6.64	4.43	6.16	≥ 2.21	85	2.36	11.04	16.12	≥ 0
35	3.74	10.47	16.18	≥ 0	86	3.34	14.95	21.67	≥ 0
37	3.54	8.8	13.0	≥ 0	87	5.26	10.09	15.5	≥ 0
39	2.92	2.73	3.85	≥ 0.19	88	2.39	10.07	15.16	≥ 0
40	8.0	11.18	15.81	≥ 0	90	1.75	7.03	10.84	≥ 0
41	7.23	7.6	11.27	≥ 0	91	4.22	8.11	12.12	≥ 0
42	3.86	16.25	22.66	≥ 0	93	1.59	4.68	6.66	≥ 0
43	5.82	12.53	18.69	≥ 0	94	3.58	7.53	11.74	≥ 0

Tabelle B.3: Zusammenfassung der zensierten Patienten des *NEUROBLASTOM* Datensatzes.

Literaturverzeichnis

[Abe] U. Abel. *Die Bewertung diagnostischer Tests*. Hippokrates Verl.

[AdFDJ03] C. Andrieu, N. de Freitas, A. Doucet, and M.I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1):5–43, 2003.

[ALL+08] A.S. Adler, L.E. Littlepage, M. Lin, T.L.A. Kawahara, D.J. Wong, Z. Werb, and H.Y. Chang. CSN5 Isopeptidase Activity Links COP9 Signalosome Activation to Breast Cancer Progression. *Cancer Research*, 68(2):506, 2008.

[BDCS08] M BBettencourt-Dias and Z. Carvalho-Santos. Double life of centrioles: CP110 in the spotlight. *Trends Cell Biology*, 18(1):8–11, 2008.

[BGMS03] AD Bradshaw, DC Graves, K. Motamed, and EH Sage. SPARC-null mice exhibit increased adiposity without significant differences in overall body weight. *Proceedings of the National Academy of Sciences*, 100(10):6045, 2003.

[CCW+03] J.T. Chi, H.Y. Chang, N.N. Wang, D.S. Chang, N. Dunphy, and P.O. Brown. Genomewide view of gene silencing by small interfering RNAs. *Proceedings of the National Academy of Sciences*, 100(11):6343, 2003.

[CLW+06] Y. Che, A. Luo, H. Wang, J. QI, J. GUO, and Z. LIU. The differential expression of SPARC in esophageal squamous cell carcinoma. *INTERNATIONAL JOURNAL OF MOLECULAR MEDICINE*, 17(6):1027, 2006.

[CO84] D.R. Cox and D. Oakes. *Analysis of Survival Data*. Chapman & Hall/CRC, 1984.

[Cox72] DR Cox. Regression Models and Life-Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

[CQL] L. Chen, Z. Qin, and J.S. Liu. Exploring Hybrid Monte Carlo in Bayesian Computation. *sigma*, 2:2–5.

[DHS00] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[DKPR87] S. Duane, AD Kennedy, B.J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

[Gam97] D. Gamerman. Markov Chain Monte Carlo: Stochastic Simulation of Bayesian Inference. 1997.

[GGA+93] S. Geman, D. Geman, K. Abend, TJ Harley, and LN Kanal. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*. *Journal of Applied Statistics*, 20(5):25–62, 1993.

[GHRPS90] A.E. Gelfand, S.E. Hills, A. Racine-Poon, and A.F.M. Smith. Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

[GRS96] W.R. Gilks, S. Richardson, and DJ Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1996.

[GS90] A.E. Gelfand and A.F.M. Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[HZ05] P.J. Heagerty and Y. Zheng. Survival Model Predictive Accuracy and ROC Curves. *Biometrics*, 61(1):92–105, 2005.

[KA07] A.M. Krensky and Y.T. Ahn. Mechanisms of Disease: regulation of RANTES (CCL5) in renal disease. *NATURE CLINICAL PRACTICE NEPHROLOGY*, 3(3):164, 2007.

[Kad06] L. Kaderali. *A Hierarchical Bayesian Approach to Regression and its Application to Predicting Survival Times in Cancer*. Shaker, 2006.

[Kim03] VN Kim. RNA interference in functional genomics and medicine. *J Korean Med Sci*, 18(3):309–18, 2003.

[Kni06] Rolf Knippers. *Molekulare Genetik*. Thieme, 2006.

[KR80] J.D. Kalbfleisch and L. Ross. *The statistical analysis of failure time data*. Wiley New York, 1980.

[KZF+06] L. Kaderali, T. Zander, U. Faigle, J. Wolf, J.L. Schultze, and R. Schrader. CASPAR: a hierarchical bayesian approach to predict survival times in cancer from gene expression data. *Bioinformatics*, 22(12):1495, 2006.

[LYS+04] T. Liu, J.Q. Yin, B. Shang, Z. Min, H. He, J. Jiang, F. Chen, Y. Zhen, and R. Shao. Silencing of hdm2 oncogene by siRNA inhibits p53-dependent human breast cancer. *Cancer Gene Therapy*, 11:748–756, 2004.

[Man66] N. Mantel. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemother Rep*, 50(3):163–70, 1966.

[MRI+07] S.S. Müerköster, AV Rausch, A. Isberner, J. Minkenberg, E. Blaszczuk, M. Witt, UR Fölsch, F. Schmitz, H. Schäfer, and A. Arlt. The apoptosis-inducing effect of gastrin on colorectal cancer cells relates to an increased IEX-1 expression mediating NF-kappa B inhibition. *Oncogene*, 2007.

[Nea96] R.M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996.

[OBW+06] A. Oberthuer, F. Berthold, P. Warnat, Y. Hero, B. Kahlert, R. Spitz, K. Ernestus, R. Kœnig, S. Haas, R. Eils, M. Schwab, B. Brors, F. Westermann, and M. Fischer. Customized oligonucleotide microarray gene expression-based classification of neuroblastoma patients outperforms current clinical risk stratification. *J. Clin. Oncol.*, 24(31):5070–5078, Nov 2006.

[OKK+08] A. Oberthuer, L. Kaderali, Y. Kahlert, B. Hero, F. Westermann, F. Berthold, B. Brors, R. Eils, and M. Fischer. Sub-classification and individual survival time prediction from gene-expression data of neuroblastome patients using "CASPAR". 2008.

[PNKS08] S. Prasad, n. Nigam, N. Kalra, and Y. Shukla. Regulation of signaling pathways involved in lupeol induced inhibition of proliferation and induction of apoptosis in human prostate cancer cells. *Mol Carcinog*, 2008.

[PTVF] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical Recipes in C++.

[Rit07] Daniel Ritter. Markov Chain Monte Carlo Methods. *VIROQUANT Modeling and Discrete Optimization*, 2007.

[vN51] J. von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12:36–38, 1951.

[VSS+07] K. Voss, S. Stahl, E. Schleider, S. Ullrich, J. Nickel, T.D. Mueller, and U. Felbor. CCM3 interacts with CCM2 indicating common pathogenesis for cerebral cavernous malformations. *neurogenetics*, 8(4):249–256, 2007.

[WDJB+05] G. Watkins, A. Douglas-Jones, R. Bryce, R. E Mansel, and W.G. Jiang. Increased levels of SPARC (osteonectin) in human breast cancer tissues and its association with clinical outcomes. *Prostaglandins, Leukotrienes & Essential Fatty Acids*, 72(4):267–272, 2005.

[WLC+04] CS Wang, KH Lin, SL Chen, YF Chan, and S. Hsueh. Overexpression of SPARC gene in human gastric carcinoma and its clinic-pathologic significance. *Br J Cancer*, 91(11):1924–30, 2004.

[YCL+06] M.Y. Yang, J.G. Chang, P.M. Lin, K.P. Tang, Y.H. Chen, H.Y.H. Lin, T.C. Liu, H.H. Hsiao, Y.C. Liu, and S.F. Lin. Downregulation of circadian clock genes in chronic myeloid leukemia: Alternative methylation pattern of hPER3. *Cancer Science*, 97(12):1298–1307, 2006.

[YWS+] M. Yokouchi, T. Wakioka, H. Sakamoto, H. Yasukawa, S. Ohtsuka, A. Sasaki, M. Ohtsubo, M. Valius, A. Inoue, S. Komiya, et al. APS, an adaptor protein containing PH and SH2 domains, is associated with the PDGF receptor and c-Cbl and inhibits PDGF-induced mitogenesis.

[ZCS+07] XC Zhang, J. Chen, CH Su, HY Yang, and MH Lee. Roles for CSN5 in control of p53/MDM2 activities. *J Cell Biochem*, 2007.