



Stochastische Modellierung und numerische Minimierung in der Immunologie

Thomas Stolte

Zentrum Mathematik
Technische Universität München
D-85747 Garching
2007



Zentrum Mathematik
Lehrstuhl für Numerische Mathematik
der Technischen Universität München

Stochastische Modellierung und numerische Minimierung in der Immunologie

THOMAS STOLTE

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor der Naturwissenschaften (Dr. rer. nat)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. G. Kemper

Prüfer der Dissertation:

1. Univ.-Prof. Dr. P. Rentrop
2. Univ.-Prof. Dr. J. Müller
3. Univ.-Prof. Dr. M. Kiehl, Technische Universität Darmstadt

Die Dissertation wurde am 04.12.2006 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 12.02.2007 angenommen.

Danke

Zum Gelingen dieser Arbeit haben einige Personen besonders beigetragen, bei denen ich mich herzlich bedanken möchte.

An erster Stelle ist Prof. Peter Rentrop zu nennen. Er bot mir bereits im Rahmen meiner Diplomarbeit an der Universität Karlsruhe die Gelegenheit, mich in die Biomathematik einzuarbeiten. Anschließend ermöglichte er mir als Mitglied seiner Arbeitsgruppe an der TU München dieses Gebiet in einer Promotion vertiefend zu bearbeiten. Seine konstruktive Kritik und fachlichen Anregungen waren mir eine große Hilfe. Ein herzliches Dankeschön geht auch an Prof. Johannes Müller, meinem zweiten Betreuer. Er hat sich von Anfang an als ein offener und interessierter Gesprächspartner erwiesen und war für mich eine große Unterstützung.

Des Weiteren möchte ich mich bei meinen Kooperationspartnern bedanken, die mir ihr Datenmaterial zur Verfügung stellten und in anregenden Gesprächen ihre Fachkenntnisse vermittelten: Dr. Michael Speicher vom „Klinikum rechts der Isar der Technischen Universität München“ aus dem Fachgebiet Humangenetik, Dr. Volker Hösel von der „GSF - Forschungszentrum für Umwelt und Gesundheit“ am Institut für Biomathematik und Biometrie (IBB), Dr. Stefan Tenzer von der Johannes-Gutenberg-Universität Mainz am Institut für Immunologie, Ph.D. Alexander K. Nussbaum am Institut Curie, Centre de Recherche in Paris, Dr. Christina Kuttler von der „GSF - Forschungszentrum für Umwelt und Gesundheit“ am Institut für Biomathematik und Biometrie (IBB) und Prof. Klaus Dietz von der Eberhard-Karls-Universität Tübingen am Institut für Medizinische Biometrie.

Mein Dank geht auch an meine Kolleginnen und Kollegen am Institut M2 in München für viele interessante Gespräche und gemeinsame Kaffeerunden.

Besonders dankbar bin ich für den Rückhalt, den mir meine Freunde gegeben haben. Trotz großer Entfernungen waren sie immer für mich da, lieferten hin und wieder sehr willkommene Ablenkungen, hatten immer ein offenes Ohr und sorgten bei Bedarf für den Motivationsaufbau.

Unschätzbar war die Hilfe meiner Großfamilie. Sie hat mich in den vergangenen Jahren durch alle Hochs und Tiefs begleitet und jederzeit unterstützt - sei es durch fleißi-

ge Korrekturen, fruchtbare Gespräche und Informationen zu den mir weitgehend unbekanntem Fachgebieten der Biochemie.

Danken möchte ich auch meinen drei Hauptkorrektoren Frank Reinbold, Andrea Gießen und Jessika Herrmann, die mich mit viel Geduld um die Stolperfallen der deutschen Sprache gelotst haben.

Den größten Beitrag leisteten meine Eltern Helga und Hans Stolte, denen ich diese Arbeit widme. Sie haben mich stets unterstützt und zur rechten Zeit immer wieder daran erinnert, was im Leben wirklich zählt.

Thomas Stolte

München, 01.12.2006

Inhaltsverzeichnis

Danke	i
Einleitung	1
1 Einführung in die Fallstudien aus der Immunologie	3
1.1 Fallstudie 1: Die Entwicklung von Mutationen in prekanzerösen Zellen . . .	3
1.2 Fallstudie 2: Die Proteinspaltung durch das Enzym Proteasom	9
1.3 Fallstudie 3: Immunisierungsstrategie gegen Hepatitis in Krankenhäusern	14
1.4 Ausblick	18
2 Mathematische Grundlagen	19
2.1 Zur Statistischen Datenanalyse	19
2.2 Zur Singulärwertzerlegung	30
2.3 Zur Maximum-Likelihood-Methode	31
2.4 Zur Minimierung	33
3 Entwicklung von Mutationen in prekanzerösen Zellen	47
3.1 Datenanalyse	48
3.2 Implementierung und Ergebnisse	50
3.3 Medizinische Interpretation der Ergebnisse	58
3.4 Zusammenfassung	61

4 Proteinspaltung durch das Enzym Proteasom	63
4.1 Modellanalyse	63
4.2 Datenanalyse	67
4.3 Implementierung und Ergebnisse	72
4.4 Ausblick	74
5 Immunisierungsstrategie gegen Hepatitis in Krankenhäusern	77
5.1 Modellanalyse	77
5.2 Datenanalyse	79
5.3 Implementierung und Ergebnisse	81
5.4 Statistische Modellanalyse und Ausblick	82
6 Die Immunisierungsstrategie als Kontrollproblem	85
6.1 Einführung in die Stochastische Programmierung	85
6.2 Kontrollproblem	91
6.3 Berechnung einer Steuerstrategie	103
6.4 Beispiel Patient 35	105
Zusammenfassung	107
Anhang	109
Anhang: Mutation	109
Anhang: Proteasom	110
Anhang: Immunisierungsstrategie	111
Glossar	113
Literaturverzeichnis	117

Einleitung

Die Forschungsaktivitäten in der Immunologie haben in den vergangenen Jahren stark zugenommen. Aufgrund der Komplexität und Weiträumigkeit dieses Gebiets treffen hier verschiedene wissenschaftliche Disziplinen aufeinander, insbesondere die Medizin und die Naturwissenschaften Biologie, Chemie und Physik. Bisher spielt die Mathematik in der Immunologie eine eher untergeordnete Rolle, obwohl ihre Verfahren einen wertvollen Beitrag zur immunologischen Forschung leisten könnten. Aktuell lassen sich u.a. folgende Anwendungsfelder für die Mathematik identifizieren:

Innerhalb der einzelnen Fachbereiche kann die Mathematik als Abstraktionswerkzeug dienen. Auf diese Weise ist es möglich, mathematische Verfahren aus anderen Bereichen wie zum Beispiel dem Finanz- oder Ingenieurwesen in die Immunologie zu übertragen und nutzbringend einzusetzen.

Ein weiteres Problem, das mit Hilfe der Mathematik gelöst werden kann, ist die Datenflut. In der Immunologie werden ständig neue Analyseverfahren entwickelt und angewandt, dabei werden so viele bzw. so große Datensätze generiert, dass diese mit den herkömmlichen heuristischen Verfahren nicht mehr ausgewertet werden können. Mit mathematisch fundierten Analyseverfahren ist es möglich, diese Datenmengen zu beherrschen und sinnvoll zu interpretieren. Außerdem werden die Auswahlkriterien somit transparenter und die Auswertungen objektiviert.

Um steuernd in immunologische Prozesse eingreifen zu können, ist es notwendig, verlässliche Vorhersagen zu treffen. Bisher gründen sich diese Vorhersagen vornehmlich auf Experimente, die aber oft sehr kosten- und zeitintensiv sind. Außerdem gibt es viele Sachverhalte, die derart komplex sind, dass sie sich nicht in praktischen Experimenten abbilden lassen. Mit numerischen Simulationen ist es der Mathematik möglich, auch komplexe Gegebenheiten abzubilden sowie schnell und kostensparend belastbare Vorhersagen zu treffen.

In der vorliegenden Arbeit werden diese Anwendungsmöglichkeiten der Mathematik anhand dreier Fallstudien exemplarisch dargestellt:

In Kapitel 1 wird zunächst ein Einblick in die jeweilig relevanten biochemischen und immunologischen Vorgänge gegeben, welche die Grundlage für eine mathematische

Modellierung bilden. Nachdem die Fallstudien in mathematische Formulierungen übertragen worden sind, werden diese zu Parameteridentifikationen als Optimierungsprobleme unter Restriktionen klassifiziert.

In Kapitel 2 werden anschließend die mathematischen Hilfsmittel vorgestellt. Diese unterteilen sich in statistische und analytische Werkzeuge zur Modellierung und Analyse sowie in Numerische Verfahren zur Parameteridentifikation.

Der Schwerpunkt der Arbeit liegt auf der separaten Analyse der drei Fallstudien und der Vorstellung der jeweiligen Lösungsverfahren: In Kapitel 3 wird mittels eines Populationsmodells die „Entwicklung von Mutationen in prekanzerösen Zellen“ als ein Optimierungsproblem unter Differentialgleichungsrestriktionen beschrieben. Um die Daten reduzieren zu können, wird eine Scoring-Funktion eingeführt. Danach wird ein Lösungsverfahren für große lineare Differentialgleichungssysteme mit konstanter unterer Dreiecksmatrix vorgestellt, welches mit bekannten Lösungsansätzen verglichen wird. Im Anschluss daran werden die Ergebnisse visualisiert, diskutiert und interpretiert.

In Kapitel 4 wird gezeigt, dass man aus der „Proteinspaltung durch das Enzym Proteasom“ durch einen Binomialen Ansatz ein Optimierungsproblem unter Ungleichungsrestriktionen erhält. Anhand einer formellen Untersuchung kann dargelegt werden, dass sich die Modellfunktion vereinfachen und so der Rechnungsaufwand wesentlich reduzieren lässt. Analysiert man das vereinfachte Modell, wird ein direkter Zusammenhang der Singulärwerte sowohl mit den Eingabedaten als auch mit den gesuchten Parametern deutlich, der in einem Lemma bewiesen wird. Abschließend werden erste Ergebnisse vorgestellt, darüber hinaus wird ein Ausblick auf das weitere Vorgehen und das langfristige Ziel gegeben.

Die Kapitel 5 und 6 bearbeiten daraufhin die Fallstudie „Immunisierungsstrategie gegen Hepatitis in Krankenhäusern“. Hierbei wird zunächst in Kapitel 5 das aus einem Normalverteilungsansatz gewonnene Optimierungsproblem mit Ungleichungsrestriktionen gelöst. Die Ergebnisse werden anschließend einer Hauptkomponentenanalyse unterzogen, deren Resultate darauf hindeuten, dass es möglich ist, das Modell zu vereinfachen. Ob diese Vermutung zutrifft, müsste durch weiterführende Untersuchungen geklärt werden.

In Kapitel 6 erfolgt eine Einführung in die Kontrolltheorie. Die Fallstudie „Immunisierungsstrategie“ wird mit Hilfe der stochastischen Programmierung in ein Kontrollproblem überführt, das im kontinuierlichen Fall auf Konsistenz und Eindeutigkeit untersucht wird. Unter Berücksichtigung einschränkender Annahmen wird der konvergente Fall diskutiert und die Existenz einer optimalen Lösung nachgewiesen. Anschließend wird ein numerisches Verfahren vorgestellt, mit dem sich die optimale Lösung bestimmen lässt. Zum Schluss finden die vorgestellten Verfahren beispielhaft eine Anwendung auf dem konkreten Fall eines Patienten.

Kapitel 1

Einführung in die Fallstudien aus der Immunologie

Die vorliegende Arbeit basiert auf drei Fallstudien. Diese sollen zunächst vorgestellt und die jeweils relevanten biochemischen und immunologischen Vorgänge beschrieben werden, auf deren Grundlage die mathematische Modellierung erfolgt. Die folgenden Erläuterungen verwenden die entsprechenden naturwissenschaftlichen Fachbegriffe. Zur besseren Lesbarkeit finden sich deren Definitionen nicht im Text sondern im Glossar.

1.1 Fallstudie 1: Die Entwicklung von Mutationen in prekanzerösen Zellen

Eine der größten Herausforderungen in der Cytogenetik liegt darin, Wechselbeziehungen zwischen der Krebsentwicklung und den chromosomalen Veränderungen in den anomalen Zellen aufzudecken. Bei der Zellteilung kann es zu drei verschiedenen Abweichungen kommen: Teile der Chromosomen können verloren gehen, neue Teile können hinzukommen oder das Chromosom kann sich verdoppeln. Die Analyse der Chromosomen erfolgt häufig mittels der Fluoreszenzinsituhybridation (FISH). Diese Technik verfügt über eine Auflösung, die den meisten konkurrierenden experimentellen Techniken überlegen ist, und bietet außerdem eine automatisierte Bildanalyse. Es existieren einige verfeinerte Versionen von FISH. Dazu gehört u. a. die Multi-Color-FISH (M-FISH) (Abbildung 1.1), die einige Arten von Abweichungen gleichzeitig abbilden kann. Mit dieser Technik werden Chromosome farbig markiert, so dass sich Stücke eines Chromosoms detektieren lassen, die in andere Chromosomen eingesetzt wurden. Auch fehlende Teile eines Chromosoms oder die Chromosomen, die dreifach vorliegen, können lokalisiert werden [SPE96], [SPE96/2].

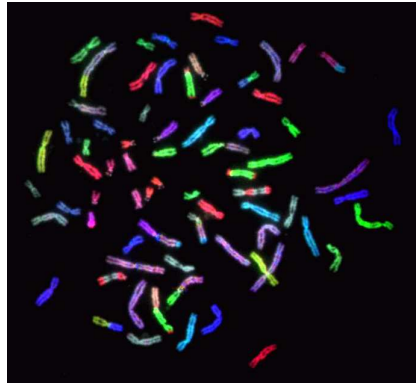


Abbildung 1.1: Beispiel einer M-FISH Analyse.

Um die Mechanismen zu untersuchen, die zur Krebskrankheit führen, wird gutartiges Brustgewebe *in vitro* durch Krebs erzeugende Substanzen [FAU04] angeregt. In der Folge kommt es vermehrt zu Mutationen, die sich in der gesamten Zellpopulation ausbreiten. In festgelegten Zeitabständen werden viermal Beispielzellen nach dem Zufallsprinzip entnommen. Ihre Abweichungen werden mittels M-FISH sichtbar gemacht und anschließend klassifiziert. Hieraus erhält man einen unstrukturierten und schwierig zu interpretierenden Datensatz, dessen Analyse normalerweise durch einen Experten erfolgt.

Zum einen ist dieses Verfahren sehr zeitintensiv, zum anderen birgt die heuristische Analyse die Gefahr, dass man sich auf irrelevante Details des Datensatzes konzentriert und wichtige Eigenschaften übersieht.

Im Folgenden wird ein Modell vorgestellt, mit dem es möglich ist, auf objektive Art Einblicke in die innere Struktur eines M-FISH-Datensatzes zu gewinnen. Die wichtigsten Fragen lauten dabei: Warum treten instabile Zelllinien nur zu bestimmten Zeiten auf? Welche Mutationen führen zu einem instabilen Zelltyp?

Das Modell beruht auf dem Standardmodell der Populationsgenetik (siehe [CRO70, HOF85, BUE00, HER02]). Hier wird angenommen, dass der Genotyp jedes Individuums aus einer begrenzten Menge von Möglichkeiten stammt, die *a priori* spezifiziert sind. Diese Einschränkung der Möglichkeiten lässt sich *a posteriori* rechtfertigen, wenn man reale Daten betrachtet: Während eines bestimmten Zeitintervalls ist nur eine begrenzte Anzahl von Typen zu beobachten.

Im Standardmodell wird angenommen, dass jeder Typ seine eigene Reproduktionsrate hat. Interaktionen, wie die Konkurrenz um Ressourcen, finden nicht statt. Es erscheint gerechtfertigt, dieses Modell anzuwenden, da die analysierten Daten in einem Experiment erzeugt werden, in dem genügend Nährstoffe zur Verfügung stehen, um insbesondere direkte Konkurrenzeffekte auszuschließen. Problematisch ist allerdings, dass das Standardmodell entwickelt wurde, um die grundlegenden Effekte der

Bevölkerungsgenetik zu untersuchen, nicht aber um reale Datensätze zu analysieren. Die Anzahl der üblicherweise für das Standardmodell benötigten Parameter ist bei weitem zu groß, um diese zuverlässig aus einem bearbeitbaren Datensatz zu bestimmen. Bei der Anpassung des Modells liegt der Schwerpunkt deshalb darauf, die Kodierung der unterschiedlichen Zelltypen durch Gruppen von Mutationen und das Verhalten der einzelnen Mutationen so zu formulieren, dass die Zahl der Parameter möglichst klein bleibt. Wichtige Eigenschaften des Modells, die für die Beschreibung der Daten notwendig sind, müssen aber erhalten bleiben. Erste Simulationsergebnisse zeigen, dass es nicht notwendig ist, die Interaktionen zwischen Mutationen zu modellieren, um die wesentliche Struktur der Daten zu reproduzieren. Ohne die Interaktionen reduziert sich die Anzahl der Parameter wesentlich und entspricht der Anzahl der Mutationen (Abarrationen).

Man nehme an, dass es höchstens l unterschiedliche Arten von Mutationen der Chromosomen in jeder Zelle gibt. Eine Mutation vom Typ i wird als vorhanden in einer Zelle gezählt, wenn sie mindestens in einem Chromosom eines diploiden Paares auftritt. Fehlt ein Chromosom, existiert also nur die Kopie eines der Chromosome, oder sind während der Zellteilung mehrfache Kopien entstanden, werden diese nicht gesondert betrachtet, da immer eine funktionierende Kopie vorhanden ist. Jede Zelle kann folglich durch einen binären Vektor $\vec{n} = (n_1, \dots, n_l) \in \{0, 1\}^l$ gekennzeichnet werden, wobei eine Komponente $n_i = 1$ anzeigt, dass eine Mutation vom Typ i vorliegt.

Nun lässt sich ein dynamisches Populationsmodell für die Zellvermehrung entwickeln. Hierbei müssen zwei Prozesse betrachtet werden: Das Populationswachstum und die Dynamik der Mutationen.

Für den ersten Prozess sei angenommen, dass die Vermehrung der Zellen ohne Mutationen durch eine positive Reproduktionsrate r_0 gegeben ist. Betrachtet man eine Zelle mit nur einer Mutation, d.h. vom Typ $\vec{n}_i = (n_j)_{j=1, \dots, l}$ mit $n_j = \delta_{j,i}$ für $j \in \{1, \dots, l\}$, so wird die Reproduktionsrate r_i dieser Zelle im Allgemeinen nicht mit der von Zellen ohne Mutationen übereinstimmen, d.h. man sucht ein r_i , so dass die Gesamtreproduktionsrate der Zelle gleich $r_0 + r_i$ ist (i.A. $r_i \neq 0$). Um die Anzahl der Parameter beherrschbar zu halten, werden die Interaktionen von Mutationen nicht berücksichtigt. Weiterhin wird angenommen, dass das Vorhandensein verschiedener Mutationen in der gleichen Zelle die Reproduktionsrate durch lineare Superposition ändert. Die Gesamtreproduktionsrate $r(\vec{n})$ einer Zelle vom Typ \vec{n} lässt sich dann berechnen mit

$$r(\vec{n}) = r_0 + \sum_{i=1}^l n_i r_i.$$

Der zweite Prozess modelliert die Dynamik des Auftretens der Mutationen. Hierfür wird angenommen, dass die Mutationsrate einer Zelle unabhängig von ihrem Typ \vec{n} sei.

Mit a_j sei die Rate für das Auftreten einer neuen Mutation j benannt, also die Rate für den Wechsel einer Zelle vom Typ \vec{n} mit $n_j = 0$ zu einer Zelle vom Typ \vec{n} mit $\tilde{n}_i = n_i + \delta_{i,j}$. Da Mutationen nur auftreten, aber nicht verschwinden können, gilt: $a_j \geq 0$. Rückmutationen werden somit vernachlässigt.

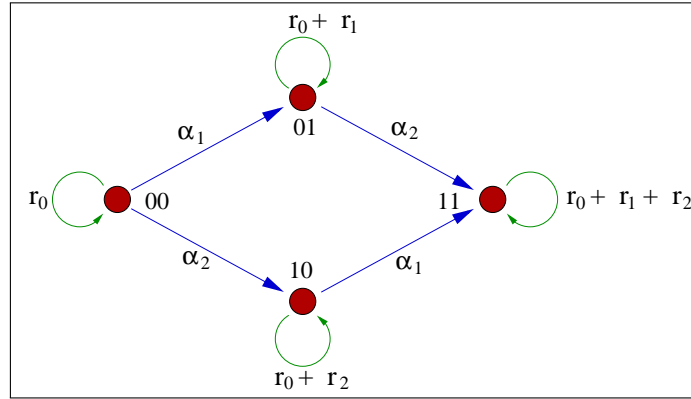


Abbildung 1.2: Zelltypen mit ihren Übergängen für zwei Mutationen.

In Abbildung 1.2 sind die Übergänge der einzelnen Zelltypen am Beispiel einer Population mit zwei verschiedenen Mutationen dargestellt.

$x_{\vec{n}}$ bezeichne die Anzahl der Zellen vom Typ \vec{n} . Man beachte, dass die Funktion $x_{\vec{n}}$ von der Zeit t und den Parametern abhängt. Weiter sei mit $h(\vec{n}, \vec{\bar{n}})$ der Hamming-Abstand zwischen den binären Vektoren \vec{n} und $\vec{\bar{n}}$ definiert:

$$h(\vec{n}, \vec{\bar{n}}) := \sum_{i=1}^l |n_i - \bar{n}_i|$$

und es gelte $\vec{n} \geq 0$ für einen Vektor \vec{n} , wenn jede Komponente nicht negativ ist. Man beachte, dass dadurch eine Halbordnung der Vektoren eingeführt wird:

$$\vec{n} \geq \vec{\bar{n}} \Leftrightarrow \vec{n} - \vec{\bar{n}} \geq 0.$$

Diese Definitionen und Annahmen machen es nun möglich, folgendes System von gewöhnlichen Differentialgleichungen (DGL) aufzustellen:

$$\dot{x}_{\vec{n}} = \underbrace{\left(r_0 + \sum_{i=1}^l n_i r_i - \sum_{i=1}^l (1 - n_i) \alpha_i \right)}_{\text{Term I}} x_{\vec{n}} + \underbrace{\left(\sum_{\vec{\bar{n}} \leq \vec{n}, h(\vec{n}, \vec{\bar{n}})=1} x_{\vec{\bar{n}}} \sum_{i=1}^l \alpha_i ((n_i - \bar{n}_i)) \right)}_{\text{Term II}},$$

$$x_{\vec{n}}(0) = x_{\vec{n}}^0 \quad \forall \vec{n} \in \{0, 1\}^l.$$

Term *I* gibt die Reproduktion und das Fortmutieren wieder und Term *II* entspricht dem Zuwachs von neuen Zellen durch Mutationen aus anderen Zellen eines anderen Zelltyps. Hier erfüllen die Anfangswerte die Bedingung $x_{\vec{n}}^0 \geq 0$ für alle Zelltypen \vec{n} .

Allgemein ist man nicht an der Populationsgröße selbst interessiert, sondern an der relativen Häufigkeit der verschiedenen Zelltypen:

$$y_{\vec{n}}(t) = \frac{x_{\vec{n}}(t)}{\sum_{\vec{n} \in \{0,1\}^l} x_{\vec{n}}(t)}.$$

Demnach kann man $x_{\vec{n}}(t)$ durch $\tilde{x}_{\vec{n}} = e^{-r_0 t} x_{\vec{n}}$ ersetzen, was die übermäßige Menge der mutierten Zellen vom Typ \vec{n} mit der Menge der nicht mutierten Zellen vergleicht, d.h., die für alle Zelltypen identische Grundreproduktionsrate entfällt aus der DGL. Somit gilt:

$$y_{\vec{n}}(t) = \frac{x_{\vec{n}}(t)}{\sum_{\vec{n} \in \{0,1\}^l} x_{\vec{n}}(t)} = \frac{\tilde{x}_{\vec{n}}(t)}{\sum_{\vec{n} \in \{0,1\}^l} \tilde{x}_{\vec{n}}(t)}$$

und man erhält das neue DGL-System:

$$\begin{aligned} \dot{\tilde{x}}_{\vec{n}} &= \left(\sum_{i=1}^l n_i r_i - \sum_{i=1}^l (1 - n_i) \alpha_i \right) \tilde{x}_{\vec{n}} + \left(\sum_{\vec{n} < \vec{h}, h(\vec{n}, \vec{n})=1} \tilde{x}_{\vec{n}} \sum_{i=1}^l \alpha_i (n_i - \bar{n}_i) \right), \\ \tilde{x}_{\vec{n}}(0) &= \tilde{x}_{\vec{n}}^0 \quad \forall \vec{n} \in \{0,1\}^l. \end{aligned}$$

Zwei Anfangsbedingungen mit $\tilde{x}_{\vec{n}} = \zeta \tilde{x}'_{\vec{n}}$ für $\zeta > 0$ und alle $\vec{n} \in \{0,1\}^l$ ergeben dieselbe relative Häufigkeit. Demnach lassen sich oBdA die $x_{\vec{n}}^0$ folgendermaßen wählen:

$$x_{\vec{n}}^0 \geq 0, \quad \sum_{\vec{n} \in \{0,1\}^l} x_{\vec{n}}^0 = 1.$$

Zur Vereinfachung wird im Folgenden wieder $x_{\vec{n}}$ für $\tilde{x}_{\vec{n}}$ geschrieben. Dieses Modell hat $2l+2^l-1$ Parameter: l Reproduktions- und l Mutationsraten und 2^l-1 Parameter für die Anfangsbedingungen.

Um die große Anzahl von Parametern noch weiter zu reduzieren, wird angenommen, dass zum Zeitpunkt Null keine Mehrfach-Mutationen in der Population vorliegen. In diesem Fall reduziert sich die Anzahl der Anfangsbedingungen auf l : Es gilt $l+1$ Parameter für die relative Häufigkeit des Typs $\vec{n}^0 := (0, \dots, 0)$ (keine Mutationen) und \vec{n}^i mit $n_j^i := \delta_{i,j}$ (genau eine Mutation) $i = 1, \dots, l$ zu bestimmen. Da sich aber die Summe der relativen Häufigkeiten auf Eins beläuft, reduziert der Freiheitsgrad wieder um Eins. Insgesamt sind also nun $3l$ Parameter nötig.

Das Modell ist eine erste Approximation an die komplexen Mechanismen, die bei den Mutationen innerhalb einer Zelle ablaufen. Verfeinerungen des Modells könnten nötig sein, so sind z.B. die Abhängigkeiten der Mutationen untereinander unter Berücksichtigung ihrer Reproduktions- und Mutationsraten zu beachten. Inwieweit dieses einfache Modell Vorhersagen bezüglich der Haupteffekte erlaubt, müssen die Anpassungen an den Datensatz zeigen. Hierfür ist also eine Parameterschätzung nötig.

Zunächst gilt es, die Zeit-durchlaufende Stichproben einer Population, zu betrachten. Dies bedeutet, die Proben s mit $s = 1, \dots, k$ wurden zum Zeitpunkt t_s mit $t_1 < \dots < t_k$ genommen. Die Anzahl der entnommenen Zellen der Probe s ist m_s . Deren Typ sei bezeichnet durch $N_1^{t_s}, \dots, N_{m_s}^{t_s}$. Anhand dieser Proben versucht man, die Parameter des Modells mit Hilfe der Maximum-Likelihood-Methode zu schätzen. Dazu ersetzt man die Wahrscheinlichkeit einer Stichproben-Zelle des Typs $N_i^{t_s}$ durch die berechnete Häufigkeit $y_{N_i^{t_s}}(t_s)$. Der gesamte Datensatz sei bezeichnet mit D . Nimmt man an, dass die Zellen innerhalb und zwischen den Stichproben unabhängig sind, weil die Populationsgröße deutlich größer als die Stichprobe ist, so erhält man folgende Likelihood-Funktion:

$$L_D(r_i, \alpha_i, x_{n_i}^0 | i = 1, \dots, l) = \prod_{i=1}^k \prod_{j=1}^{m_i} y_{N_j^{t_i}}.$$

Nun sucht man den Parametersatz unter Berücksichtigung der Nebenbedingungen, die diese Funktion maximieren. Gewöhnlich wandelt man dieses Problem noch durch die log-Likelihood-Funktion

$$F_D(r_i, \alpha_i, x_{n_i}^0 | i = 1, \dots, l) := - \sum_{i=1}^k \sum_{j=1}^{m_i} \log(y_{N_j^{t_i}})$$

in ein Minimierungsproblem um. Daraus ergibt sich das nichtlineare Optimierungsproblem mit nichtlinearen Gleichungs- und linearen Ungleichungsrestriktionen:

Fallstudie 1.1

Gegeben: Datensatz D

$$\begin{aligned} & \min F_D(\vec{r}, \vec{\alpha}, \vec{x}^0) \\ & a_i \geq 0 \quad \forall i = 0 \dots l, \\ & x_{\vec{n}i}^0 \geq 0 \quad \forall i = 0 \dots l, \quad \sum_{i=0}^l x_{\vec{n}i}^0 = 1, \\ & \dot{x}_{\vec{n}} = \left(\sum_{i=1}^l n_i r_i - \sum_{i=1}^l (1 - n_i) \alpha_i \right) x_{\vec{n}} + \left(\sum_{\vec{n} \leq \vec{n}, h(\vec{n}, \vec{n})=1} x_{\vec{n}} \sum_{i=1}^l \alpha_i (n_i - \bar{n}_i) \right), \\ & x_{\vec{n}}(0) = x_{\vec{n}}^0 \quad \forall \vec{n} \in \{0, 1\}^l. \end{aligned}$$

1.2 Fallstudie 2: Die Proteinspaltung durch das Enzym Proteasom

Das Enzym Proteasom spaltet Proteine innerhalb einer Zelle in Fragmente, meistens in eine Länge von sieben bis neun Aminosäuren. Diese Fragmente gelangen durch einen MHC-class-I an die Zelloberfläche (siehe Abbildung 1.3). Wird ein Fragment durch cytotoxische T-Lymphozyten als nicht-körperigen erkannt, z.B. aufgrund einer Vireninfektion, wird die entsprechende Zelle getötet und die Verbreitung des Virus gestoppt (siehe [RAM93], [PAM98]). Man hegt die Hoffnung, auf Grundlage dieses Mechanismus spezifische und effektive Impfstoffe zu entwickeln. Folglich versucht man, den proteasomalen Spaltungsprozess zu verstehen.

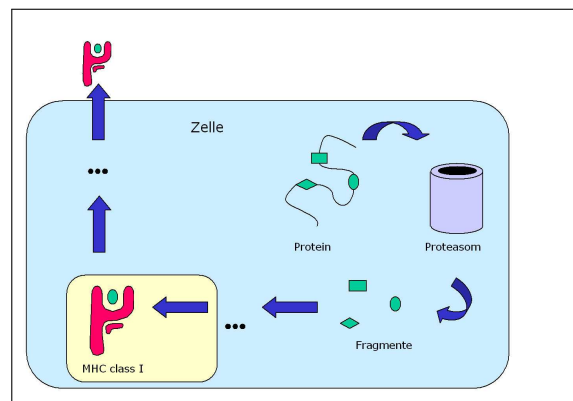


Abbildung 1.3: Ablauf des proteasomalen Proteinabbaus in einer Zelle.

Um einen tieferen Einblick in diesen Prozess zu gewinnen, werden Experimente durchgeführt, bei denen das Proteasom ausgewählte Modell-Proteine *in vitro* spaltet (siehe [NUS98]). Das Ziel ist hierbei, eine Methode zu entwickeln, mit der sich voraussagen lässt, welche Fragmente entstehen. Die resultierenden Fragmente werden identifiziert und statistisch analysiert. Die Fragmentdetektion und -identifikation ist sehr kompliziert. Daher werden viele Fragmente nicht entdeckt, obwohl die experimentelle Technik sehr hoch entwickelt ist. Auch wird im Allgemeinen unterschätzt, wie oft eine erkannte Fragmentart tatsächlich auftritt. Folglich ist es sehr schwer, die Ergebnisse der Experimente zu interpretieren. Diese Fallstudie konzentriert sich auf zwei Experimente, bei denen das Proteasom 20S das Substrat Hefe-Enolase oder Prion spaltet. Hierbei werden das entsprechende Protein und das Proteasom vermengt und erwärmt. Anschließend werden die Spaltungsprodukte entsprechend ihrer Länge durch das Verfahren „high-pressure-liquid-chromatography“ (HPLC) getrennt. Die Fragmente im Eluat werden nun mittels Massenspektrometrie und Edmansequenzierung analysiert. Normalerweise ist es möglich, die genaue Sequenz der Fragmente

festzustellen, welcher dann ihre jeweils gemessene Anzahl zugewiesen werden kann. Für die experimentellen Details wird auf die Literatur [KUT00] und [TOE01] verwiesen.

Die Daten sind also das Ergebnis der Kombination von zwei Prozessen: Der erste ist der Abbauprozess des Proteins durch das Proteasom und der zweite ist der experimentelle Detektierungsprozess. Es ist dabei nicht zu erkennen, welche Struktur innerhalb der Daten auf welchen Prozess zurückzuführen ist. Um diese beiden Teile zu unterscheiden, verwendet man einen modellbasierten Ansatz. Es werden zuerst der proteasomale Abbauprozess sowie das Detektierungsverfahren einzeln modelliert, anschließend wird das komplette Experiment simuliert.

Das im weiteren Verlauf vorgestellte Modell soll Einblicke in die innere Struktur eines proteasomalen Datensatzes ermöglichen. Es soll kein Algorithmus zur Vorhersage der Fragmente entwickelt werden, sondern eine Methode, um die Daten besser zu verstehen.

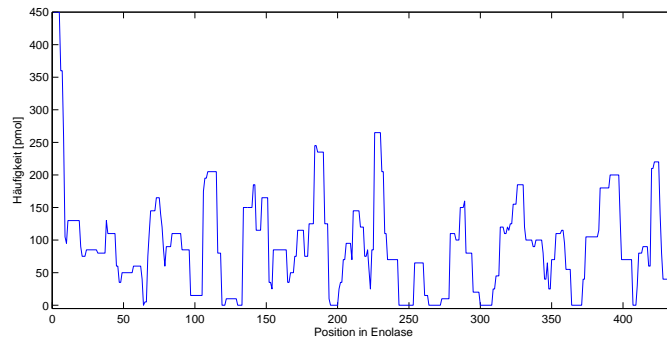


Abbildung 1.4: Detektierte Häufigkeiten der Aminosäurenposition in Enolase.

Um ein Modell für den proteasomalen Abbau und die Fragmentendetektion zu entwickeln, müssen die Daten zunächst statistisch analysiert werden. Im Folgenden wird sich rein auf den Enolase-Datensatz bezogen. Der verfügbare Datensatz (siehe Anhang **Proteasom**) beschreibt 109 verschiedene Fragmente mit ihren Anfängen, Enden und den gemessenen Häufigkeiten. Man nimmt an, dass die Wahrscheinlichkeit zur Detektion eines Fragmentes von seiner Länge und seiner Häufigkeit abhängt. Summiert man für jede Position auf dem Protein die gemessene Menge der Fragmente, die diese Position enthalten, erhält man Abbildung 1.4. Die Verteilung der Positionshäufigkeiten zeigt, dass für die meisten Positionen nur eine kleine Menge gefunden wurde.

Von den 435 möglichen Trennungsstellen in der Enolase (436 Aminosäuren lang) beginnen oder enden Fragmente nur an 114 verschiedenen Positionen. Hierbei handelt es sich nicht um die Stellen in einem einzelnen Molekül, sondern um alle jemals im Datensatz detektierten Stellen. Der Abstand zwischen zweien folgt einer gekürzten geometrischen Verteilung (χ^2 -Test mit $df = 6$ ergibt $\chi^2 = 3.29$ mit $p = 0.23$).

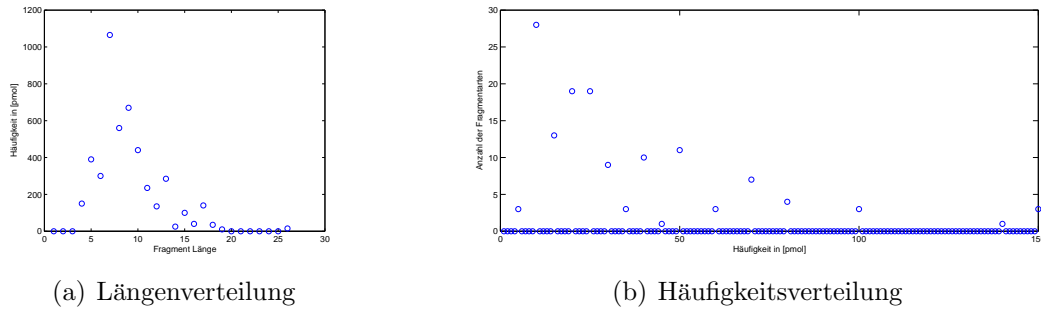


Abbildung 1.5: Längen- und Häufigkeitsabhängigkeit innerhalb des Enolase Datensatzes.

Falls die Trennungsstellen unabhängig sind, erwartet man eine derartige Verteilung. Ebenso würde man für die Fragmentenlängen eine geometrische Verteilung erwarten. Aber wie in Abbildung 1.5 zu erkennen ist, sind insbesondere kurze Fragmente sehr selten und Fragmente aus sieben bis neun Aminosäuren sehr häufig. Es liegt auf der Hand, dass die Ergebnisse des Detektierungsprozesses auch von der Häufigkeit der Fragmente abhängen. Die Verteilung der Fragmenthäufigkeiten folgt keinem Schema. Dennoch steigt sie bis 20pmol und fällt dann wieder. Daraus kann man schließen, dass seltene Fragmente schwieriger zu detektieren sind. Da beide Prozesse nacheinander ablaufen, wird für die Herleitung des Modells vorausgesetzt, dass sie voneinander unabhängig sind. Die Prozesse werden mit jeweils zwei Wahrscheinlichkeiten modelliert: Trennung oder keine Trennung und detektiert oder nicht detektiert. Diese Vorgabe erlaubt die Verwendung einer Binomialverteilung, siehe dazu auch [HIG02].

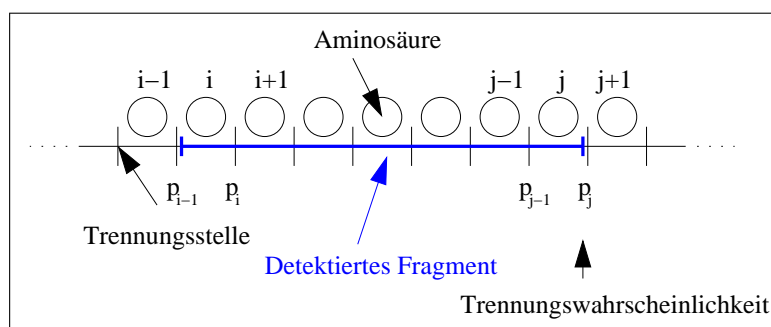


Abbildung 1.6: Proteinfragment, das aus Aminosäuren besteht, sowie die zugehörigen Wahrscheinlichkeiten.

Es sei nun die Trennungswahrscheinlichkeit in einem Protein nach der i -ten Aminosäure p_i mit $p_i \in I := \{p \mid 0 \leq p \leq 1\}$. $[i, j]$ beschreibt das Fragment zwischen den

Positionen i und j des Proteins (Abbildung 1.6). Dann ergibt sich die Entstehungswahrscheinlichkeit eines Fragmentes mit p_{ij} zu:

$$p_{ij} := p_{i-1} \prod_{k=i}^{j-1} (1 - p_k) p_j \quad i < j.$$

$q(l, h)$ sei die Detektierwahrscheinlichkeit, abhängig von der Länge l des Fragments und der Häufigkeit h . Nun bedeutet $[i, j] = \nu$, dass das Fragment $[i, j]$ mit der Häufigkeit ν detektiert wurde. Mit n bezeichnet man die Anzahl der vorhandenen Proteinmoleküle.

Um das Modell möglichst einfach zu halten, wird für die Detektierwahrscheinlichkeit $q(l, h)$ die Unabhängigkeit der längenabhängigen Wahrscheinlichkeit $q_l(l)$ und der häufigkeitsabhängigen Wahrscheinlichkeit $q_h(h)$ angenommen, so dass sich

$$q(l, h) = q_l(l)q_h(h)$$

schreiben lässt. Die Funktion q_l (ebenso q_h) sei stetig, gleich Null unterhalb einer Länge l_1 (einer Häufigkeit h_1) und 1 oberhalb von l_2 (h_2). q_l sei dabei stückweise linear und q_h sei das Quadrat einer stückweisen linearen Funktion (siehe Abbildung 1.7). Für die genauere Herleitung sei auf [MUE02] verwiesen.

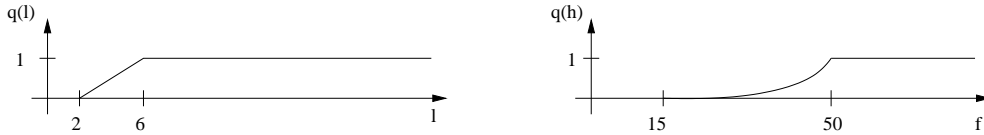


Abbildung 1.7: Die Funktionen $q_l(l)$ und $q_h(h)$.

Nun lässt sich aus der Summe aller Kombinationen, bei denen m Fragmente entstanden aber nur ν detektiert wurden, die Wahrscheinlichkeit P eines einzelnen Fragmentes $[i, j]$ mit der detektierten Häufigkeit ν berechnen. Hierbei ermöglicht der binomiale Ansatz das Aufspalten in eine Detektions- und eine Entstehungswahrscheinlichkeit.

$$P([i, j] = \nu) = \sum_{m=\nu}^n \underbrace{\binom{m}{\nu} q(l, h)^\nu (1 - q(l, h))^{m-\nu}}_{\text{Detektionswahrscheinlichkeit}} \underbrace{\binom{n}{m} p_{ij}^m (1 - p_{ij})^{n-m}}_{\text{Entstehungswahrscheinlichkeit}}.$$

Sei k die Anzahl aller möglichen Fragmente, so lässt sich mittels Kombinatorik zeigen:

$$k = \sum_{r=1}^s r = \frac{(1+r)r}{2}.$$

Nun weist man den Fragmenten $[i, j]$ die detektierten Häufigkeiten zu. Falls ein Fragment nicht detektiert wurde, erhält es die Häufigkeit Null. Daraus resultiert der Datensatz D . Mit s als Länge des untersuchten Proteins und $\vec{p} = (p_1, \dots, p_{s-1})^T$ als Parametersatz lässt sich die Likelihood-Funktion eines solchen Datensatzes bestimmen:

$$M_D(p) := \prod_{r=1}^k P([i_r, j_r] = \nu_r).$$

Nach der Maximum-Likelihood-Methode liegt die Problemstellung darin, den Parametersatz \hat{p} zu finden, der die Likelihood-Funktion maximiert. Es soll also für \hat{p} gelten

$$M_D(\hat{p}) = \max_p M_D(p).$$

Üblicherweise wird dieses Maximierungsproblem durch das Minimierungsproblem mit Hilfe der Log-Likelihood-Funktion $-\log(M_D(p))$ untersucht. Berücksichtigt man

$$-\log(M_D(p)) = \sum_{r=1}^k -\log(P([i_r, j_r] = \nu_r))$$

mit $0 \leq P([i_r, j_r] = \nu_r) \leq 1$, lässt sich das Problem in ein Problem der Summe der kleinsten Quadrate umwandeln. Hierfür definiert man die Funktion $F_D(p)$ mit

$$F_D(p) := \begin{pmatrix} -\log(P([i_1, j_1] = \nu_1)) \\ \dots \\ -\log(P([i_k, j_k] = \nu_k)) \end{pmatrix}.$$

Daraus ergibt sich das nichtlineare Optimierungsproblem mit linearen Ungleichungsrestriktionen:

Fallstudie 1.2

Gegeben: Datensatz D

$$\min \|F_D(\vec{p})\|_2$$

$$0 \leq p_i \leq 1.$$

1.3 Fallstudie 3: Immunisierungsstrategie gegen Hepatitis in Krankenhäusern

In Krankenhäusern versucht man das Personal vor Krankheiten wie z.B. Hepatitis zu schützen, indem man es einer Grundimmunisierung unterzieht. Ein Individuum gilt als immun, falls der Antikörpertiter über einem bestimmten Wert liegt. Um dieses zu überprüfen bzw. zu gewährleisten, stehen zwei Maßnahmen zur Verfügung: Man kann die Anzahl der Antikörper mittels Blutprobentests bestimmen oder eine Impfung durchführen. Bei beiden Maßnahmen fallen Kosten an, wobei eine Kontrolluntersuchung billiger als eine Impfung ausfällt. Ist jedoch der Antikörpergehalt zu niedrig, muss anschließend sofort geimpft werden. Das Ziel ist also, eine Strategie zu entwickeln, welche die Immunität des Personals bei geringstem Kostenaufwand sicher stellt. Im Folgenden wird für ein Mitglied des Personals der Ausdruck „Patient“ gewählt.

Um ein geeignetes Modell zu entwickeln, sind zunächst die erhobenen Daten zu sichten (siehe Anhang **Immunisierungsstrategie**). Dabei werden erste Zusammenhänge erkannt und zugrunde liegende Mechanismen und Probleme erfasst.

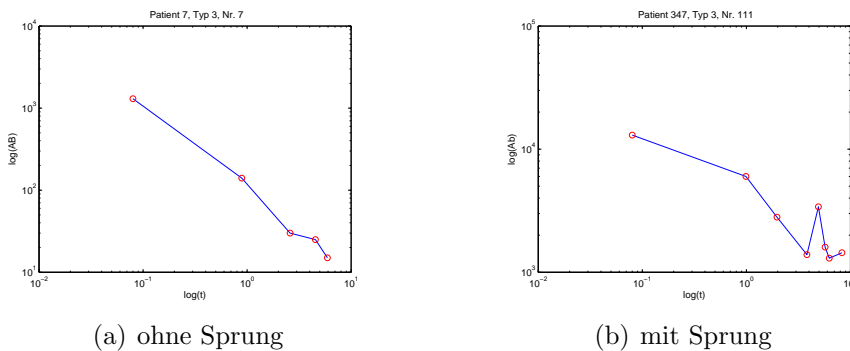


Abbildung 1.8: Log-Log Plot der Anzahl der Antikörper eines Patienten.

In Abbildung 1.8 ist eine Beziehung zwischen der Zeit und der Anzahl der Antikörper dargestellt, die sich gut modellieren lässt, weil der Logarithmus der Anzahl der Antikörper Ab linear im Logarithmus der Zeit t fällt. Es treten jedoch immer wieder unvorhersehbare sprunghafte Veränderungen auf, was sich für die Parameteridentifikation als problematisch erweist. Ein weiteres Problem ist, dass Ab nach einer Impfung nicht bekannt und mit einem Test nicht exakt bestimmt werden kann. Auch das muss im Modell berücksichtigt werden. Insgesamt kann die Antikörperzahl nur diskret bestimmt werden, wobei ihr kontinuierlicher Verlauf unbekannt bleibt. Dennoch lässt sich mit wahrscheinlichkeitstheoretischen Mitteln ein erstes vereinfachtes Modell aufstellen.

Mit $z(t)$ sei die Anzahl der Antikörper im Blut eines Patienten bezeichnet, gegeben durch eine zeitabhängige Zufallsvariable Z_t . Über diese Zufallsvariable weiß man, dass sie ohne äußere Einflüsse (Steuerung oder Störung) im Durchschnitt wegen des natürlichen Abbaus der Antikörper mit der Zeit fällt. D.h. für ihren Erwartungswert gilt nach einer Zeit τ $E[Z_{t+\tau}] = E[Z_t] - a\tau$, wobei $a > 0$ und konstant ist. Weiterhin wird in diesem Fall das Wissen um die Anzahl der Antikörper immer ungenauer, was sich in der Veränderung der Varianz ausdrückt: $Var[Z_{t+\tau}] = Var[Z_t] + b\tau$, wobei $b > 0$.

Das einfachste System, das diese Zusammenhänge beschreibt, ist

$$Z_t \sim N(\mu, \sigma^2) \Rightarrow Z_{t+\tau} \sim N(\mu - a\tau, \sigma^2 + b\tau)$$

oder

$$dZ_t = -a dt + \sqrt{b} dW_t.$$

Hierbei steht W_t für das „Gaußsche Rauschen“.

Die Aufgabenstellung lässt sich nun in folgendes Ziel übersetzen: Das System soll in dem Zeitraum 0 bis T über einem bestimmten Grenzwert bleiben. Da man den exakten Zustand des Patienten nicht kennt, möchte man $P(Z_t < 0) \leq \gamma$ garantieren, wobei $t \in [0, T]$. γ sollte klein sein, um den Impfschutz mit hoher Wahrscheinlichkeit gewähren zu können, weil es mit positiver Wahrscheinlichkeit kurze Zeitintervalle gibt, bei denen $z(t)$ negativ wird.

Für

$$Z_t \sim N(\mu, \sigma^2) \text{ mit } (\mu, \sigma^2) \in \mathbb{R} \times \mathbb{R}_+$$

existiert eine Konstante $k_\gamma > 0$, so dass $P(Z < 0) \leq \gamma$ gilt. Dadurch erhält man in der (μ, σ^2) -Ebene eine erlaubte Region Ω (siehe Abbildung 1.9).

Sei $\Omega := \{(\mu, \sigma^2) | \mu^2 > k_\gamma^2 \sigma^2\}$ mit $\partial\Omega := \Gamma^+ \cup \Gamma^-$.

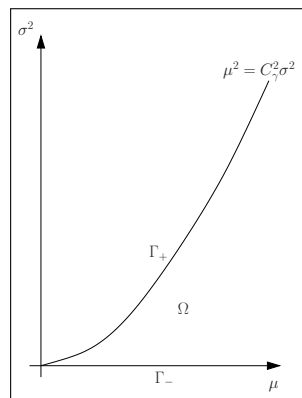


Abbildung 1.9: Erlaubte Region Ω innerhalb der (μ, σ^2) -Ebene.

In dieser Ebene bewegt sich die Zufallsvariable mit der Zeit, beginnend an einem Punkt (μ_I, σ_I^2) , der der Verteilung nach einer Impfung entspricht. Also besitzt unser Modell vier Parameter: μ_I, σ_I, a, b . Diese müssen aus den Daten gewonnen werden.

Der Datensatz D liefert zwei verschiedene Informationsarten. Man erhält den Zeitpunkt der Impfungen und den Zeitpunkt einer Kontrolle mit entsprechend gemessenem Antikörperwert.

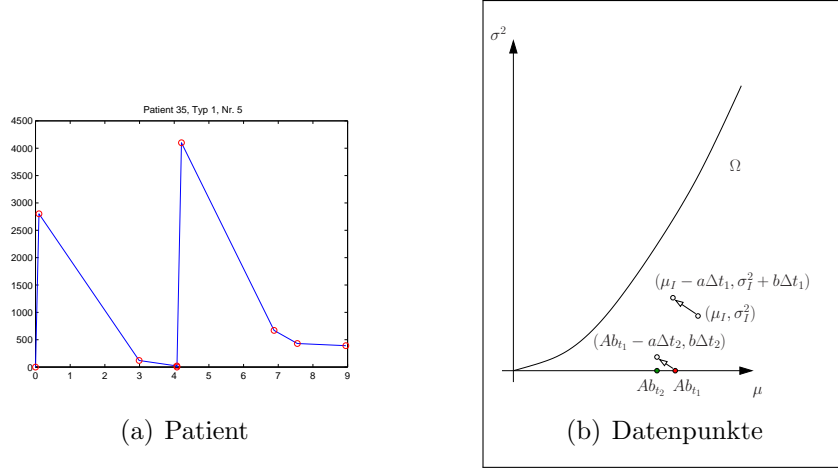


Abbildung 1.10: Ab-Wert eines Patienten mit zwei Impfungen und der Verlauf zweier Datenpunkte in der (μ, σ^2) -Ebene.

Plottet man den Datensatz eines einzelnen Patienten, wobei der Wert für Ab zum Zeitpunkt einer Impfung (zur deutlichen Abgrenzung) gleich Null gesetzt wird, so erhält man Abbildung 1.10 (a). Die Daten werden so umstrukturiert, dass sie durch das Modell leichter zu handhaben sind. Man kann diese Informationen in zwei auswertbare Datentypen d_1 und d_2 umwandeln, und zwar zu:

d_1 : Zeit Δt_i seit letzter Impfung und der gemessene Antikörperwert $Ab_{\Delta t_i}$

d_2 : Zeit Δt_j seit letzter Kontrolle und der gemessene Antikörperwert $Ab_{\Delta t_j}$.

Für diese beiden Datentypen erhält man folgenden Zusammenhang (siehe auch in der (μ, σ^2) Ebene Abbildung 1.10 (b)):

$$Ab_{\Delta t_i} \sim N(\mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i)$$

$$Ab_{\Delta t_j} \sim N(Ab_{\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j).$$

Im ersten Schritt wird jeder Patient einzeln betrachtet und ein einfaches Modell mit seinem jeweiligen Datensatz $D = d_1 \cup d_2$ erstellt. Da man eine normalverteilte

Zufallsvariable erwartet, lautet die Dichtefunktion

$$\pi(x; \mu, \sigma) = \frac{1}{2\pi\sqrt{\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma}}.$$

An dieser Stelle lässt sich im Gegensatz zu den beiden anderen Fallstudien die Likelihood-Funktion noch nicht aufstellen. Da es sich hier um keine diskrete, sondern um eine kontinuierliche Zufallsvariable handelt, ist anstelle der Wahrscheinlichkeit die Dichtefunktion zu verwenden. (Zur Erläuterung sei auf das kommende Kapitel verwiesen.)

Entspricht n_i der Anzahl der Datenpunkte des Datentyps d_i , ist die Likelihood-Funktion eines Datensatzes definiert durch:

$$\begin{aligned} L_D(\mu_I, \sigma_I, a, b) &= \prod_{i=1}^{n_1} \pi(Ab_{1,\Delta t_i}; \mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i) \\ &\quad \cdot \prod_{j=1}^{n_2} \pi(Ab_{2,\Delta t_j}; Ab_{2,\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j). \end{aligned}$$

Den Maximum-Likelihood-Ansatz

$$\max_{\mu_I, \sigma_I, a, b} L_D(\mu_I, \sigma_I, a, b)$$

wandelt man mit Hilfe der Log-Likelihood-Funktion

$$\begin{aligned} F_D(\mu_I, \sigma_I, a, b) &:= - \sum_{i=1}^{n_1} \log(\pi(Ab_{1,\Delta t_i}; \mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i)) \\ &\quad - \sum_{j=1}^{n_2} \log(\pi(Ab_{2,\Delta t_j}; Ab_{2,\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j)) \end{aligned}$$

wie schon gezeigt in ein Minimierungsproblem um. So ergibt sich das nichtlineare Optimierungsproblem mit linearen Ungleichungsrestriktionen:

Fallstudie 1.3

Gegeben: Datensatz D

$$\min F_D(\mu_I, \sigma_I, a, b)$$

$$\mu_I \in \mathbb{R}, \sigma_I \geq 0$$

$$a \geq 0, b \geq 0.$$

1.4 Ausblick

Die weitere Vorgehensweise bei den einzelnen Fallstudien ist, entsprechend angepasst, jedes Mal dieselbe und verläuft in vier Schritten.

Im ersten Schritt erfolgt die mathematische Formulierung. Das resultierende Problem wird anschließend klassifiziert und analysiert. D.h., es wird nach möglichen Vereinfachungen der mathematischen Struktur gesucht, ohne den immunologischen Hintergrund zu berücksichtigen.

Im zweiten Schritt erfolgt die Datenanalyse. Man untersucht hierbei das vorliegende Datenmaterial mit einem besonderen Augenmerk auf Redundanzen und Fehler, die entweder experimentell oder menschlich bedingt auftreten. Diese versucht man zu entfernen.

Im dritten Schritt werden vorhandene Optimierungsverfahren auf die jeweilige Problemstellung angepasst. D.h., man optimiert diese Verfahren bezüglich der Geschwindigkeit mit Hilfe von formulierungsbedingten Informationen aus dem ersten Schritt oder mit problembezogenen Informationen aus dem zweiten Schritt.

Im vierten Schritt werden die Ergebnisse untersucht. Zum einen wird geprüft, ob modellbedingte Redundanzen existieren. Man fragt sich, inwieweit im Modell unter den zu identifizierenden Parametern Abhängigkeiten erzwungen wurden oder sich generell solche Zusammenhänge aus dem natürlichen Prozess heraus ergeben. Anschließend stellt sich die Frage, ob man mit den so gewonnenen Informationen das Modell weiter anpassen oder die Komplexität reduzieren kann.

Zum anderen versucht man dann das eigentlich Ziel zu erreichen: Mit den gewonnenen Informationen sollen tiefere Einblicke in die Struktur des naturwissenschaftlichen Prozesses ermöglicht und (bzw. oder) zuverlässige Vorhersagen getroffen werden.

Kapitel 2

Mathematische Grundlagen

Im folgenden Kapitel wird ein genereller Überblick über die einzelnen in dieser Arbeit verwendeten mathematischen Hilfsmittel gegeben.

2.1 Zur Statistischen Datenanalyse

Die vorgestellten Grundlagen sind im Wesentlichen an [FAH97] angelehnt, verwendet wurden aber auch Erkenntnisse aus [SAL96] und [CLO00].

2.1.1 Allgemeine Statistik

Die Häufigkeit eines Ereignisses A in einem Experiment, bestehend aus n Versuchen, ist $n(A)/n$, wobei $n(A)$ die Anzahl der Versuche ist, bei denen Ereignis A eintritt. Ein Grundraum Ω ist ein nicht leerer Satz von sich gegenseitig ausschließenden elementaren Ereignissen e .

Vorausgesetzt, Ω ist eine diskrete Menge, d.h. Ω ist entweder finit oder abzählbar unendlich, dann ist eine Wahrscheinlichkeitsfunktion p auf Ω gegeben, wenn gilt: p bildet Ω auf das Einheitsintervall $[0, 1]$ ab und $\sum_{e \in \Omega} p(e) = 1$. Für eine Teilmenge $A \subset \Omega$ berechnet sich dann die Wahrscheinlichkeit zu $Pr[A] = \sum_{e \in A} p(e)$. Daraus ergeben sich die folgenden Wahrscheinlichkeiten (Axiome von Kolmogoroff):

1. $Pr[0] = 0, Pr[\Omega] = 1$.
2. $Pr[\Omega - A] = 1 - Pr[A]$.
3. $Pr[A \cup B] = Pr[A] + Pr[B] - Pr[A \cap B], \forall A, B \subset \Omega$.

Die Wahrscheinlichkeit eines Ereignisses $e \in \Omega$ ist dann $Pr[\{e\}] = p(e)$.

Im allgemeinen Fall, wobei Ω entweder diskret oder kontinuierlich ist, ist eine Wahrscheinlichkeitsfunktion Pr eine Abbildung der σ -Algebra aus Ω in das Einheitsintervall $[0, 1]$, welche obige drei Bedingungen erfüllt.

Für zwei Teilmengen $A, B \subset \Omega$ wird an Stelle von $Pr[A \cap B]$ oft $Pr[A, B]$ geschrieben.

Definition 2.1

(Bedingte Wahrscheinlichkeit) $Pr[A|B] = \frac{Pr[A, B]}{Pr[B]}$. Ereignisse A, B sind unabhängig genau dann, wenn $Pr[A, B] = Pr[A]Pr[B]$, also wenn $Pr[A] = Pr[A|B]$.

Vorausgesetzt, B_1, \dots, B_n sind messbare, sich gegenseitig ausschließende und nicht leere Mengen; d.h. $B_i \cap B_j = \emptyset, \forall i, j$ und weiter gilt

$$\Omega = B_1 \cup \dots \cup B_n, \quad Pr[\Omega \setminus \bigcup_{i=1}^n B_i] = 0.$$

Dann ergibt sich

$$Pr[A] = \sum_k Pr[A|B_k] Pr[B_k].$$

Mit $Pr[A|B] = \frac{Pr[A, B]}{Pr[B]}$ und $Pr[B|A] = \frac{Pr[B, A]}{Pr[A]}$ ergibt sich

$$Pr[A|B]Pr[B] = Pr[B|A]Pr[A]$$

und somit (Satz von Bayes):

$$Pr[B|A] = \frac{Pr[A|B] Pr[B]}{Pr[A]}.$$

Definition 2.2

Eine Zufallsvariable X ist eine Funktion $X : \Omega \rightarrow \mathbb{R}$.

Üblicherweise geht man von einer diskreten Zufallsvariable aus; im kontinuierlichen Fall muss es sich bei der Zufallsvariable um eine messbare oft reellwertige Funktion handeln. Im diskreten Fall schreibt man für $Pr[\{e \in \Omega | X(e) = x\}]$ oder $\sum_{e \in \Omega | X(e)=x} p(e)$ auch $Pr[X = x]$. Dann gilt $\sum_{x=-\infty}^{\infty} Pr[X = x] = 1$. Im kontinuierlichen Fall gilt für ein gegebenes $x \in \mathbb{R}$ üblicherweise $Pr[X = x] = 0$ und somit wird eine nicht negative Wahrscheinlichkeitsdichtefunktion $p_X : \mathbb{R} \rightarrow \mathbb{R}$ für eine kontinuierliche Zufallsvariable X definiert, die folgendes erfüllt:

$$\begin{aligned} Pr[a \leq X \leq b] &= Pr[e \in \Omega | a \leq X(e) \leq b] \\ &= \int_a^b p_X(x) dx. \end{aligned}$$

Man beachte, dass $\int_{-\infty}^{\infty} p_X(x)dx = 1$. Die Verteilungsfunktion ist für eine kontinuierliche Variable X definiert als

$$\Phi_X(x) = \int_{-\infty}^x p_X(y)dy,$$

so dass

$$Pr[a \leq X \leq b] = \Phi_X(b) - \Phi_X(a) = \int_a^b p_X(x)dx.$$

Definition 2.3

Der Erwartungswert einer diskreten Zufallsvariable X , geschrieben mit $E[X]$, ist im Falle der Konvergenz definiert durch

$$E[X] = \sum_{i=-\infty}^{\infty} iP_r[X = i]$$

und im kontinuierlichem Fall durch

$$E[X] = \int_{x=-\infty}^{\infty} xp_X(x)dx.$$

Anhand Definition 2.3 ist leicht zu erkennen, dass $E[X]$ linear im Sinne von

$$E[\alpha X + \beta Y] = \alpha E[X] + \beta E[Y]$$

für die Konstanten α, β ist. Mehr sogar für unabhängige Zufallsvariablen X_1 und X_2 gilt:

$$E[X_1 X_2] = E[X_1]E[X_2].$$

Definition 2.4

Das r -te Moment von X ist im Falle der Konvergenz definiert durch

$$E[X^r] = \sum_{-\infty}^{\infty} x^r Pr[X = x]$$

und im kontinuierlichem Fall durch

$$E[X^r] = \int_{-\infty}^{\infty} x^r p_X(x)dx.$$

Definition 2.5

Die Varianz $V[X]$ ist im Falle der Konvergenz definiert als das zweite Moment von $X - \mu$, wobei $\mu = E[X]$ ist, also

$$\begin{aligned} V[X] &= E[X - \mu]^2 \\ &= E[X^2] - 2\mu E[X] + \mu^2 \\ &= E[X^2] - 2\mu^2 + \mu^2 \\ &= E[X^2] - \mu^2. \end{aligned}$$

Für die Standardabweichung σ gilt $\sigma = \sqrt{V[X]}$. Mit obiger Definition gilt im diskreten Fall $V[\gamma X] = \gamma^2 V[X]$ und weiter im Falle, dass X und Y unabhängige Zufallsvariablen sind, dass die Varianz additiv ist:

$$V[X + Y] = V[X] + V[Y].$$

Zusätzlich zu den Kenngrößen für eine einzelne Zufallsvariable X gibt es auch Kenngrößen, die den Zusammenhang zwischen zwei Zufallsvariablen X und Y misst.

Definition 2.6

Die Kovarianz ist definiert als

$$Cov[X, Y] = E[X - E[X]] - E[Y - E[Y]].$$

Hat die Kovarianz den Wert Null, so sagt man: X und Y sind unkorreliert. Dies ist zum Beispiel bei stochastisch unabhängigen Zufallsvariablen der Fall. Die Umkehrung ist aber im Allgemeinen falsch. Bei normal verteilten Zufallsvariablen jedoch folgt aus der Unkorreliertheit auch die stochastische Unabhängigkeit.

Für Konstanten $\alpha, \beta, \gamma, \delta$ gilt

$$Cov[\alpha X + \beta, \gamma Y + \delta] = \alpha \gamma Cov[X, Y].$$

Definition 2.7

Die Korrelation ist definiert als

$$\rho = \rho[X, Y] = \frac{Cov[X, Y]}{\sqrt{V[X]}\sqrt{V[Y]}} = Corr[X, Y].$$

Es handelt sich hierbei um eine normierte Größe, deren Wert immer zwischen -1 und 1 liegt.

2.1.2 Spezielle Verteilungen

Im Folgenden werden kurz diejenigen wichtigen Verteilungen skizziert, die auch in dieser Arbeit verwendet werden.

Die Gleichverteilung

Die Gleichverteilung erfüllt im diskreten Fall

$$Pr[X = \omega] = \frac{1}{|\Omega|}.$$

Im kontinuierlichen Fall ist X gleich verteilt auf dem reellen Intervall $[a, b]$, wenn

$$\Pr[c \leq X \leq d] = \int_c^d \frac{dx}{b-a} = \frac{d-c}{b-a} \quad \forall a < c < d < b,$$

weswegen die Dichtefunktion gegeben ist durch:

$$p_X(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{sonst.} \end{cases}$$

Damit berechnet sich der Erwartungswert im kontinuierlichem Fall zu

$$\int_a^b \frac{x}{b-a} dx = \frac{b^2 - a^2}{2(b-a)} = \frac{a+b}{2}$$

und die Varianz beträgt

$$\begin{aligned} \int_a^b \frac{x^2}{b-a} dx - \left(\frac{a+b}{2}\right)^2 &= \frac{b^3 - a^3}{3(b-a)} - \left(\frac{a+b}{2}\right)^2 \\ &= \frac{b^2 + ab + a^2}{3} - \frac{a^2 + 2ab + b^2}{4} \\ &= \frac{b^2 - 2ab + a^2}{12}. \end{aligned}$$

Die Binomialverteilung

Ein Bernoulli-Experiment ist ein Versuch mit der Wahrscheinlichkeit p für einen Erfolg; d.h. eine Bernoulli-Zufallsvariable $Y : \Omega \rightarrow \mathbb{R}$ nimmt nur die Werte 0 oder 1 (1 für Erfolg und 0 für Misserfolg) an. Da die Versuche unabhängig ablaufen und mit X die Zufallsvariable, die die Erfolge in n Versuchen zählt, definiert ist, gilt

$$b(n; k) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}.$$

Man sagt, die Zufallsvariable X ist binomial verteilt. Man beachte $X = Y_1 + \dots + Y_n$, wobei jedes Y_i eine unabhängige Bernoulli-Zufallsvariable ist. Da $E[Y_i] = p$ und $V[Y_i] = E[Y_i^2] - E[Y_i]^2 = p - p^2 = p(1-p)$ gilt, folgt durch die Additivität des Erwartungswertes und der Varianz bei Unabhängigkeit, dass

$$E[X] = np$$

und

$$V[X] = np(1-p)$$

ist. Eine Multinomialverteilung verallgemeinert die Binomialverteilung im Falle von m verschiedenen Ereignissen eines Experimentes. Wenn p_i die Wahrscheinlichkeit des i -ten Ereignisses in einem Versuch ist, dann berechnet sich die Wahrscheinlichkeit $P[X = (n_1, \dots, n_m)]$, dass in n aufeinanderfolgenden Versuchen n_i -mal das i -te Ereignis eintritt zu

$$M(n; n_1, \dots, n_m) = P[X = (n; n_1, \dots, n_m)] = \frac{n!}{n_1! \cdots n_m!} p_1^{n_1} \cdots p_m^{n_m}.$$

Die Normalverteilung

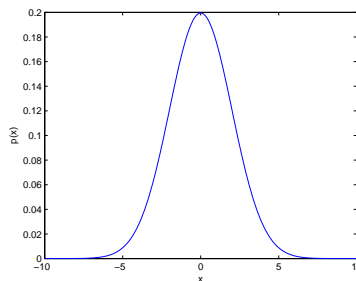


Abbildung 2.1: Die Dichte der Normalverteilung für $\mu = 0$ und $\sigma^2 = 2$.

Als Normal- oder Gauß-Verteilung bezeichnet man die Zufallsvariable X , welche auf der Dichtefunktion

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

beruht. Beim Graphen der Dichte p_X handelt es sich also um die Glockenkurve. Das Integral

$$\Phi_X(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{u^2}{2}} du,$$

welches die Verteilung definiert, kann in geschlossener Form nicht berechnet und muss daher numerisch approximiert werden. Durch die Symmetrie der Glockenkurve folgt für den Erwartungswert

$$E[X] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x e^{-\frac{x^2}{2}} dx = 0$$

und durch partielle Integration erhält man für die Varianz

$$\begin{aligned}
 V[X] = E[X^2] - E[X]^2 &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x^2 e^{-x^2/2} dx - 0^2 \\
 &= \int_{-\infty}^{\infty} (-x) \frac{-x e^{-x^2/2}}{\sqrt{2\pi}} dx \\
 &= \frac{-x e^{-x^2/2}}{\sqrt{2\pi}} \Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx \\
 &= 0 + 1 = 1.
 \end{aligned}$$

Allgemeiner gefasst ist eine Zufallsvariable X mit der Dichte

$$\pi(x; \mu, \sigma^2) := p_X(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

normal verteilt mit Erwartungswert μ und Varianz σ^2 (siehe Abbildung 2.1). In diesem Fall schreibt man, dass X die Verteilung $N(\mu, \sigma)$ hat.

Bei der Durchführung statistischer Tests ist die Verteilung von Größen wie

$$\sum_{i=1}^n (x_i - \bar{x})^2, \quad \frac{\sqrt{n}(\bar{x} - \mu)}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}}, \quad \text{mit } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

für normal verteilte x_i interessant. Solche Verteilungen werden auch Prüfverteilungen genannt.

Die χ^2 -Verteilung

Sind X_1, \dots, X_n unabhängige $N(0, 1)$ verteilte Zufallsvariablen, so heißt die Verteilung von $\sum_{i=1}^n X_i^2$ eine Zentrale χ^2 -Verteilung mit n Freiheitsgraden. Diese Verteilung besitzt die Dichte

$$p_{\chi_n^2}(x) = \frac{1}{2^{\frac{n}{2}} \Gamma(\frac{n}{2})} x^{n/2-1} e^{-x/2}, \quad x > 0.$$

Die Funktion Γ bezeichnet die Gamma-Funktion

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt,$$

für die $\Gamma(x+1) = x\Gamma(x)$ gilt. Für ganzzahlige $x = n$ ergibt sich gerade $\Gamma(n) = (n-1)!$.

Der Erwartungswert ergibt

$$E[\chi_n^2] = n$$

und die Varianz berechnet sich zu

$$V[\chi_n^2] = 2n.$$

Liegen nach einem Versuch n unabhängige Beobachtungen x_1, \dots, x_n des selben Experimentes vor, so möchte man überprüfen, ob diese Beobachtungen aus einer Grundgesamtheit stammen, welche bezüglich des betrachteten Merkmales einer vorgegebenen Verteilung z.B. der Normalverteilung $N(\mu_0, \sigma_0^2)$ folgt.

Bei bekanntem μ_0 lässt sich die Varianz σ_0^2 durch

$$\sigma_0^2 \approx s_{\mu_0}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_0)^2.$$

schätzen.

Für die Verteilung dieser Größe gilt

$$n \frac{s_{\mu_0}^2}{\sigma_0^2} \chi_n^2.$$

Ist dagegen μ_0 unbekannt, so benutzt man die empirische Varianz

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

als Schätzwert für σ^2 . Für deren Verteilung gilt dann

$$(n-1) \frac{s^2}{\sigma_0^2} \chi_{n-1}^2.$$

Hypothesentest

Mit den n unabhängigen Beobachtungen x_1, \dots, x_n möchte man sich zwischen zwei sich gegenseitig ausschließenden Hypothesen H_0 und H_1 (Null- und Alternativhypothese) entscheiden. Man kann zwei Fehler begehen. Man entscheidet sich für H_1 , obwohl H_0 vorliegt [Fehler 1. Art (α -Fehler)], oder aber man entscheidet sich für H_0 , obwohl H_1 vorliegt [Fehler 2. Art (β -Fehler)]. Hierbei soll die Wahrscheinlichkeit, die falsche Wahl zu treffen, unter einem vorgegebenen Wert, dem sogenannten Signifikanzniveau α , liegen. Von einem Test zum Niveau α spricht man dann, wenn das Testverfahren nur mit einer Wahrscheinlichkeit von höchstens α den Fehler 1. Art macht, ohne die B Wahrscheinlichkeit β des Fehlers 2. Art zu berücksichtigen.

Mit Hilfe des χ^2 -Anpassungstests auf Normalverteilung lassen sich die zwei Hypothesen zum Signifikanzniveau α testen:

H1: die Grundgesamtheit ist $N(\mu_0, \sigma_0^2)$ verteilt.

H2: die Grundgesamtheit ist nicht $N(\mu_0, \sigma_0^2)$ verteilt.

Hierbei müssen für μ_0 und σ_0^2 spezifische Werte verwendet werden.

Beim Test durchläuft man folgende fünf Schritte:

1. Unterteile das reelle Intervall $(-\infty, \infty)$ in k Klassen.
2. Berechne $O_i =$ Anzahl der beobachteten x_j in der i -ten Klasse.
3. Berechne die Wahrscheinlichkeit $p_i = Pr(X \in I_i), i = 1, \dots, k$, dass die Beobachtung unter Hypothese H_0 in der i -ten Klasse liegt.
4. Berechne $E_i = np_i$ für die erwartete Anzahl von Beobachtungen in der i -ten Klasse nach n Versuchen.
5. Berechne

$$T = \sum_{i=1}^k \frac{1}{E_i} (O_i - E_i)^2.$$

T ist asymptotisch χ^2 -verteilt mit $k - 1$ Freiheitsgraden. Diese Approximation ist hinreichend genau, wenn nicht mehr als 20% der E_i kleiner als 5 sind und kein Wert E_i kleiner als 1 ist.

Die Nullhypothese H_0 ist mit dem Signifikanzniveau α zu verwerfen, falls gilt:

$$T > \chi_{k-1; 1-\alpha}^2 = \Phi_{\chi^2}(\alpha).$$

Möchte man diesen Test ohne bestimmte, festgelegte Werte μ_0 und σ_0^2 verwenden, so kann man unter anderem $\mu_0 = \bar{x}$ und $\sigma_0^2 = s^2$ setzen. Dabei ist das Vorgehen nur noch näherungsweise richtig und wird erst für $n \rightarrow \infty$ exakt. Man sollte an dieser Stelle beachten, dass der χ^2 -Test bei beliebigen Verteilungen funktioniert. Wenn man speziell auf Normalverteilung testen möchte, so gibt es spezialisiertere Tests.

2.1.3 Multivariate Statistik

Bei multivariaten Verfahren (MVA) betrachtet man nicht eine Variable isoliert (univariat verteilt), sondern das Zusammenwirken mehrerer Variablen zugleich, also ihre Abhängigkeitsstruktur. Häufig verwendete MVA sind z.B. die Regressionsanalyse oder die Faktoranalyse. Die klassischen Verfahren sind lineare Modelle, die besondere Anforderungen an die verwendeten Daten stellen. So sollten die Daten ausreißerfrei und nicht asymmetrisch verteilt sein. Bei einer Abweichung der Daten

von der geforderten Struktur behilft man sich beispielsweise, indem man vorhandene Ausreißer entfernt oder die Daten einer nichtlinearen Transformation, etwa Logarithmieren unterzieht. Es existieren vielfach alternative Methoden, die iterativ gewonnene Lösungen ermöglichen (z.B. verallgemeinerte lineare Modelle).

Als Gütekriterien werden oft die Euklidische Distanz in einem mehrdimensionalen Raum oder aber die Varianz verwendet und diese dann minimiert beziehungsweise maximiert.

Faktoranalyse

Als Faktoranalyse bezeichnet man Verfahren, mit denen mehrere Variablen zu einigen wenigen Faktoren zusammengefasst werden können.

Voraussetzungen für eine Faktoranalyse sind hierbei:

1. Alle Variablen müssen mindestens intervallskaliert sein;
2. Die Repräsentativität der Stichprobe;
3. Eine inhaltlich gute Vorstellung über die Bedeutung der Variablen;
4. Die entstehenden Faktoren müssen sich hinsichtlich ihrer Eigenwerte und Ladungen hinreichend gut abgrenzen lassen.

Die Faktoranalyse durchläuft unabhängig vom Rechenverfahren die folgenden Schritte:

1. Rohdatenaufbereitung
2. Wahl des Extraktionsverfahrens
3. Extraktion
4. (kriteriengeleitete) Wahl der Faktoren
5. Hypothesenanpassung
6. ggf. Rotation
7. Auswertung.

Zur formalen Vorgehensweise benötigt es erst einige Definitionen.

Definition 2.8

- a) *Unter Variablen versteht man die zugrunde liegenden, gesuchten Parameter der Gewählten Modell/Hypothesen-Funktion.*

- b) Unter Faktoren versteht man die Variablen, die durch eine Faktorisierung entstanden und somit ohne Hypothese inhaltlich nutzlos sind.
- c) Die Rechenmethode, mit der die in den Variablen enthaltene Varianz extrahiert wird und welche die Faktoren erzeugt, nennt man Extraktion.
- d) Mit Faktorladung wird die Korrelation einer Variablen mit einem Faktor bezeichnet.
- e) Der Wert der Gesamtvarianz aller Variablen, den ein Faktor aufklärt, heißt Eigenwert dieses Faktors.

Hauptkomponentenanalyse

Eine häufig angewendete Extraktions-Verfahren ist die Hauptkomponentenanalyse. Hierbei wird eine Hauptachsentransformation durchgeführt. Man minimiert die Korrelation mehrdimensionaler Merkmale durch Überführung in einen Vektorraum mit neuer Basis. Die Hauptachsentransformation lässt sich durch eine orthonormale Matrix angeben, die aus den Eigenvektoren der Kovarianzmatrix gebildet wird. D.h., für jeden Datensatz wird eine eigene Transformationsmatrix berechnet.

Als Schätzer für die Korrelation ρ verwendet man die Stichprobenkorrelation

$$r_{XY} = \frac{s_{XY}}{s_X s_Y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

$$= \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sqrt{\left(\sum_{i=1}^n x_i^2 - n \bar{x}^2\right) \left(\sum_{i=1}^n y_i^2 - n \bar{y}^2\right)}}$$

Dabei sind x_1, \dots, x_n die Realisationen von X und y_1, \dots, y_n die Realisationen von Y , welche paarweise (x_i, y_i) erhoben worden sind.

Sind nun X_1, \dots, X_m Zufallsvariablen, so fasst man die geschätzten Korrelationen zwischen je zwei Merkmalen in der Korrelationsmatrix R zusammen:

$$R = \begin{pmatrix} 1 & r_{X_1 X_2} & \cdots & r_{X_1 X_m} \\ r_{X_2 X_1} & 1 & & \vdots \\ \vdots & & & r_{X_{m-1} X_m} \\ r_{X_m X_1} & \cdots & r_{X_m X_{m-1}} & 1 \end{pmatrix}.$$

Diese Matrix ist symmetrisch ($R = R^T$) und besitzt daher nur reelle, nicht negative Eigenwerte $\lambda_1, \dots, \lambda_m$ und ist diagonalisierbar. Somit existiert eine orthonormale

Matrix $O = [\vec{v}_1, \dots, \vec{v}_m]$, bestehend aus den normierten Eigenvektoren, so dass gilt:

$$R = O \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & & & \vdots \\ \vdots & & & 0 \\ 0 & \dots & 0 & \lambda_m \end{pmatrix} O^T.$$

Die Eigenwerte können als Varianzen der entsprechenden Hauptkomponenten interpretiert werden.

Die Aufgabe der Hauptkomponentenanalyse besteht nun darin, zu den gegebenen Zufallsvariablen X_1, \dots, X_m neue Variablen Y_1, \dots, Y_m zu finden, die unkorreliert sind und deren Varianz mit wachsendem Index $j = 1, \dots, m$ fallen. Dabei lässt sich jedes Y_j als Linearkombination der X schreiben:

$$Y_j = \sum_{i=1}^n v_{ij} X_i.$$

Aus dem oben Genannten folgt gerade, dass $\vec{v}_j = (v_{1j}, \dots, v_{mj})^T$ der Eigenvektor zum Eigenwert λ_j ist.

Zum einen lassen sich nun anhand der Hauptvektoren Gruppierungen der Variablen erkennen und zum anderen gibt die Anzahl der „großen“ Eigenwerte die effektive Dimension der Daten an. Dieses ermöglicht eine Reduzierung der Dimension der Daten.

2.2 Zur Singulärwertzerlegung

Jede reelle (m, n) -Matrix kann wie folgt unterteilt werden:

$$A = U \cdot S \cdot V^T.$$

Dabei handelt es sich bei U um eine orthogonale (m, m) -Matrix, bei V um eine orthogonale (n, n) -Matrix und bei S um eine eindeutige (m, n) -Matrix, welche nur reelle nicht negative Elemente σ_i , $i = 1, \dots, \min(m, n)$ den Singulärwerten (SV) auf der Diagonale besitzt. Diese sind in absteigender Reihenfolge angeordnet, so dass $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m, n)} \geq 0$ gilt. S hat dann die Form

$$\begin{pmatrix} \Sigma \\ 0 \end{pmatrix}, \text{ wenn } m \geq n \text{ und } [\Sigma \ 0] \text{ wenn } m < n,$$

wobei Σ eine Diagonalmatrix mit den Diagonalelementen σ_i ist. Somit kann S auch zur Rangbestimmung verwendet werden.

Wie allgemein bekannt, bietet die SVD eine numerisch robuste Methode, um ein lineares Ausgleichsproblem zu lösen:

$$x = (A^T A)^{-1} A^T b.$$

Ersetzt man $A = U \cdot S \cdot V^T$, folgt

$$x = V[\Sigma^{-1}]U^T b.$$

Für weitere Details wird auf [BOC98] und [GOL96] verwiesen.

Hat man eine Modellfunktion $F_D(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ für ein Problem und einen Datensatz D gegeben, so bestimmt man für einen Startvektor \vec{x}^0 die erste Ableitung und führt die SVD durch:

$$DF_D(\vec{x}^0) = \begin{pmatrix} * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \end{pmatrix} \Leftrightarrow DF_D(\vec{x}^0) = U \cdot S \cdot V^T.$$

Um nun mit der SVD die Daten zu analysieren, verwendet man folgende Interpretation von $u_{i,j}$ und $v_{i,j}$:

$$\begin{aligned} u_{i,j} &= \text{Anteil vom } i - \text{ten Datensatz am } j - \text{ten SV} \\ v_{i,j} &= \text{Anteil vom } i - \text{ten Parameter am } j - \text{ten SV.} \end{aligned}$$

Meist gilt nun für die Singulärwerte

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_l \gg \sigma_{l+1} \geq \dots \geq \sigma_{\min(m,n)} \geq 0.$$

Die Vermutung liegt nun nahe, dass das System überwiegend durch die ersten l Singulärwerte charakterisiert ist. Über den obigen Zusammenhang lässt sich nun feststellen, welche Datensätze und auch Parameter gerade zu diesen charakterisierenden Singulärwerten gehören. Die übrigen lässt man zur Reduzierung der Dimension entfallen. Dieses Verfahren findet seine Anwendung z.B. im Rahmen der Bildkompression, siehe [ARN00].

2.3 Zur Maximum-Likelihood-Methode

Sei $\{D := d_1, \dots, d_n\}$ der beobachtete Datensatz und M das stochastische Modell, welches D angeblich erzeugt. M wird hierbei durch die Parameter \vec{v} repräsentiert. Durch den durch Nebenbedingungen definierten Parameterraum $\vec{v} \in I$ ergibt sich somit eine Klasse von Modellen $M_{\vec{v}}$.

Definition 2.9

Für eine Fallstudie, welche von den Parametern $\vec{v} \in I$ abhängt, erhält man die Modellklasse:

$$M_{\vec{v}} = \{M(\vec{v}), \vec{v} \in I\}.$$

Nun kann man die Wahrscheinlichkeit $L_D(M)$ des Modells in Abhängigkeit des Datensatzes berechnen.

Definition 2.10

Die Likelihood-Funktion ist für einen gegebenen Datensatz D definiert durch:

$$L_D(M_{\vec{v}}) := P(D|M_{\vec{v}}) = \prod_{i=1}^n p(d_i|M_{\vec{v}}).$$

Die Likelihood-Funktion ist also definiert als die bedingte Wahrscheinlichkeit P , wobei der Datensatz D im Falle von $M_{\vec{v}}$ entstanden ist. Um nun den besten Parametersatz innerhalb dieser Modellfamilie zu finden, verwendet man die übliche statistische Vorgehensweise des *Maximum-Likelihood*. Man sucht den Parametersatz \vec{v} , der die Wahrscheinlichkeit maximiert. In einfachen Fällen ist $L_D(M_{\vec{v}})$ durch einen geschlossenen Ausdruck gegeben, für den man das Maximum direkt berechnen kann. Meist erhält man jedoch Sattelpunkte anstelle eines lokalen Maximums.

Gewöhnlich wandelt man diese Maximumsuche in eine Minimumsuche um. Hierfür definiert man sich die log-Likelihoodfunktion.

Definition 2.11

Die Log-Likelihood-Funktion ist für einen gegebenen Datensatz D definiert durch:

$$F_D(M_{\vec{v}}) := -\log(L_D(M_{\vec{v}})).$$

Da für eine Wahrscheinlichkeit P immer $0 \leq P \leq 1$ gilt, ergibt sich für $-\log(P)$ $\infty \geq -\log(P) \geq 0$ und mit der Monotonie des Logarithmus erhält man somit die Umwandlung von der Maximumsuche in die Minimumsuche. Da die bedingten Wahrscheinlichkeiten $p(d_1|M)$ oft sehr klein sind, stellt sich das Problem der Handhabbarkeit des Produktes derselben. Wegen

$$-\log\left(\prod_{i=1}^n p(d_i|M_{\vec{v}})\right) = \sum_{i=1}^n -\log(p(d_i|M_{\vec{v}}))$$

löst sich dieses Problem aber bei der Log-Likelihood-Funktion.

Noch zu beachten ist, dass es sich bei den beobachteten Prozessen oft um kontinuierliche und nicht diskrete handelt, siehe dazu Fallstudie 1.3. Dabei erhält man für die

Zustände der Zufallsvariablen Z eine Dichtefunktion $\hat{p}_Z(z)$ (z.B. Abbildung 2.1) anstelle einer diskreten Wahrscheinlichkeit. Diskretisiert man diese Dichte z.B. äquidistant mit Δp_Z , lässt sich die Wahrscheinlichkeit durch $p(z) := \hat{p}_Z(z)\Delta p_Z$ abschätzen. Setzt man dies in die Log-Likelihood-Funktion ein, ergibt sich folgendes:

$$\begin{aligned} F_D(M_{\vec{v}}) &= \sum_{i=1}^n -\log(p(d_1|M_{\vec{v}})) = \sum_{i=1}^n -\log(p_Z(d_1|M_{\vec{v}})\Delta p_Z) \\ &= \sum_{i=1}^n -\log(p_Z(d_1|M_{\vec{v}})) - n\log(\Delta p_Z). \end{aligned}$$

Der letzte Summand ist nicht von den Daten abhängig und konstant, muss also bei der Minimierung nicht berücksichtigt werden. D.h. bei kontinuierlichen Zufallsvariablen lässt sich anstelle der Wahrscheinlichkeit die Dichte verwenden. Da jedoch die Dichte im Allgemeinen größer Eins ist, treten hierbei negative Werte durch den Term $-\log$ auf.

2.4 Zur Minimierung

In den folgenden Unterkapiteln wird ein kurzer Einblick in die Verfahrensvorschriften der in dieser Arbeit zur Anwendung kommenden Verfahren gegeben. Für genauere Erläuterungen und theoretische Überlegungen (wie Konvergenzsätze) sei auf die Literatur [ALT02] und [SPE93] verwiesen.

Nichtlineare Optimierungsaufgaben sind in der Regel nicht analytisch sondern nur numerisch zu lösen. Hierbei berechnet man ausgehend von einem Startwert \vec{x}^0 eine Folge $\vec{x}^{(k)}$, $k = 1, 2, \dots$, die als Grenzwert eine lokale Lösung hat. Ist das Minimum gesucht, liegt es daher nahe, die $\vec{x}^{(k)}$ so zu bestimmen, dass

$$f(\vec{x}^{(k+1)}) < f(\vec{x}^{(k)})$$

gilt. Verfahren mit einer solchen Strategie nennt man *Abstiegsverfahren*. Ein derartiges Verfahren berechnet $\vec{x}^{(k+1)}$ mit Hilfe einer Abstiegsrichtung $d^{(k)}$, welches also die Eigenschaft

$$f(\vec{x}^{(k)} + td^{(k)}) < f(\vec{x}^{(k)}) \quad \forall t \in]0, s_k[$$

mit $s_k > 0$ besitzt. Zur Konstruktion von Abstiegsverfahren gibt es nun zwei Vorgehensweisen:

Schrittweitengesteuerte Verfahren: Man bestimmt aufgrund lokaler Informationen über die Zielfunktion im Iterationspunkt $x^{(k)}$ eine Abstiegsrichtung $d^{(k)}$ und anschließend berechnet man eine Schrittweite $\sigma_k \in]0, s_k[$, mit der man einen größtmöglichen Abstieg erzielt und setzt $x^{(k+1)} = x^{(k)} + \sigma_k d^{(k)}$.

Trust-Region-Verfahren: Man bestimmt basierend auf einem lokalem Modell der Zielfunktion (z.B. quadratische Approximation) eine Trust-Region (Vertrauensbereich), auf der das Modell die Zielfunktion hinreichend gut approximiert. Das lokale Modell erlaubt dann die Berechnung einer Abstiegsrichtung $d^{(k)}$ und man setzt $x^{(k+1)} = x^{(k)} + d^{(k)}$.

2.4.1 Ableitungsfreie Minimierungsverfahren

Da die Zielfunktion eines Minimierungsverfahrens oft nicht überall differenzierbar oder deren Berechnung sehr aufwendig ist, wurden Verfahren entwickelt, die keine Ableitung verwenden. Im weiteren Verlauf wird hier das Nelder-Mead-Verfahren zur Lösung von unrestrictierten Optimierungsaufgaben vorgestellt. Dabei lautet das zugrunde liegende Optimierungsproblem:

$$\boxed{\begin{array}{c} \min f(\vec{x}) \\ x \in \mathbb{R}^n \end{array}} .$$

Das Nelder-Mead-Verfahren

Obwohl das Verfahren auch „Simplexverfahren von Nelder-Mead“ genannt wird, hat es nichts mit dem Simplexverfahren der linearen Optimierung zu tun. Der Name beruht auf der Tatsache, dass das Verfahren einen Simplex als Grundstruktur verwendet.

Definition 2.12

Seien $\vec{x}^0, \dots, \vec{x}^n \in \mathbb{R}^n$ affin unabhängige Vektoren, d.h., die Vektoren $\vec{x}^i - \vec{x}^0$, $i = 1, \dots, n$ seien linear unabhängig. Dann heißt die Menge

$$S = \left\{ \sum_{i=0}^n \lambda_i \vec{x}^i \mid \lambda_i \geq 0, i = 0, \dots, n, \sum_{i=0}^n \lambda_i = 1 \right\}$$

(die konvexe Hülle der Punkte $\vec{x}^0, \dots, \vec{x}^n$) (n -dimensionales) Simplex mit den Ecken $\vec{x}^0, \dots, \vec{x}^n$.

Zu Beginn des Verfahrens wird ein Simplex S_0 vorgegeben. Danach durchläuft das Verfahren in jedem Iterationsschritt k folgende Punkte:

- Man berechnet zu dem aktuellen Simplex S_{k-1} mit den Ecken $\vec{x}^0, \dots, \vec{x}^n$ den Punkt \vec{x}^m mit dem größten Funktionswert, also

$$f(\vec{x}^m) = \max f(\vec{x}^0), \dots, f(\vec{x}^n).$$

- Man berechnet einen Punkt mit einem kleineren Funktionswert und ersetzt die Ecke x^m durch diesen neuen Punkt. Damit erhält man den neuen Simplex S_k .

Bemerkung 2.13

Sei $S \subset \mathbb{R}^n$ ein Simplex mit den Ecken $\vec{x}^0, \dots, \vec{x}^n$, so berechnet sich der Schwerpunkt der Ecken bzgl. \vec{x}^j für $j \in 0, \dots, n$ zu

$$\vec{s}^j = \frac{1}{n} \sum_{i=0, i \neq j}^n \vec{x}^i.$$

Das hier angegebene Nelder-Mead-Verfahren [ALT02] benutzt drei Konstruktionsprinzipien zur Bestimmung des neuen Punktes.

1. **Reflektion:** Dabei wird die Ecke \vec{x}^j am Schwerpunkt \vec{s}^j reflektiert und der neue Punkt \vec{x}^r berechnet sich aus

$$\vec{x}^r = \vec{s}^j + \gamma(\vec{s}^j - \vec{x}^j)$$

mit der Reflektionskonstanten $0 < \gamma \leq 1$ (siehe dazu Abbildung 2.2).

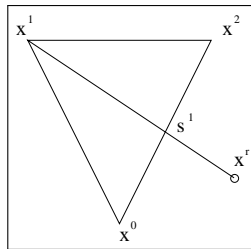


Abbildung 2.2: Reflektion im \mathbb{R}^2 mit $j = 1$ und $\gamma = \frac{1}{2}$.

2. **Expansion:** Dabei berechnet sich der Punkt \vec{x}^r als Verschiebung in Richtung $\vec{s}^j - \vec{x}^j$ (=Richtung $\vec{s}^j - \vec{x}^r$) und der neue Punkt aus

$$\vec{x}^e = \vec{s}^j + \beta(\vec{x}^r - \vec{s}^j)$$

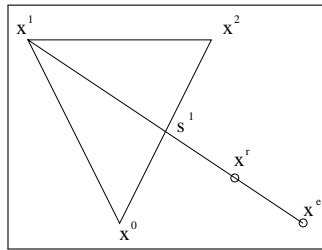
mit der Expansionskonstanten $\beta > 1$ (vgl. Abbildung 2.3).

3. **Kontraktion:** Dabei unterscheidet man drei Untertypen:

- Partielle innere Kontraktion von \vec{x}^j in Richtung $\vec{s}^j - \vec{x}^j$. Der neue Punkt berechnet sich aus

$$\vec{x}^c = \vec{s}^j + \alpha(\vec{x}^j - \vec{s}^j)$$

mit der Kontraktionskonstanten $0 < \alpha < 1$ (vgl. Abbildung 2.4 (links)).

Abbildung 2.3: Expansion mit $\beta = 2$.

- Partielle äußere Kontraktion von \vec{x}^j in Richtung $\vec{s}^j - \vec{x}^r$.
Der neue Punkt berechnet sich aus

$$\vec{x}^c = \vec{s}^j + \alpha(\vec{x}^r - \vec{s}^j)$$

mit der Kontraktionskonstanten $0 < \alpha < 1$ (vgl. Abbildung 2.4 (mitte)).

- Totale Kontraktion bzgl. \vec{x}^j .
Dabei werden die Punkte $\vec{x}^i, i = 0, \dots, n, i \neq j$, durch die Punkte

$$\hat{\vec{x}}^i = \vec{x}^i + \frac{1}{2}(\vec{x}^j - \vec{x}^i) = \frac{1}{2}(\vec{x}^i + \vec{x}^j)$$

ersetzt (vgl. Abbildung 2.4 (rechts)).

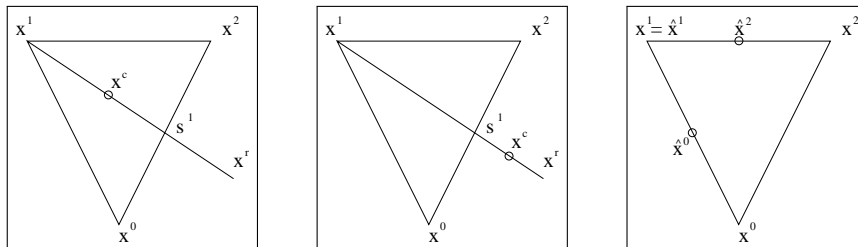


Abbildung 2.4: (links) Innere Kontraktion mit $j = 1$ und $\alpha = \frac{1}{2}$, (mitte) Äußere Kontraktion mit $j = 1$ und $\alpha = \frac{1}{2}$, (rechts) Totale Kontraktion mit $j = 1$.

Mit diesen Konstruktionen lässt sich die Verfahrensvorschrift angeben.

Nelder-Mead-Verfahren

Gegeben sei: $0 < \alpha < 1$ (Kontraktionskonstante), $\beta > 1$ (Expansionskonstante) und $0 < \gamma \leq 1$ (Reflektionskonstante).

1. Wähle einen Startpunkt $\vec{x}^{(0,0)} \in \mathbb{R}^n$ und bestimme die Ecken des Startsimplex S_0 durch

$$\vec{x}^{(0,j)} = \vec{x}^{(0,0)} + \vec{e}^j, \quad j = 1, \dots, n,$$

wobei $\vec{e}^j \in \mathbb{R}^n$ die Einheitsvektoren des \mathbb{R}^n sind. Setze $k := 0$.

2. Bestimme den Punkt $\vec{x}^{(k,m)}$ mit

$$f(\vec{x}^{(k,m)}) = \max\{f(\vec{x}^{(k,0)}), \dots, f(\vec{x}^{(k,n)})\},$$

die Ecke von S_k mit dem größten Funktionswert und den Punkt $\vec{x}^{(k,l)}$ mit

$$f(\vec{x}^{(k,l)}) = \min\{f(\vec{x}^{(k,0)}), \dots, f(\vec{x}^{(k,n)})\},$$

die Ecke von S_k mit dem kleinsten Funktionswert und

$$\vec{s}^{(k,m)} = \frac{1}{n} \sum_{i=0, i \neq m}^n \vec{x}^{(k,i)}$$

den Schwerpunkt der Ecken bzgl. $\vec{x}^{(k,m)}$.

3. Führe eine Reflektion von $\vec{x}^{(k,m)}$ am Schwerpunkt $\vec{s}^{(k,m)}$ durch, d.h., berechne den Punkt

$$\vec{x}^r = \vec{s}^{(k,m)} + \gamma(\vec{s}^{(k,m)} - \vec{x}^{(k,m)}).$$

Dabei erwartet man, dass $f(\vec{x}^r) < f(\vec{x}^{(k,m)})$ ist.

4. Nun unterscheidet man drei Fälle:

- (a) Es ist $f(\vec{x}^r) < f(\vec{x}^{(k,m)})$, d.h., der Reflektionsschritt hat einen neuen minimalen Punkt erzeugt. Dann versucht man durch Expansion von \vec{x}^r in Richtung $\vec{x}^r - \vec{s}^{(k,m)}$ einen noch besseren Punkt zu finden; man berechnet

$$\vec{x}^e = \vec{s}^j + \beta(\vec{x}^r - \vec{s}^{(k,m)})$$

und ersetzt $\vec{x}^{(k,m)}$ durch den besseren der beiden Punkte \vec{x}^r und \vec{x}^e , d.h. man setzt

$$\vec{x}^{(k+1,l)} := \begin{cases} \vec{x}^e, & \text{falls } f(\vec{x}^e) < f(\vec{x}^r) \\ \vec{x}^r, & \text{falls } f(\vec{x}^r) \leq f(\vec{x}^e). \end{cases}$$

Die restlichen Ecken von S_k bleiben unverändert.

- (b) Es ist

$$f(\vec{x}^{(k,l)}) \leq f(\vec{x}^r) \leq \max\{f(\vec{x}^{(k,j)}) | j \neq m\}.$$

Dann ist \vec{x}^r nicht schlechter als alle Ecken außer $\vec{x}^{(k,l)}$, in der Regel besser als $\vec{x}^{(k,m)}$, und man ersetzt $\vec{x}^{(k,m)}$ durch \vec{x}^r . Die restlichen Ecken von S_k bleiben unverändert.

- (c) Es ist

$$f(\vec{x}^r) > \max\{f(\vec{x}^{(k,j)}) | j \neq m\}.$$

Ist $f(\vec{x}^r) \geq f(\vec{x}^{(k,m)})$, dann war es möglicherweise falsch, von $\vec{s}^{(k,m)}$ in Richtung \vec{x}^r zu gehen und man versucht die entgegengesetzte Richtung;

man führt eine partielle innere Kontraktion von $\vec{x}^{(k,m)}$ in Richtung $\vec{s}^{(k,m)} - \vec{x}^{(k,m)}$ aus, d.h., man berechnet

$$\vec{x}^c = \vec{s}^{(k,m)} + \alpha(\vec{x}^{(k,m)} - \vec{s}^{(k,m)}).$$

Ist $f(\vec{x}^r) < f(\vec{x}^{(k,m)})$, dann war es möglicherweise richtig, von $\vec{s}^{(k,m)}$ in Richtung \vec{x}^r zu gehen; da aber alle Ecken außer $\vec{x}^{(k,m)}$ besser als \vec{x}^r sind, sollte man wieder etwas näher an das Simplex gehen; man führt eine partielle äußere Kontraktion von \vec{x}^r in Richtung $\vec{s}^{(k,m)} - \vec{x}^r$ aus, d.h., man berechnet

$$\vec{x}^c = \vec{s}^{(k,m)} + \alpha(\vec{x}^r - \vec{s}^{(k,m)}).$$

Ist $f(\vec{x}^c) < f(\vec{x}^{(k,m)})$, dann setzt man $\vec{x}^{(k+1,m)} := \vec{x}^c$, und die restlichen Ecken von S_k bleiben unverändert. Andernfalls haben alle Versuche nichts gebracht; man führt eine totale Kontraktion bzgl. $\vec{x}^{(k,l)}$ durch, d.h., für $i \neq l$ setzt man

$$\vec{x}^{(k+1,i)} = \frac{1}{2}(\vec{x}^{(k,i)} + \vec{x}^{(k,l)}).$$

Dabei bleibt die Ecke mit minimalem Funktionswert gleich.

5. Setze $k := k + 1$ und gehe zu 2.

Als aktuellen Iterationspunkt in Iteration k lässt sich hierbei der Punkt $\vec{x}^{(k,l)}$ betrachten. Da explizit keine Suchrichtungen und keine Schrittweiten bestimmt werden, sind allgemeine Konvergenzaussagen, wie im Fall von gradientenbasierten Verfahren, nicht bekannt. Durch dieses Verfahren ist nur sichergestellt, dass

$$f(\vec{x}^{(k+1,l)}) \leq f(\vec{x}^{(k,l)}).$$

Als Abbruchkriterium verwendet die Matlab-Funktion `fminsearch` den Durchmesser von S_k . D.h. es bricht ab, wenn der Durchmesser von S_k hinreichend klein ist (Parameter (TolX)).

2.4.2 Gradientenbasierte Minimierungsverfahren

$$\boxed{\boxed{\begin{array}{c} \min f(\vec{x}) \\ x \in \mathbb{R}^n \end{array}}}$$

Wie zu Beginn des Unterkapitels 2.1 schon beschrieben berechnet man zur Lösung des oben stehenden Problems im $(k + 1)$ -ten Iterationsschritt zum Iterationswert $\vec{x}^{(k)}$ eine Abstiegsrichtung $\vec{d}^{(k)}$. Die üblichen Verfahren, wie z.B. das Gradienten-

oder auch das Newtonverfahren, werden in dieser Arbeit nicht weiter behandelt und auch nicht vorgestellt. Jedoch zu den Trust-Region-Verfahren und den Verfahren zur Lösung nichtlinearer Ausgleichsprobleme soll eine kleine Einführung erfolgen.

Trust-Region-Verfahren

Bei dem Trust-Region-Verfahren wird ein lokales Modell f_k für die Zielfunktion f benutzt, beispielsweise

$$f_k(\vec{d}) = f(\vec{x}^{(k)}) + \nabla f(\vec{x}^{(k)})^T \vec{d}$$

oder

$$f_k(\vec{d}) = f(\vec{x}^{(k)}) + \nabla f(\vec{x}^{(k)})^T \vec{d} + \frac{1}{2} \vec{d}^T f''(\vec{x}^{(k)}) \vec{d}.$$

Außerdem wird ein Radius $\rho_k > 0$ definiert, der einen Vertrauensbereich $B(\vec{x}^{(k)}, \rho_k)$ (trust region) für das Modell definiert. Damit berechnet man die Suchrichtung $\vec{d}^{(k)}$ als globale Lösung des Minimierungsproblems

$$\min_{\|\vec{d}\| \leq \rho_k} f_k(\vec{d}).$$

Verwendet man das obere Beispiel, so erhält man, falls $\nabla f(\vec{x}^{(k)}) \neq 0$ ist, als Lösung

$$\vec{d}^{(k)} = -\rho_k \frac{\nabla f(\vec{x}^{(k)})}{\|\nabla f(\vec{x}^{(k)})\|},$$

welche lokal mit der Richtung des Gradientenverfahrens übereinstimmt.

Die zugrunde liegende Idee beim Trust-Region-Verfahren liegt darin, den Radius ρ_k so zu wählen, dass die Genauigkeit der Modellfunktion f_k auf $B(\vec{x}^{(k)}, \rho_k)$ ausreicht, um ihr zu „trauen“. Dieses erreicht man, indem man überprüft, ob mit $\vec{d}^{(k)}$ der Abstieg des Funktionswertes annähernd dem Abstieg der Modellfunktion ist. Andernfalls war die Trust-Region zu groß und man verkleinert den Radius ρ_k , um erneut $\vec{d}^{(k)}$ zu bestimmen.

Exemplarisch wird dies am zweiten Beispiel

$$f_k(\vec{d}) = f(\vec{x}^{(k)}) + \nabla f(\vec{x}^{(k)})^T \vec{d} + \frac{1}{2} \vec{d}^T f''(\vec{x}^{(k)}) \vec{d}$$

vorgestellt und man erhält somit das Trust-Region-Newton-Verfahren.

Trust-Region-Newton-Verfahren

Gegeben sei: $0 < \delta_1 < \delta_2 < 1, \sigma_1 \in]0, 1[, \sigma_2 > 1$ und ein Radius $\rho > 0$.

1. Wähle einen Startpunkt $\vec{x}^{(0)} \in \mathbb{R}^n$, berechne $\vec{b}^{(0)} = \nabla f(\vec{x}^{(0)})$, $A(0) = f''(\vec{x}^{(0)})$ und setze $k := 0$.

2. Berechne eine globale Lösung $d^{(k)}$ des Problems

$$\min_{\|\vec{d}\| \leq \rho_k} f_k(\vec{d}) = f(\vec{x}^{(k)}) + (\vec{b}^{(k)})^T \vec{d} + \frac{1}{2} \vec{d}^T A^{(k)} \vec{d},$$

falls $(\vec{x}^{(k)}) = f_k(\vec{d}^{(k)}) \Rightarrow$ Stopp.

3. Berechne

$$r_k = \frac{f(\vec{x}^{(k)}) - f(\vec{x}^{(k)} + \vec{d}^{(k)})}{f(\vec{x}^{(k)}) - f_k(\vec{d}^{(k)})}.$$

Ist $r_k \geq \delta_1$ (erfolgreicher Iterationsschritt):

- Setze $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \vec{d}^{(k)}$.
- Berechne $\vec{b}^{(k+1)} = \nabla f(\vec{x}^{(k+1)})$ und $A^{(k+1)} = f''(\vec{x}^{(k+1)})$.
- Aktualisiere ρ_k :
 Falls $r_k \in [\delta_1, \delta_2[$, dann wähle $\rho_{k+1} \in [\sigma_1 \rho_k, \rho_k]$;
 falls $r_k \geq \delta_2$, dann wähle $\rho_{k+1} \in [\rho_k, \sigma_2 \rho_k]$.
- Setze $k := k + 1$ und gehe zu 2.

4. Es war $r_k < \delta_1$ (nicht erfolgreicher Iterationsschritt):

- Wähle $\rho_{k+1} \in]0, \sigma_1 \rho_k]$.
- Setze $\vec{x}^{(k+1)} := \vec{x}^{(k)}$, $\vec{b}^{(k+1)} := \vec{b}^{(k)}$, $A^{(k+1)} := A^{(k)}$.
- Setze $k := k + 1$ und gehe zu 2.

Die Erhöhung des Iterationszählers in Schritt 4 erfolgt, um auch die nicht erfolgreichen Iterationsschritte mitzuzählen.

Nichtlineare Ausgleichsprobleme

Sei $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ eine zweimal stetig differenzierbare Abbildung und $m > n$. Somit lässt sich das nichtlineare Ausgleichsproblem schreiben als

$$\boxed{\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x}) = \min_{\vec{x} \in \mathbb{R}^n} \frac{1}{2} \|F(\vec{x})\|^2}$$

Mit Hilfe der besonderen Struktur des Ausgleichproblems können sich viele Optimierungsverfahren für diese Problemklasse modifizieren. Aus dem obigen Zusammenhang heraus gilt für die Ableitungen der Zielfunktion f

$$\nabla f(\vec{x}) = F'(\vec{x})^T F(\vec{x})$$

und

$$f''(\vec{x}) = F'(\vec{x})^T F'(\vec{x}) + R(\vec{x}) \quad \text{mit} \quad R(\vec{x}) = \sum_{i=1}^m F_i(\vec{x}) F_i''(\vec{x}).$$

Zur Vereinfachung wird hierbei oft der Term $R(\vec{x})$ unberücksichtigt gelassen, da in der Nähe der Lösung \tilde{x} die Funktion $f(\tilde{x})$ klein ist und somit die $F_i(\tilde{x})$ sehr klein sind.

Gauß-Newton-Verfahren

Wendet man das gedämpfte Newton-Verfahren auf

$$\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x}) = \min_{\vec{x} \in \mathbb{R}^n} \frac{1}{2} \|F(\vec{x})\|^2$$

an, so erhält man als Problem für die Richtungsableitung

$$\min_{\vec{d} \in \mathbb{R}^n} \nabla f(\vec{x}^{(k)})^T \vec{d} + \frac{1}{2} \vec{d}^T F'(\vec{x}^{(k)})^T F'(\vec{x}^{(k)}) \vec{d},$$

das so genannte Gauß-Newton-Verfahren. Eine einfache Rechnung zeigt, dass das Problem zur Berechnung der Gauß-Newton-Richtung äquivalent zu

$$\min_{\vec{d} \in \mathbb{R}^n} \frac{1}{2} \left\| F(\vec{x}^{(k)}) + F'(\vec{x}^{(k)}) \vec{d} \right\|^2$$

ist.

Levenberg-Marquardt-Verfahren

Wendet man entsprechend der Trust-Region-Modifikation des gedämpften Newton-Verfahrens eine solche Modifikation des Gauß-Newton-Verfahrens an, so erhält man als Problem für die Richtungsableitung

$$\min_{\vec{d} \in \mathbb{R}^n} f_k(d) = f(\vec{x}^{(k)}) + [F'(\vec{x}^{(k)})^T F(\vec{x}^{(k)})]^T \vec{d} + \frac{1}{2} \vec{d}^T F'(\vec{x}^{(k)})^T F'(\vec{x}^{(k)}) \vec{d},$$

das so genannte Levenberg-Marquardt-Verfahren. Dieses lässt sich durch eine einfache Rechnung wiederum in das äquivalente Problem

$$\min_{\vec{d} \in \mathbb{R}^n} \frac{1}{2} \left\| F(\vec{x}^{(k)}) + F'(\vec{x}^{(k)}) \vec{d} \right\|^2$$

mit der Restriktion $\|\vec{d}\| \leq \rho_k$ überführen.

Hierbei muss über den lokal zu wählenden Parameter ρ_k noch diskutiert werden. Diese Nebenbedingung lässt sich in üblicher Weise durch einen Lagrange-Multiplikator $p \geq 0$ an das Minimierungsproblem koppeln, d.h.

$$p \left(\|\vec{d}\|^2 - \rho_k \right) \geq 0.$$

Daraus folgt für $\|\vec{d}\|^2 = \rho_k$, dass $p > 0$ sein kann und für $\|\vec{d}\| < \rho_k$, dass $p = 0$ gilt. Gekoppelt erhält man:

$$\min_{\vec{d} \in \mathbb{R}} \left\| F(\vec{x}) + F'(\vec{x})\vec{d} \right\| + p \|\vec{d}\|.$$

Nach üblicher Vorgehensweise ergibt sich für die Minimierung dieser Funktion eine Gleichung für \vec{d} mit

$$(F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n) \vec{d}_k = -F'(\vec{x}_k)^T F(\vec{x}_k).$$

Diese lässt sich, da die auf der linken Seite auftretende Matrix symmetrisch, positiv definit und somit invertierbar ist, auflösen zu:

$$\vec{d}_k = - (F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n)^{-1} F'(\vec{x}_k)^T F(\vec{x}_k).$$

Um nun zu zeigen, dass es sich hierbei um eine Abstiegsrichtung handelt, d.h. $F(\vec{x}_k) \geq F(\vec{x}_{k+1})$ gilt, betrachtet man den Winkel zwischen \vec{d}_k und dem Gradienten des Minimierungsproblems also $2F'(\vec{x}_k)^T F(\vec{x}_k)$. Die Beweisführung folgt derselben in [KUH80].

Satz 2.14

Der Winkel zwischen $F'(\vec{x}_k)^T F(\vec{x}_k) := G(\vec{x}_k)$ und \vec{d}_k ist immer stumpf und damit ist \vec{d}_k ein Abstiegsvektor, solange $F'(\vec{x}_k)^T F(\vec{x}_k) \neq 0$ ist.

Beweis:

$$(F'(\vec{x}_k)^T F'(\vec{x}_k)) \vec{d}_k = -G(\vec{x}_k)^T (F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n)^{-1} G(\vec{x}_k).$$

Hierbei ist $(F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n)^{-1}$ symmetrisch und positiv definit. Daraus folgt, dass der Ausdruck kleiner 0 und somit der Winkel zwischen beiden Vektoren stumpf ist. \diamond

Interessant ist das Verhalten von \vec{d}_k bezüglich des Parameters p . Hierzu folgender Satz:

Satz 2.15

Die euklidische Norm $\|\vec{d}_k\|_2$ des wie oben definierten Abstiegsvektors ist mit wachsendem p eine monoton fallende Funktion.

Beweis:

$F'(\vec{x}_k)^T F'(\vec{x}_k)$ ist eine symmetrische, semidefinite Matrix. Daher existiert eine orthogonale Matrix $U \in \mathbb{R}^{n \times n}$ mit

$$U^T F'(\vec{x}_k)^T F'(\vec{x}_k) U = D$$

und

$$U^T F'(\vec{x}_k)^T F'(\vec{x}_k + pI)U = D + pI$$

mit $D = \text{diag}(\delta_1, \dots, \delta_n)$, $\delta_i \geq 0$. Aus

$$\vec{d}_k = - (F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n)^{-1} F'(\vec{x}_k)^T F(\vec{x}_k).$$

und dem eben erkannten ergibt sich für die Norm

$$\begin{aligned} \|\vec{d}_k\|_2^2 &= \vec{d}_k^T \vec{d}_k = F(\vec{x}_k)^T F'(\vec{x}_k)U(D + pI)^{-1}U^T U(D + pI)^{-1}U^T F'(\vec{x}_k)^T F(\vec{x}_k) \\ &= \sum_{j=1}^n \frac{h_j^2}{(\delta_j + p)^2} \end{aligned}$$

mit $h := U^T F'(\vec{x}_k)^T F(\vec{x}_k) = (h_1, \dots, h_n)^T$.

Hieraus folgt die oben genannte Behauptung. \diamond

Eine Aussage über die Konvergenzgeschwindigkeit des Levenberg-Marquardt liefert der folgende Satz (mit Beweis nach [KUH80]).

Satz 2.16

Sei $B \subset \mathbb{R}^n$ offen und konvex, $\vec{x}^* \in B$ ein stationärer Punkt, $F(\vec{x}) \in C^{1,1}(B)$ und für eine Umgebung $U(\vec{x}^*)$ gelte für die Jacobimatrix $F'(\vec{x})$ die Lipschitz-Bedingung

$$\|F'(\vec{x}) - F'(\vec{x}')\| \leq c_1 \|\vec{x} - \vec{x}'\| \quad \forall \vec{x}, \vec{x}' \in U(\vec{x}^*) \text{ und ein } c_1 \in \mathbb{R}^+.$$

Dann konvergiert das Levenberg-Marquardt Verfahren von jedem Punkt $\vec{x} \in U(\vec{x}^*)$ gegen einen stationären Punkt aus $U(\vec{x}^*)$ mit $\|\vec{x} - \vec{x}^*\| \leq r < \infty$. Die Konvergenz erfolgt zumindest linear.

Beweis:

Mit D und U aus dem vorherigen Satz sei $((\delta_1)_k + p_k)$ der kleinste Eigenwert der Matrix $F'(\vec{x}_k)^T F'(\vec{x}_k) + p_k I_n$. Hieraus ergibt sich nun

$$\|F'(\vec{x}_k)^T F'(\vec{x}_k) + p_k I_n\| = \|U_k^T (D + p_k I)^{-1} U_k\| \leq ((\delta_1)_k + p_k)^{-1}.$$

Aufgrund der Taylorentwicklung von $F(\vec{x})$ um \vec{x}^* ergibt sich für ein $x \in [\vec{x}_k, \vec{x}^*]$

$$F(\vec{x}_k) = F(\vec{x}^*) + F'(\vec{x})(\vec{x}_k - \vec{x}^*).$$

Hieraus lässt sich mit Hilfe der Voraussetzung $2F'(\vec{x}^*)^T F(\vec{x}^*) = 0$ und der Dreiecksungleichung herleiten:

$$\begin{aligned} \|\vec{x}_{k+1} - \vec{x}^*\| &= \|\vec{x}_k + \vec{d}_k - \vec{x}^*\| \\ &= \|\vec{x}_k - (F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n)^{-1} F'(\vec{x}_k)^T F(\vec{x}_k) - \vec{x}^*\| \\ &\leq \| - (F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n)^{-1} \| \\ &\quad \| (F'(\vec{x}_k)^T F'(\vec{x}_k) + pI_n) (\vec{x}_k - \vec{x}^*) + F'(\vec{x}_k)^T F(\vec{x}_k) \| \end{aligned}$$

Setzt man die Taylorentwicklung für $F(\vec{x}_k)$ ein, so ergibt sich:

$$= ((\delta_1)_k + p_k)^{-1} \left\| (F'(\vec{x}_k)^T F'(\vec{x}_k) + p_k I_n - F'(\vec{x}_k)^T F'(\vec{x})) (\vec{x}_k - \vec{x}^*) + (F'(\vec{x}_k) - F'(\vec{x}^*))^T F(\vec{x}^*) \right\|$$

und mit der Dreiecksungleichung

$$\leq ((\delta_1)_k + p_k)^{-1} (p_k + \|F'(\vec{x}_k)\| \|F'(\vec{x}_k) - F'(\vec{x})\| + c_1 \|F(\vec{x}^*)\|) \|\vec{x}_k - \vec{x}^*\| \\ \leq ((\delta_1)_k + p_k)^{-1} (p_k + c_2 c_1 \|\vec{x}_k - \vec{x}^*\| + c_1 \|F(\vec{x}^*)\|) \|\vec{x}_k - \vec{x}^*\|$$

mit $c_3 = \sup_{\vec{x} \in U(\vec{x}^*)} \|F'(\vec{x})\|$, welches wegen $F(\vec{x}) \in C^{1,1}(B)$ existiert.

Für $b_1 := \sup_{k > k_0} ((\delta_1)_k + p_k)^{-1} (p_k + c_2 c_1 r + c_1 \|F(\vec{x}^*)\|)$ ergibt sich

$$\lim_{k \rightarrow \infty} \frac{\|\vec{x}_{k+1} - \vec{x}^*\|}{\|\vec{x}_k - \vec{x}^*\|} \leq b_1 < \infty.$$

Wählt man $r > 0$ so klein, dass $((\delta_1)_k + p_k) \geq p_k + c_2 c_1 r + c_1 \|F(\vec{x}^*)\|$, also $(\delta_1)_k \geq c_2 c_1 r + \|F(\vec{x}^*)\|$ für $k > k_0$ erfüllt ist, so gilt $b_1 < 1$ und somit folgt aus $\|\vec{x}_{k+1} - \vec{x}^*\| < b_1 \|\vec{x}_k - \vec{x}^*\|$ zumindest eine lineare Konvergenz. \diamond

Ein Problem des Levenberg-Marquardt Verfahrens besteht darin, dass es sich im Gegensatz zum Gauß-Newton Verfahren nicht als affin-invariant erweist. D.h. die Lösung des Problems $F(x) = 0$ ist nicht äquivalent zu $G(x) := AF(x) = 0$, wobei A eine invertierbare Matrix sei. Dagegen steht jedoch die Robustheit dieses Verfahrens, da es auch bei rangdefekten Jacobimatrizen Ergebnisse aufweist. Bei entsprechender Implementierung geht auch der Parameter p in der Nähe von lokalen Minima gegen Null und das Verfahren wechselt über zum Gauß-Newton Verfahren.

Das Fehlen der Affinvarianz lässt sich durch eine Modifikation, welche schon Levenberg anregte, abschwächen. Anstelle der Einheitsmatrix I verwendet man eine positiv definite Matrix $D^T D$, welche zu folgendem Iterationsschritt führt:

$$\vec{d}_k = - (F'(\vec{x}_k)^T F'(\vec{x}_k) + p D^T D)^{-1} F'(\vec{x}_k)^T F(\vec{x}_k).$$

Damit erhält man eine Invarianz des Levenberg-Marquardt Verfahrens bezüglich der Skalierung des Parameterraumes (nach [KUH80]). Das Verfahren liefert die gleichen Ergebnisse, wenn man mit \vec{x}^0 startet und die Funktion $F(\vec{x})$ minimiert, bzw. wenn man bei $\vec{x}^+ := D\vec{x}^*$ beginnt und die Funktion $F^+(\vec{x}) := F(D^{-1}\vec{x})$ minimiert. Für den negativen Gradienten des Verfahrens $-2F'(\vec{x}_k)^T F(\vec{x}_k)$ ergibt sich nämlich

$$F'(\vec{x}) = \frac{dF(\vec{x}^+)}{d\vec{x}} = \frac{dF(\vec{x}^+)}{d\vec{x}^+} \frac{d\vec{x}^+}{d\vec{x}} = F'(\vec{x}^+) D^{-1} = F'^+ \text{ und } F^+ = F^+(\vec{x}^+) = F(D^{-1}\vec{x}^+) = F(\vec{x}) = F \Rightarrow -2F'^+(\vec{x}_k)^T F^+(\vec{x}_k) = -D^{-1} (2F'(\vec{x}_k)^T F(\vec{x}_k)),$$

also der mit D^{-1} abgebildete Gradient des nicht transformierten Raumes. Den Iterationsschritt erhalten wir wie folgt:

$$\begin{aligned}\vec{d}_k &= -(F'^T F' + pD^T D)^{-1} F'^T F \\ &= -D^{-1} (F'^{+T} F'^+ + pI)^{-1} D^{-1} F'^T F \\ &= -D^{-1} (F'^{+T} F'^+ + pI)^{-1} F'^{+T} F^+ \\ &= D^{-1} \vec{d}_k^+.\end{aligned}$$

Das bedeutet, dass im transformierten Raum für D wieder die Einheitsmatrix und entsprechend die Umkehrabbildung verwendet wird.

Kapitel 3

Entwicklung von Mutationen in prekanzerösen Zellen

In diesem Kapitel wird mittels eines Populationsmodells die „Entwicklung von Mutationen in prekanzerösen Zellen“ als ein Optimierungsproblem unter Differentialgleichungsrestriktionen beschrieben. Um die Daten reduzieren zu können, wird eine Scoring-Funktion eingeführt. Danach wird ein Lösungsverfahren für große lineare Differentialgleichungssysteme mit konstanter unterer Dreiecksmatrix vorgestellt, welches mit bekannten Lösungsansätzen verglichen wird. Im Folgenden sollen die Ergebnisse visualisiert, diskutiert und interpretiert werden.

Definition 3.1

- a) l sei die Anzahl der berücksichtigten Mutationen.
- b) Jeder Zelltyp ist eindeutig durch einen binären Vektor $\vec{n} = (n_1, \dots, n_l) \in \{0, 1\}^l$ gekennzeichnet, wobei eine Komponente $n_i = 1$ anzeigt, dass eine Mutation vom Typ i vorliegt.

Mit dieser Definition ist die Fallstudie gegeben durch:

Gegeben: Datensatz D

$$\begin{aligned} & \min F_D(\vec{r}, \vec{\alpha}, \vec{x}^0) \\ & a_i \geq 0 \forall i = 0 \dots l, \quad x_{\vec{n}i}^0 \geq 0 \forall i = 0 \dots l, \quad \sum_{i=0}^l x_{\vec{n}i}^0 = 1, \\ & \dot{x}_{\vec{n}} = \left(\sum_{i=1}^l n_i r_i - \sum_{i=1}^l (1 - n_i) \alpha_i \right) x_{\vec{n}} + \left(\sum_{\vec{\bar{n}} \leq \vec{n}, h(\vec{n}, \vec{\bar{n}})=1} x_{\vec{\bar{n}}} \sum_{i=1}^l \alpha_i (n_i - \bar{n}_i) \right), \\ & x_{\vec{n}}(0) = x_{\vec{n}}^0 \quad \forall \vec{n} \in \{0, 1\}^l. \end{aligned}$$

Bemerkung 3.2

Indem man den Vektor \vec{n} als binären Vektor $(\)_2$ betrachtet, wird auf die Zelltypen die Ordnung der natürlichen Zahlen induziert.

Anhand dieser Ordnung lässt sich das lineare Differentialgleichungssystem in Vektor-Matrix Schreibweise überführen.

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} \\ \vec{x}(0) &= \vec{x}^0.\end{aligned}$$

Hierbei gilt für $A := (a_{k,m})$

$$\begin{aligned}a_{k,k} &= \left(\sum_{i=1}^l n_i r_i - \sum_{i=1}^l n_i \alpha_i \right) \\ a_{k,m} &= 0, \quad k < m \\ a_{k,m} &= 0, \quad k > m, \quad \bar{\vec{n}} \geq \vec{n} \quad \text{oder} \quad h(\vec{n}, \bar{\vec{n}}) = 1 \\ a_{k,m} &= \sum_{i=1}^l \alpha_i (n_i - \bar{n}_i), \quad k > m, \quad \bar{\vec{n}} \leq \vec{n} \quad \text{und} \quad h(\vec{n}, \bar{\vec{n}}) = 1.\end{aligned}$$

Dies ist eine reguläre, untere Dreiecksmatrix und somit ist die Existenz und Eindeutigkeit einer Lösung gegeben.

3.1 Datenanalyse

Im vorhandenen Datensatz, siehe Anhang **Mutationen**, treten 48 verschiedene Mutationen auf. Für das vorgestellte Modell bedeutet dies, dass $2^{48} = 281474976710656$ verschiedene Zelltypen und damit Gleichungen mit 144 Parametern zu berücksichtigen sind. Eine solche Problemdimension ist mit den derzeitigen Computern nicht in vernünftiger Laufzeit zu bewältigen. Daher benötigt man eine Methode, um wichtigere von unwichtigeren Mutationen zu unterscheiden.

Hierfür eignen sich Scoring-Funktionen, die jeder Mutation anhand der Daten einen Wert zuweisen, der die Bedeutung der Mutation für die Dynamik widerspiegelt. Um Subjektivität zu reduzieren und die Auswahlkriterien transparenter zu machen, wird eine Scoring-Funktion aufgestellt, die folgenden Anforderungen Rechnung trägt:

1. Die relative Häufigkeit einer Mutation innerhalb eines Zeitschrittes sollte berücksichtigt werden.
2. Alle Zeitschritte sollten gleichgewichtet behandelt werden.

3. Mutationen, die an mehreren Zeitpunkten vorhanden sind, sollten höher gewichtet werden.

Da man mittels der Scoring-Funktion die Anzahl der berücksichtigten Mutationen vor der Parameterbestimmung festlegt, kann man die Veränderung der Reproduktionsrate nicht zur Bestimmung der „Überlebefähigkeit“ verwenden. Jedoch kann man davon ausgehen, dass eine Mutation, die über mehrere Zeitschritte im Datensatz erscheint, einen Vorteil zum Überleben und somit zum Reproduzieren besitzt.

Beginnend mit den nachstehenden Definitionen versucht man nun eine parameterabhängige Funktion aufzustellen, die diese Anforderungen erfüllt.

Definition 3.3

- $f(j, s)$ sei die relative Häufigkeit der Mutation j zum Zeitpunkt s :

$$f(j, s) = \frac{\sum_{i=1}^{m_s} (N_i^{t_s})_j}{m_s}.$$

- $C(j)$ sei die Gesamtzahl des Auftretens der Mutation j über alle Zeitpunkte:

$$C(j) = \sum_{s=1}^k \sum_{i=1}^{m_s} (N_i^{t_s})_j.$$

Definition 3.4

Der Wert der Scoring-Funktion $S_{\alpha,\beta}(j)$ für eine Mutation j berechnet sich aus:

$$S(j) := \alpha \max_{s=1..k} f(j, s) + \beta C(j).$$

Bemerkung 3.5

Die Parameter α und β ermöglichen eine Gewichtung der einzelnen Terme.

Abbildung 3.1 zeigt die Werte der Scoring-Funktion im oberen Teil für $\alpha = 1, \beta = 1$ und im unteren Teil für $\alpha = 1, \beta = 0.25$. Dabei wird $\beta C(j)$ positiv und $\alpha \max_{s=1..k} f(j, s)$ negativ auf der Y-Achse aufgetragen, so dass die Länge des Balkens gerade der Scoring-Funktion entspricht, jedoch die beiden Terme einzeln zu erkennen sind. Zur Vereinfachung werden die einzelnen Mutationen nach ihrer gleichgewichteten Ordnung (oben) von 1 bis 48 durchnummeriert (siehe auch Anhang **Mutationen**).

Betrachtet man sich den oberen Teil der Abbildung genauer, erkennt man, dass die Mutationen im Wesentlichen nach der Gesamtzahl ihres Auftretens $C(j)$ sortiert sind. Deutlich sind darüber hinaus zwei Abstufungen bei Mutation 4 und Mutation 36 zu erkennen. Diese zeigen an, dass die entsprechenden Mutationen nur in einem Zeitschritt gehäuft auftreten. Der Parameter β ermöglicht diese Mutationen stärker in der Scoring-Funktion zu berücksichtigen. Eine solche Umordnung, die durch eine

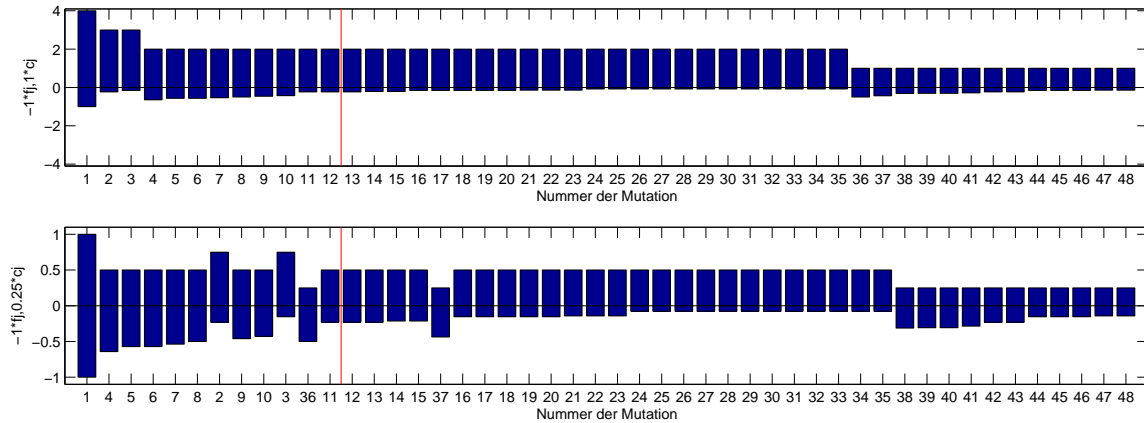


Abbildung 3.1: Die Ergebnisse der Scoring Funktion.

Veränderung der Gewichtung entsteht, erkennt man im unteren Teil der Abbildung. Bei der Einschränkung auf zum Beispiel 12 Mutationen - entspricht 36 Parametern und 4096 Differentialgleichungen - durch die Markierung hervorgehoben, erkennt man, dass lediglich eine Mutation (Nr. 12 durch Nr. 36) ersetzt werden muss. D.h., die Reihenfolge ist bezüglich der Gewichtung sehr robust.

Die Anwendung der Scoring-Methode liefert ein Ranking der 48 verschiedenen Mutationen. Durch die exponentiell wachsende Komplexität und der kleinen Größe des Datensatzes wurde das Modell für die 12 am höchsten gewerteten Mutationen aufgestellt. Diese sind in Tabelle 3.2 aufgeführt. Verwendet wurde hierbei nach Rücksprache mit den beteiligten Experimentalwissenschaftlern die obere Ordnung mit $\alpha = 1$ und $\beta = 1$.

3.2 Implementierung und Ergebnisse

3.2.1 Lösung der Differentialgleichung

Wendet man ein Optimierungsverfahren an, so muss man zur Auswertung der Likelihood-Funktion die Lösung der Differentialgleichung

$$\begin{aligned} \dot{x}_{\vec{n}} &= \left(\sum_{i=1}^l n_i r_i - \sum_{i=1}^l (1 - n_i) \alpha_i \right) x_{\vec{n}} + \left(\sum_{\vec{n} \leq \vec{n}, h(\vec{n}, \vec{n})=1} x_{\vec{n}} \sum_{i=1}^l \alpha_i (n_i - \bar{n}_i) \right), \\ x_{\vec{n}}(0) &= x_{\vec{n}}^0 \quad \forall \vec{n} \in \{0, 1\}^l \end{aligned}$$

kennen.

Eine Möglichkeit liegt in der Bestimmung der analytischen Lösung. Da es sich um

ein lineares Differentialgleichungssystem handelt, bedeutet dies als Aufgabe die Bestimmung der Eigenwerte und Eigenvektoren. Die Eigenwerte liegen sogar schon vor. Gegeben durch die untere Dreiecksgestalt der Matrix A finden sich die Eigenwerte auf der Diagonalen, außerdem ist dadurch auch die Existenz eines Eigenvektors zu jedem Eigenwert gegeben. Zur iterativen Bestimmung der Eigenvektoren wurde hierzu das Programm MAPLE verwendet.

Exemplarisch sind hier die Ergebnisse für zwei Mutationen angegeben:

$$A = \begin{pmatrix} -\alpha_1 - \alpha_2 & 0 & 0 & 0 \\ \alpha_1 & r_1 - \alpha_2 & 0 & 0 \\ \alpha_1 & 0 & r_2 - \alpha_2 & 0 \\ 0 & \alpha_2 & \alpha_1 & r_2 + r_1 \end{pmatrix}$$

mit den Eigenwerten $\{r_1 - \alpha_2, r_2 - \alpha_1, -\alpha_1 - \alpha_2, r_1 + r_2\}$ und den dazu gehörigen Eigenvektoren (spaltenweise)

$$\begin{bmatrix} 0 & 0 & \frac{(r_1 + \alpha_1)(r_2 + \alpha_2)}{\alpha_2 \alpha_1} & 0 \\ -\frac{r_2 + \alpha_2}{\alpha_2} & 0 & -\frac{r_2 + \alpha_2}{\alpha_2} & 0 \\ 0 & -\frac{r_1 + \alpha_1}{\alpha_1} & -\frac{r_1 + \alpha_1}{\alpha_1} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

In nachfolgender Tabelle 3.1 erkennt man das Problem dieses Ansatzes. Die Bestimmung der Eigenvektoren benötigt einen enormen Speicherbedarf und auch die Rechenzeit wächst exponentiell, was dazu führt, dass man mit einem Centrino 1,6 GHZ mit 1GB Hauptspeicher nur für kleine l , also für wenige Mutationen, überhaupt zu einem Ergebnis kommt. bei $l = 5$ meldet Maple sogar folgenden Fehler: „Maple was unable to allocate enough memory to complete this computation“. Maple scheint die besondere Struktur der Matrix nicht auszunutzen.

Anzahl der Mutationen	Rechenzeit Sekunden	Speicherbedarf Bytes
1	0.04	536990
2	0.05	1548102
3	0.27	5889174
4	3145.70	5483433114

Tabelle 3.1: Rechenzeit und Speicherbedarf in Abhängigkeit der Anzahl der berücksichtigten Mutationen

Eine weitere Möglichkeit der analytischen Lösung liegt in der Ausnutzung der speziellen Struktur der Matrix A . Die untere Dreiecksstruktur ermöglicht nämlich ein iteratives analytisches Lösen der Differentialgleichungen, indem man die schon bestimmten Lösungen y_1, \dots, y_{i-1} als inhomogenen Anteil der i -ten Differentialgleichung

$$y_i' = a_{ii}y_i + \vec{a}_i \vec{y}_i$$

mit $\vec{a}_i = (a_{i1}, \dots, a_{ii-1})$ und $\vec{y}_i = (y_1, \dots, y_{i-1})^T$ betrachtet. Mit Hilfe des folgenden Lemmas lässt sich die Lösung eines linearen Differentialgleichungssystems mit konstanter unterer Dreiecksmatrix A als Matrix-Vektor-Produkt einer konstanten unteren Dreiecksmatrix B schreiben.

Lemma 3.6

Ein lineares Differentialgleichungssystem mit konstanter unterer Dreiecksmatrix A

$$y' = Ay$$

hat die Lösung

$$y(x) = B\vec{e}$$

mit $\vec{e} = (e^{a_{11}x}, \dots, e^{a_{nn}x})^T$ und für $B = b_{ij}$ gilt

1 $b_{11} = y_{10}$

2 $b_{ij} = \frac{1}{a_{ii} - a_{ii}} \left(\sum_{l=1}^{i-1} a_{il} b_{lj} \right)$ für $1 \leq j < i$

3 $b_{ii} = y_{i0} - \sum_{j=1}^{i-1} b_{ij}$

4 $b_{ij} = 0$ für $j > i$.

Beweis:

Die i -te Zeile ist eine inhomogene, lineare Differentialgleichung 1. Ordnung, deren Lösung sich gerade zu

$$\begin{aligned} y_i &= \left(\int_0^x e^{-a_{ii}\tau} \vec{a}_i \vec{y}_i(\tau) d\tau + y_{i0} \right) e^{a_{ii}x} \\ &= \left(\int_0^x e^{-a_{ii}\tau} \sum_{j=1}^{i-1} a_{ij} y_j(\tau) d\tau + y_{i0} \right) e^{a_{ii}x} \end{aligned}$$

berechnen lässt. Bekannt ist weiter, dass sich die Lösungen y_j als Linearkombinationen der $e^{a_{ll}x}$ schreiben lassen:

$$y_j(\tau) = \sum_{l=1}^j b_{jl} e^{a_{ll}\tau}.$$

Setzt man dieses ein und führt einige Umformungen durch, so erhält man:

$$\begin{aligned}
y_i &= \left(\int_0^x e^{-a_{ii}\tau} \sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} e^{a_{ll}\tau} \right) d\tau + y_{i0} \right) e^{a_{ii}x} \\
&= \left(\sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \int_0^x e^{(a_{ll}-a_{ii})\tau} d\tau \right) + y_{i0} \right) e^{a_{ii}x} \\
&= \left(\sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \frac{1}{a_{ll} - a_{ii}} (e^{(a_{ll}-a_{ii})x} - 1) \right) + y_{i0} \right) e^{a_{ii}x} \\
&= \sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \frac{1}{a_{ll} - a_{ii}} (e^{a_{ll}x} - e^{a_{ii}x}) \right) + y_{i0} e^{a_{ii}x} \\
&= \underbrace{\left(y_{i0} - \sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \frac{1}{a_{ll} - a_{ii}} \right) \right)}_{Term1} e^{a_{ii}x} + \underbrace{\sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \frac{1}{a_{ll} - a_{ii}} e^{a_{ll}x} \right)}_{Term2}.
\end{aligned}$$

Betrachtet man sich *Term2*, so erhält man eine Bedingung für die b_{ij} mit $j < i$, welche nur von den a_{ij} und b_{jl} abhängt, wobei $j, l < i$

$$\begin{aligned}
\sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \frac{1}{a_{ll} - a_{ii}} e^{a_{ll}x} \right) &= \sum_{j=1}^{i-1} \left(\sum_{l=1}^j \frac{a_{ij} b_{jl}}{a_{ll} - a_{ii}} e^{a_{ll}x} \right) \\
&= \sum_{l=1}^{i-1} \left(\sum_{j=1}^{i-1} \frac{a_{ij} b_{jl}}{a_{ll} - a_{ii}} e^{a_{ll}x} \right) \\
&= \sum_{l=1}^{i-1} \frac{1}{a_{ll} - a_{ii}} \underbrace{\left(\sum_{j=1}^{i-1} a_{ij} b_{jl} \right)}_{b_{il}} e^{a_{ll}x}.
\end{aligned}$$

Analog erhält man durch *Term1* eine Bedingung für b_{ii} :

$$\begin{aligned}
\sum_{j=1}^{i-1} a_{ij} \left(\sum_{l=1}^j b_{jl} \frac{1}{a_{ll} - a_{ii}} \right) &= \sum_{j=1}^{i-1} \left(\sum_{l=1}^j \frac{a_{ij} b_{jl}}{a_{ll} - a_{ii}} \right) = \sum_{l=1}^{i-1} \left(\sum_{j=1}^{i-1} \frac{a_{ij} b_{jl}}{a_{ll} - a_{ii}} \right) \\
&= \sum_{l=1}^{i-1} \frac{1}{a_{ll} - a_{ii}} \underbrace{\left(\sum_{j=1}^{i-1} a_{ij} b_{jl} \right)}_{b_{il}}.
\end{aligned}$$

Zusammengefasst ergibt sich:

$$\mathbf{A1} \quad b_{11} = y_{10}$$

$$\mathbf{A2} \quad b_{ij} = \frac{1}{a_{il} - a_{ii}} \left(\sum_{j=1}^{i-1} a_{ij} b_{jl} \right) \text{ für } 1 \leq l < i$$

$$\mathbf{A3} \quad b_{ii} = y_{i0} - \sum_{j=1}^{i-1} b_{ij}$$

◇

Dieses lässt sich auf anderem Weg durch Einsetzen in die Differentialgleichung überprüfen.

Mit $A = \{a_{ij}\}$, $B = \{b_{ij}\}$ und $\vec{e} = (e^{a_{11}x}, \dots, e^{a_{nn}x})^T$, $\vec{1} = (1, \dots, 1)^T$ gilt

$$\vec{y}' = A\vec{y}, \quad \vec{y} = B\vec{e} \quad \text{und} \quad \vec{y}_0 = B\vec{1}$$

und es lässt sich folgern:

$$B \begin{pmatrix} a_{11}e^{a_{11}x} \\ \vdots \\ a_{nn}e^{a_{nn}x} \end{pmatrix} = \underbrace{AB}_C \begin{pmatrix} a_{11}e^{a_{11}x} \\ \vdots \\ a_{nn}e^{a_{nn}x} \end{pmatrix}.$$

$$C = \begin{pmatrix} a_{11} & & 0 \\ \vdots & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & & 0 \\ \vdots & \ddots & \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}$$

$$\Rightarrow c_{kl} = (a_{k1}, \dots, a_{kk}, 0, \dots, 0) \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b_{ll} \\ \vdots \\ b_{nl} \end{pmatrix} = \sum_{p=l}^k a_{kp} b_{pl}.$$

Hiermit ergibt sich für die k -te Zeile

$$\sum_{p=1}^k a_{pp} b_{kp} e^{a_{pp}x} = \sum_{p=1}^k c_{kp} e^{a_{pp}x}.$$

Durch Koeffizientenvergleich und mit $\vec{y}_0 = B\vec{1}$ erhält man folgende Bedingungen:

$$\mathbf{B1} \quad a_{pp} b_{kp} = \sum_{p=l}^k a_{kp} b_{pl}$$

$$\mathbf{B2} \quad y_{i0} = \sum_{j=i}^k b_{i,j},$$

die äquivalent zu den Bedingungen **A1-A3** sind.

Bemerkung 3.7

Somit erhält man eine iteratives Verfahren zur Bestimmung der Eigenvektoren der Matrix A, welche spaltenweise in der Matrix B wiederzufinden sind.

Bei der Implementierung ist auf eine speicheroptimierte Programmierung der Vektoren und Matrizen zu achten. Der Vergleich verschiedener Lösungswege findet sich am Ende dieses Abschnittes.

Um allen diesen Vorüberlegungen aus dem Weg zu gehen, besteht auch die Möglichkeit, das Differentialgleichungssystem direkt numerisch zu lösen. Hierbei muss man nur beachten, dass sich die Eigenwerte des Systems sowohl im Vorzeichen als auch in der Größe unterscheiden und es sich daher um ein leicht steifes System handelt. Hierfür wurde die Matlab-Funktion `ode23` verwendet. `ode23` ist eine explizite Runge-Kutta (2,3)-Implementierung, die sich gerade für leicht steife Probleme eignet. Der Vorteil liegt nun darin, dass keinerlei Vorarbeit benötigt wird und auch unabhängig von der Dimension, also der Anzahl der berücksichtigten Mutationen, dasselbe Verfahren verwendet werden kann. Darin liegt aber gerade auch der Nachteil. Es wird kein Vorabwissen verwendet, noch wird die Struktur der Matrix berücksichtigt. Z.B. könnte auch die Berechnung der für `ode23` nötigen *LR*-Zerlegung entfallen.

Es ist wünschenswert, das Differentialgleichungssystem numerisch zu lösen, dabei aber die Struktur der Matrix A zu berücksichtigen. Wie schon bei den analytischen Lösungswegen ermöglicht ein System mit unterer Dreiecksmatrix A

$$y' = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} y$$

das sukzessive Lösen der einzelnen Gleichungen, beginnend mit

$$y_1' = a_{11}y_1.$$

Diese Lösung verwendet man dann als inhomogenen Teil für die weiteren Gleichungen, also

$$y_i' = a_{ii}y_i + \underbrace{\vec{a}_i \vec{y}}_{\text{inhomogener Anteil}}, \quad i > 1$$

mit $\vec{a}_i = [a_{i1}, \dots, a_{i,i-1}]$ und den schon berechneten Lösungen \vec{y} .

In jeder Zeile ist also eine lineare inhomogene DGL 1. Ordnung zu lösen. Hierfür eignet sich z.B. ein explizites Euler-Verfahren, für welches die Schrittweite h noch bestimmt werden muss. Verwendet man für jede Zeile die gleiche Schrittweite, um sich spätere Interpolationen zu ersparen, so lässt sich die Schrittweite anhand der

Eigenwerte der Matrix A bestimmen. Hierfür wendet man an, dass zur Stabilität für alle $\lambda < 0$ gerade $|1 + h\lambda| < 1$ gelten muss. Somit erhält man also den folgenden Algorithmus:

1. Sei $\vec{d} := \text{diag}(A)$, sortiere \vec{d}

$$\text{sort}(\vec{d}) = (\lambda_1, \dots, \lambda_k, 0, \lambda_{k+2}, \dots, \lambda_{l-1})$$

mit $\lambda_1 < \dots < \lambda_k < 0 < \lambda_{k+2} < \dots < \lambda_{l-1}$.

2. Setze $h = -1/\lambda_1$.

3. Bestimme $y_{1,j+1} = (1 + ha_{11})y_{1,j}$

4. Für $i = 2 : 2^l$

- a) Setze $\vec{a}_i = [a_{i1}, \dots, a_{ii-1}]$

- b) Setze $\vec{y}_{ij} = [y_{1j}, \dots, y_{i-1j}]$

- b) Bestimme $y_{i,j+1} = (1 + ha_{ii})y_{i,j} + h\vec{a}_i\vec{y}_{ij}$.

Bemerkung 3.8

Dieses Vorgehen lässt sich für beliebige Integrationsverfahren anpassen, hierbei muss jedoch die Schrittweite aus den jeweiligen Stabilitätsfunktionen bestimmt werden.

Ein Vergleich der drei Verfahren jeweils für eine Funktionsauswertung sieht man in Abbildung 3.2. Man erkennt, dass die iterativen Verfahren für wenige Mutationen durchaus konkurrenzfähig sind. Doch für größere Anzahlen wirkt sich die optimierte Speichernutzung und die Schrittweitensteuerung der `ode23` Implementierung aus. Es ist zu vermuten, dass mit Hilfe einer Hardwarenäheren Programmierung und einer optimierteren Speicherausnutzung beide iterativen Verfahren sinnvoll sein könnten. Da der Schwerpunkt dieser Arbeit im Modellieren und Simulieren unter Verwendung schon bestehender Implementierungen liegt, erfolgt kein Versuch einer verbesserten Programmierung und im Folgenden wird auf die Ergebnisse der `ode23` Implementierung Bezug genommen.

3.2.2 Optimierung

Damit die Ungleichungsrestriktionen eingehalten werden, transformiert man die Parameter:

$$a = \tilde{a}^2 \quad \text{and} \quad x_{\vec{n}i}^0 = \frac{(\tilde{x}_{\vec{n}i}^0)^2}{\sum_{\vec{m}i \in \{0,1\}^l} (\tilde{x}_{\vec{m}i}^0)^2}.$$

Die Minimierung erfolgt für die unrestringierten Parameter \tilde{a} und $\tilde{x}_{\vec{n}i}$.

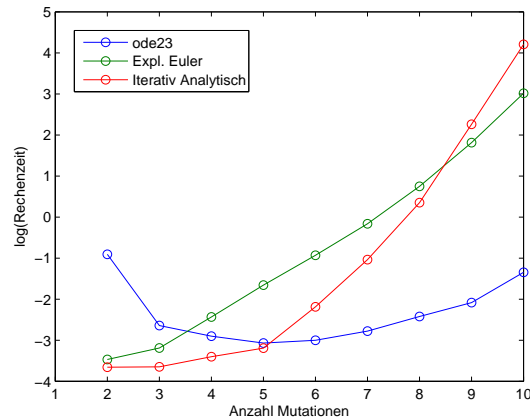


Abbildung 3.2: Die Rechenzeit in Abhängigkeit der Anzahl der berücksichtigten Mutationen.

Zur Lösung des Optimierungsproblems wurde die Matlab-Funktionen `fminsearch`, welches eine Implementierung des Nelder-Mead-Verfahrens ist, zu Hilfe gezogen. Die Ergebnisse sind in Tabelle 3.2 angegeben.

Nr.	Mutation	r_i	α_i	MutWS/Zteil.	Anfangswert
0.	keine	0	-	-	4.1133e-001
1.	der.2.t.2.3.	4.63e+000	2.13e-004	4.27e-006	1.29e-003
2.	del2p	-1.48e+000	6.61e-002	1.32e-003	5.87e-001
3.	del.3q.	-9.30e-001	8.01e-002	1.60e-003	1.16e-008
4.	der.17.t.X.17.t.X.8.	2.50e+000	2.43e-004	4.87e-006	5.20e-010
5.	del.13q.	8.12e-001	2.06e-002	4.13e-004	2.66e-009
6.	del.8q.	6.46e+000	1.78e-015	0	1.87e-007
7.	der.2.t.2.17.	1.97e+000	1.23e-003	2.47e-005	4.77e-010
8.	der.8.t.X.8.t.X.7.	5.29e+000	2.82e-018	0	8.73e-006
9.	der.X.t.X.15.	4.55e+000	8.46e-008	1.69e-009	1.13e-005
10.	der.10.t.X.10.	1.36e+000	6.51e-003	1.30e-004	3.11e-009
11.	der.6.t.3.6.	9.35e-001	6.62e-003	1.32e-004	1.03e-008
12.	der.X.t.X.13.	5.58e+000	2.02e-010	4.04e-012	1.95e-006

Tabelle 3.2: Reihenfolge der Mutationen und ihre identifizierten Parameter:

3.2.3 Visualisierung

Die Tabelle 3.2 gibt die bestimmten Parameter wieder. Um diese nun leichter zu interpretieren, soll das Verhalten der einzelnen Zelltypen oder auch nur einzelner Mutationen visualisiert werden. Noch gibt es keine standardisierten Ausgaben. Daher sollen hier zwei Möglichkeiten vorgestellt werden. Die erste besteht darin, die

marginalen Häufigkeiten (mH) zu betrachten. D.h., zu einem Zeitpunkt t_s betrachtet man die Häufigkeit einer einzelnen Mutation j ohne auf weitere Mutationen innerhalb dieser Zelle zu achten. Es werden also die relativen Häufigkeiten $y_{\vec{n}}(t_s)$ der Zellen \vec{n} summiert, welche die Mutation j , besitzen.

$$mH(j, s) = \sum_{\vec{n}, n_j=1} y_{\vec{n}}(t_s)$$

Diese Ausgabe lässt sich gut zur Überprüfung der Ergebnisse nutzen, da die marginalen Häufigkeiten einer Mutation j zum Zeitpunkt t_s des Modells gerade der relativen Häufigkeit bezüglich des Datensatzes entsprechen, welche schon für die Scoring Funktion durch $f(j, s)$ berechnet wurden.

Die zweite Ausgabe ermöglicht eine genauere Analyse der einzelnen Zelltypen und ihrer Entwicklung. Hierbei soll zu einem Zeitpunkt t_s die Dichte eines Zelltyps \vec{n} , repräsentiert durch seine relative Häufigkeit $y_{\vec{n}}(t_s)$ und der Übergang der Zelltypen untereinander, von einem Zeitpunkt zum nächsten abzulesen sein. Ersteres ist leicht zu erfüllen. Repräsentieren Kreise die Dichte eines Zelltyps zum Zeitpunkt t_s , so muss man nur den Radius proportional zur Dichte wählen. Den zweiten Punkt kann man mit Hilfe von Pfeilen darstellen, bei denen die Dicke proportional zur Menge der im Zeitraum von einem Zeitpunkt t_{s_1} bis t_{s_2} entstehenden Biomasse steht. Also entsprechen die Pfeile der Zunahme der Zellenhäufigkeit durch Mutation aus einem anderen Zelltyp bzw. durch Reproduktion des selbigen Zelltyps. Um die Zunahme der Zellenhäufigkeit (zH) zu bestimmen, ist der Anteil des Zelltyps x_j an der Zelle x_i beim Wechsel von Zeitpunkt t_{s_1} nach t_{s_2} gesucht.

Die Differentialgleichung $\dot{\vec{x}}(t) = A\vec{x}(t)$ hat die Lösung $\vec{x}(t) = e^{At}\vec{x}_0$ (1)

Daher lässt sich $\vec{x}(t_{s_2})$ aus $\vec{x}(t_{s_1})$ berechnen zu

$$\vec{x}(t_{s_2}) = e^{At_{s_2}}\vec{x}_0 = e^{A(\Delta t + t_{s_1})}\vec{x}_0 = e^{A\Delta t}\vec{x}(t_{s_1}).$$

Das Ergebnis ist dann:

$$zH(i, j) = (e^{A\Delta t})_{i,j} x(t_{s_1})_j.$$

3.3 Medizinische Interpretation der Ergebnisse

In Abbildung 3.3 sind die marginalen Häufigkeiten für das Modell und die Testdaten zu sehen. Die kontinuierlichen Kurven, errechnet vom Modell, passen sich den Datenpunkten an. Daraus lässt sich schließen, dass das Modell die Struktur der Daten gut wiedergibt. Aufgrund der Komplexität der zugrunde liegenden Prozesse kann dieses nicht im Vorneherein erwartet werden.

Im Datensatz ist zu erkennen, dass zum ersten Zeitpunkt der Messung Zellen ohne jegliche Mutation und Zellen mit der Mutation Nummer 2 (del2p) am häufigsten

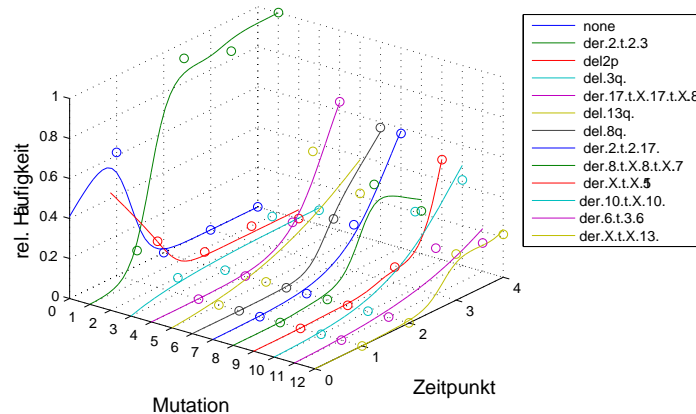


Abbildung 3.3: Vergleich der relativen Häufigkeit der Daten/Modell.

auftreten. Der Populationsverlauf dieser Zellen nimmt aber schnell ab und komplexere Mutationen treten auf. Dieser Effekt lässt sich detailliert in Abbildung 3.4 erkennen.

Bei der Abbildung 3.4 entsprechen die binären Codes entlang der Y-axis den unterschiedlichen Zellarten. Hierbei zeigt eine „1“ oder eine „0“ an, ob eine Mutation in dem Zelltyp vorhanden oder nicht vorhanden ist. Um die Abbildung nicht zu überhäufen, wurde sich auf die sechs höchstgesorten Mutationen beschränkt. Vergleichbar zu Abbildung 3.3 wurden die Häufigkeiten über die weniger wichtigen Mutationen aufsummiert. Es ist zu beachten, dass ein Zelltyp ohne Kreis in der Population vorhanden sein kann, aber eben mit einem sehr geringen Auftreten. Hier ist anzumerken, dass in diesem Modell keine Mutation wieder verloren gehen kann. Wegen der Kodierung der Zellen kann somit kein Pfeil abwärts gerichtet sein, was eben den Verlust einer Mutation bedeuten würde.

Man beachte, dass zum Vergleichen der vorherrschenden Mutationen in Abbildung 3.3 zur Abbildung 3.4 die Ergebnisse der verschiedenen Zelltypen einschließlich einer bestimmten Mutation aufzusummieren sind.

Die horizontalen Pfeile in der Abbildung sind deutlich stärker als die aufsteigenden. Wie durch das biologische System zu erwarten ist, trägt das Mutieren nicht entscheidend zur Zunahme der Biomasse eines Zelltyps bei. Die Reproduktionsrate einer solchen Zellenart entscheidet, ob sich die Zelle ausbreitet. Die Daten sowie das Modell lassen darauf schließen, dass das System zu Beginn relativ ausgeglichen ist, bis schließlich eine Zellart mit außerordentlicher Reproduktionsrate auftritt. Dieser Zelltyp überwiegt dann in einem sehr kurzen Zeitintervall.

Ein Blick auf Tabelle 3.2 lässt erkennen, dass sich die Mutationsraten in zwei Klassen

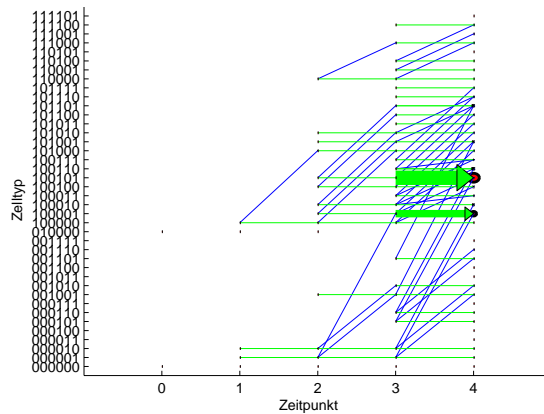


Abbildung 3.4: Mutationsentwicklung.

unterteilen lassen. Es gibt Mutationen mit hohen Raten (über 10^{-4}) und andere mit sehr geringen Raten (unter 10^{-8}). Aus der Abbildung 3.3 heraus ist ersichtlich, dass die Mutationen mit kleiner Mutationsrate erst später in der Population erscheinen, während jene mit hoher Rate die frühe Phase des Systems dominieren.

Mutation 2 (del2p) sticht dabei hervor; sie hat die zweithöchste Mutationsrate und somit durchgängig eine hohe Ausbreitung. Andererseits reduziert diese Mutation die Reproduktion einer Zelle drastisch (-1.48) und ist damit tödlich für eine Zelllinie. Dieses wird auch in Abbildung 3.4 deutlich. Zellen mit ausschließlich dieser einen Mutation (Kodierung 010000) treten zu den Zeitpunkten 0 und 1 auf. Eine Möglichkeit, mit dieser Mutation als Zelle zu überleben, ist folgende: Das Zusammentreffen der Mutationen mit positiver Reproduktionsrate gibt der Zelle einen Überlebensvorteil. Besonders das Zusammentreffen der Mutationen 6, 12, 8 und 1 (del.8q, der.X.t.X.13., der.8.t.X.8.t.X.7., der.2.t.2.3.) ergeben die höchste Reproduktionsrate. Tritt ein solcher Fall gleichzeitig mit einer zerstörerischen Mutation auf, so kann der Zelltyp dennoch weiter existieren. In dieser Simulation kann Mutation 2 zu späteren Zeitpunkten nur noch zusammen mit Mutation 1 gefunden werden (der.2.t.2.3.) (siehe Abbildung 3.4).

Insgesamt hat Mutation 6 die höchste Reproduktionsrate (6.46). Da die Mutationsrate von 6 sehr niedrig ist, erscheint sie zwar spät, dennoch herrscht sie zu späteren Zeitpunkten vor. Ein ähnliches Zeitverhalten ist auch bei der relativen Häufigkeit von Mutation 9 zu beobachten. Bei Mutation 12 scheint jedoch eine Art Sättigung einzutreten, was bedeutet, dass sich Zellen mit dieser Veränderung gemäß der durchschnittlichen Reproduktionsrate der Gesamtbevölkerung vermehren.

Zusammenfassend sei gesagt: Abbildung 3.3 zeigt die relative Häufigkeit der Mutationen über alle Zellen und unterscheidet hierbei nicht zwischen den Zelltypen.

Ein direkter Vergleich zwischen den unterschiedlichen Wachstumskurven ist folglich nicht ausreichend. So werden nur die qualitativen Resultaten hervorgehoben.

Langfristig setzten sich die Zelltypen durch, deren Mutationen die Reproduktionsrate erhöhen. Wenn aber einige Veränderungen sehr klein sind, geschehen diese erst zu sehr späten Zeitpunkten. Bis dahin können andere Effekte, wie eine vorübergehende Duldung, wichtiger sein. Als Konsequenz davon kann eine Mutation mit hoher Reproduktionsrate, allerdings mit sehr niedriger Mutationsrate, möglicherweise überhaupt nicht auftreten. Solche Effekte können selbstverständlich nicht durch dieses Modell vorausgesagt werden.

3.4 Zusammenfassung

In diesem Kapitel wurde ein Modell für die Entwicklung von Krebszellen vorgestellt, die durch M-FISH erkannt wurden. Das Modell ist in der Lage, die große Anzahl von Daten so zu strukturieren, dass diese Daten interpretierbar werden. Die Anwendung auf den gegebenen Datensatz zeigt, dass das Modell zum einen dazu in der Lage ist und dass, obwohl starke Vereinfachungen angenommen werden, die Hauptprozesse richtig reproduziert werden. Bei diesen Prozessen handelt es sich um Mutationen und Reproduktionen. Im Gegensatz zu den Daten gibt das Modell die Rolle dieser Prozesse explizit wieder: Durch Mutationen entstehen neue Zelllinien, die sich entweder reproduzieren oder aber aussterben, was deutlich zu erkennen ist. Der Vorteil des Modells liegt nun darin, explizite Raten für Mutation und Reproduktion zu liefern. So ist nachzuvollziehen, warum eine Zelle erst spät erscheint und dominiert oder aber weshalb eine frühere Zellenart ausstirbt. Das Modell kann als Werkzeug benutzt werden, um M-FISH Daten zu strukturieren und die Ergebnisse zu interpretieren. Jedoch liefert es nicht mehr Informationen als jene, die der Datensatz zur Verfügung stellt. Besondere Aufmerksamkeit benötigen kleine Datensätze, da deren Resultate sehr sorgfältig bearbeitet werden müssen.

Kapitel 4

Proteinspaltung durch das Enzym Proteasom

In diesem Kapitel wird gezeigt, dass man aus der „Proteinspaltung durch das Enzym Proteasom“ durch einen Binomialen Ansatz ein Optimierungsproblem unter Ungleichungsrestriktionen erhält. Anhand einer formellen Untersuchung kann dargelegt werden, dass sich die Modellfunktion vereinfachen und so der Rechnungsaufwand wesentlich reduzieren lässt. Analysiert man das vereinfachte Modell, wird ein direkter Zusammenhang der Singulärwerte sowohl mit den Eingabedaten als auch mit den gesuchten Parametern deutlich, der in einem Lemma bewiesen wird. Abschließend sollen erste Ergebnisse vorgestellt werden, darüber hinaus erfolgt ein Ausblick auf das weitere Vorgehen und das langfristige Ziel.

4.1 Modellanalyse

Zur Modellanalyse ist erst ein genauerer Blick auf die Herleitung des Modells zu werfen. Die beiden zugrunde liegenden Hauptprozesse, nämlich der eigentliche Proteinabbauprozess - bei dem das Protein in Fragmente gespalten wird - und der experimentelle Detektierungsprozess, nimmt man, da sie nacheinander ablaufen, als unabhängig an. Beiden weist man zwei Wahrscheinlichkeiten zu. Hierbei handelt es sich zum einen um die Wahrscheinlichkeiten „Spaltung“ oder „Keine Spaltung“ und zum anderen um „detektiert“ oder „nicht detektiert“. Diese Annahmen ermöglichen die Modellierung durch Binomialverteilungen.

Definition 4.1

- a) Sei die Trennungswahrscheinlichkeit in einem Protein nach der i -ten Aminosäure p_i mit $p_i \in I := \{p | 0 \leq p \leq 1\}$. $[i, j]$ stehe für das Proteinfragment, beginnend mit der i -ten Aminosäure des Proteins und endend mit der j -ten

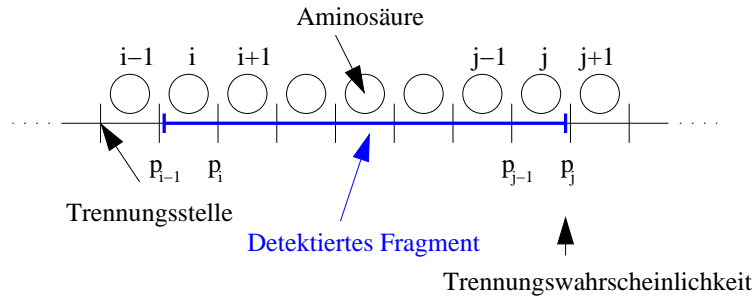


Abbildung 4.1: Proteinfragment das aus Aminosäuren besteht, sowie die zugehörigen Wahrscheinlichkeiten.

Aminosäure (Abbildung 4.1). Die Entstehungswahrscheinlichkeit eines Fragmentes $[i, j]$ sei dann bezeichnet durch p_{ij} .

- b) $[i, j] = \nu$ bedeute, dass das Fragment $[i, j]$ mit der Häufigkeit ν detektiert wurde.
- c) Mit $q(l, h)$ sei die Detektierwahrscheinlichkeit eines Fragmentes der Länge l und der Häufigkeit h gegeben.
- d) s sei die Länge des untersuchten Proteins und n sei die Anzahl der vorhandenen Proteine.

Dann ergibt sich die Entstehungswahrscheinlichkeit eines Fragmentes p_{ij} :

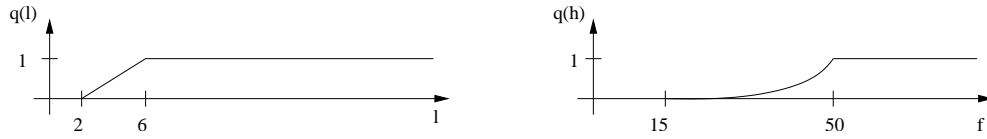
$$p_{ij} := p_{i-1} \prod_{k=i}^{j-1} (1 - p_k) p_j \quad i < j.$$

Die Detektionswahrscheinlichkeit wird benannt mit $q(l, h)$ und ist abhängig von der Länge l des Fragments und der Häufigkeit h . Um das Modell möglichst einfach zu halten, wird für die Detektionswahrscheinlichkeit $q(l, h)$ die Unabhängigkeit der längenabhängigen Wahrscheinlichkeit $q_l(l)$ und der häufigkeitsabhängigen Wahrscheinlichkeit $q_h(h)$ angenommen, so dass sich $q(l, h) = q_l(l)q_h(h)$ schreiben lässt. Die Funktion q_l (ebenso q_h) sei stetig, sie sei gleich Null unterhalb einer Länge l_1 (einer Häufigkeit h_1) und Eins oberhalb von l_2 (h_2). q_l sei dabei stückweise linear und q_h sei dabei das Quadrat einer stückweisen linearen Funktion (siehe Abbildung 4.2). Für die genauere Herleitung sei auf [MUE02] verwiesen.

Mit diesen Vorüberlegungen lässt sich die Wahrscheinlichkeit eines einzelnen Fragmentes berechnen.

Folgerung 4.2

Ein erwartungstreuer Schätzer für die Wahrscheinlichkeit P eines einzelnen Fragmentes $[i, j]$ mit der detektierten Häufigkeit ν berechnet sich aus der Summe aller

Abbildung 4.2: Die Funktionen $q_l(l)$ und $q_h(h)$.

Kombinationen, bei denen m Fragmente entstanden, aber nur ν detektiert wurden.

$$P([i, j] = \nu) = \sum_{m=\nu}^n \underbrace{\binom{m}{\nu} q(l, h)^\nu (1 - q(l, h))^{m-\nu}}_{\text{Detektionswahrscheinlichkeit}} \underbrace{\binom{n}{m} p_{ij}^m (1 - p_{ij})^{n-m}}_{\text{Entstehungswahrscheinlichkeit}}.$$

Der Ansatz durch Binomialverteilungen und der vorausgesetzten Unabhängigkeit erlaubt eine produktweise Aufteilung der beiden Wahrscheinlichkeiten.

Hierbei ergibt sich eine erste Möglichkeit, das Modell zu vereinfachen. Um die Wahrscheinlichkeit eines Fragmentes zu bestimmen, ist jedes Mal diese Summe zu bilden. Hierbei ist gewöhnlich die Größenordnung von n im Bereich von 4000 und ν von wenigen Hundert anzusetzen. Es sind also etwa 3000 Terme zu berechnen und das wiederum für jedes einzelne mögliche Fragment. Die Anzahl der möglichen Fragmente k berechnet sich zum Beispiel für das Protein Enolase mit 436 Aminosäuren zu:

$$k = \sum_{r=1}^{436} r = (1 + 436) \frac{436}{2} = 95266.$$

Für das Protein Prion, bestehend aus 260 Aminosäuren, erhält man 32896. Mit der abkürzenden Schreibweise

$$\text{bin}(\lambda, n, p) = \binom{n}{\lambda} p^\lambda (1 - p)^{n-\lambda}$$

lässt sich die Wahrscheinlichkeit eines Fragmentes schreiben mit

$$P([i, j] = \nu) = \sum_{m=\nu}^n \text{bin}(\nu, m, q(l, h)) \cdot \text{bin}(m, n, p_{i,j}).$$

Dieser Ausdruck kann mit Hilfe von Lemma 1 reduziert werden.

Lemma 4.3

Der erwartungstreue Schätzer für die Wahrscheinlichkeit eines Fragmentes $[i, j]$ mit detektierter Häufigkeit ν berechnet sich zu

$$P([i, j] = \nu) = \text{bin}(\nu, n, p_{i,j}) \cdot q(l, h)^\nu \cdot \left(\frac{1 - q(l, h) \cdot p_{i,j}}{1 - p_{i,j}} \right)^{(n-\nu)}. \quad (4.1)$$

Beweis:

$$\begin{aligned}
P([i, j] = \nu) &= \sum_{m=\nu}^n \text{bin}(\nu, m, q(l, h)) \text{bin}(m, n, p_{ij}) \\
&= \sum_{m=\nu}^n \binom{m}{\nu} q(l, h)^\nu (1 - q(l, h))^{m-\nu} \binom{n}{m} p_{ij}^m (1 - p_{ij})^{n-m} \\
&= p_{ij}^\nu q(l, h)^\nu \sum_{m=0}^{n-\nu} \binom{m+\nu}{\nu} \binom{n}{m+\nu} ((1 - q(l, h)) p_{ij})^m \\
&\quad (1 - p_{ij})^{(n-\nu)-m} \\
&= \binom{n}{\nu} p_{ij}^\nu q(l, h)^\nu \sum_{m=0}^{n-\nu} \binom{n-\nu}{m} ((1 - q(l, h)) p_{ij})^m \\
&\quad (1 - p_{ij})^{(n-\nu)-m} \\
&= \binom{n}{\nu} p_{ij}^\nu q(l, h)^\nu (1 - q(l, h) p_{ij})^{n-\nu} \\
&= \binom{n}{\nu} p_{ij}^\nu (1 - p_{ij})^{n-\nu} q(l, h)^\nu \left(\frac{1 - q(l, h) p_{ij}}{1 - p_{ij}} \right)^{n-\nu} \\
&= \text{bin}(\nu, n, p_{ij}) q(l, h)^\nu \left(\frac{1 - q(l, h) p_{ij}}{1 - p_{ij}} \right)^{n-\nu}.
\end{aligned}$$

◇

Daher müssen für jedes Fragment nicht mehr tausende Terme sondern nur noch ein einziger Term berechnet werden.

Bei der Modellfunktion müssen alle Fragmente gleichzeitig berücksichtigt werden. Dieses erreicht man mit einer Likelihood-Formulierung.

Definition 4.4

- a) Sei $D = (d_1, \dots, d_k)$ ein gegebener Datensatz mit $d_r = ([i, j] = \nu_r)$. Die Likelihood-Funktion eines solchen Datensatzes lässt sich mit $\vec{p} = (p_1, \dots, p_{s-1})^T$ definieren zu

$$M_D(\vec{p}) := \prod_{r=1}^k P([i_r, j_r] = \nu_r).$$

- b) *Optimierungsproblem*

Für eine gegebene Funktion $q(l, h)$ finde \hat{p} , welches $M_D(\vec{p})$ maximiert.

$$M_D(\hat{p}) = \max_p M_D(\vec{p}). \quad (*)$$

Oft betrachtet man anstelle von (*) das Minimierungsproblem über (siehe Kap. 2):

$$-\log(M_D(p)) = \sum_{r=1}^k -\log(P([i_r, j_r] = \nu_r)).$$

Definiert man sich

$$F_D(p) := \begin{pmatrix} -\log(P([i_1, j_1] = \nu_1)) \\ \dots \\ -\log(P([i_k, j_k] = \nu_k)) \end{pmatrix},$$

so lässt sich das Maximum-Likelihood-Problem ohne weiteres in ein äquivalentes kleinste-Quadrate-Problem umschreiben:

$$\min \|F_D(\vec{p})\|_2.$$

Und somit erhält man für einen Datensatz D :

Gegeben: Datensatz D

$$\min \|F_D(\vec{p})\|_2$$

$$0 \leq p_i \leq 1.$$

4.2 Datenanalyse

Das ursprüngliche Minimierungsproblem mit über 90000 Fragmenten zu betrachten ist nicht sinnvoll, da in den Experimenten nur 100 – 200 nachweisbare Fragmente aufgetreten sind. Bei Enolase sind es 135 und beim Prion 133. Außerdem finden sich in den Enolase-Daten (Prion) nur 114 (74) verschiedene mögliche Start- oder Endpunkte eines Fragments. Ein möglicher Weg, die Parameteranzahl zu verringern, liegt in der Nutzung der Singulärwertzerlegung der Jacobimatrix. Hierfür werden die relevanten Fragmente und Spaltwahrscheinlichkeiten über die Singulärwertzerlegung der Jacobimatrix von $F_D(\vec{p})$ identifiziert.

$$DF_D(\vec{p}) = \begin{pmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & \dots & * \end{pmatrix} \Leftrightarrow DF_D(\vec{p}) = U \cdot S \cdot V^T \quad (4.2)$$

Um die Singulärwertzerlegung zu berechnen, wird die Funktion `SVD` in MATLAB verwendet, welche mit dem LAPACK Routine `DGESVD` gleichwertig ist. Um Eigenschaften der Singulärwertzerlegung zu erkennen, wurde diese mit drei unterschiedlichen, typischen Sätzen von Anfangswerten für \vec{p} berechnet. Diese Anfangswerte seien bezeichnet durch *Initial1* bis *Initial3*.

- Initial1:

$$p_i \equiv 0, \forall i$$

- Initial2:

$$p_i \equiv 0.1, \forall i$$

- Initial3:

$$p_i = \begin{cases} 0 & i \text{ ist eine Spaltstelle} \\ 0.1 & i \text{ ist keine Spaltstelle} \end{cases}$$

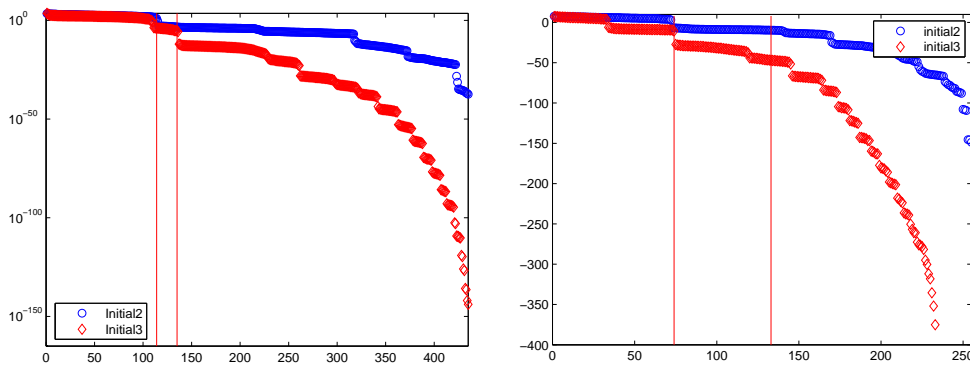


Abbildung 4.3: Singularwerte für verschiedene \vec{p} Werte (logarithmisch aufgetragen) für das Testbeispiel Enolas (links) und Prion (rechts).

In Abbildung 4.3 links und Tabelle 4.1 sind die Singularwerte (SV) für den Enolase-Datensatz angegeben. Bei den Anfangswerten *Initial2* und *Initial3*: 331 SV von 435 möglichen sind identisch 0, entsprechend der Anzahl der Nicht-Spaltungsstelle. Es ist ein Hinweis darauf, dass viele Spaltungswahrscheinlichkeiten überflüssig sind und viele Fragmente keine Information beitragen. Die SV von *Initial3* enthalten aber noch mehr Informationen. Die Anzahl der Singularwerte größer als $1.0e - 6$ (genaue Anzahl 135, erste deutliche Stufe) entspricht genau der Anzahl der detektierten Fragmente. Die mittleren SV (zweite Spalte der Tabelle) sind durch die Spaltungsstellen und die detektierten Fragmente beeinflusst. Dieses ist auch in Abbildung 4.3 links zu erkennen, gekennzeichnet durch die zwei Markierungen. Auch im rechten Teil der Abbildung kann man deutlich die erste Stufe, die der Anzahl der aufgetretenen Spaltungsstellen beim Prion entspricht, erkennen. Jedoch ist die zweite Stufe, welche der Anzahl der detektierten Fragmente entspricht, nicht zu erkennen.

Innere Extremwerte der Funktion $F_D(\vec{p})$ erkennt man an den Nullstellen der ersten Ableitung von $DF_D(\vec{p})$. Die SVD von DF_D ermöglicht somit eine Reduktion der Parameter. SV gleich Null oder nahe bei Null entsprechen den Termen ohne (oder mit) wenig Einfluss auf die Lösung. Lässt man dieses wegfallen, so reduziert sich die Komplexität des Modells erheblich.

p Value	$SV \geq 0.5$	$0.5 > SV > 1.0e - 6$	$1.0e - 6 \geq SV$
Initial 1	0	0	435
Initial 2	114	106	215
Initial 3	114	21	300

Tabelle 4.1: Die Verteilung der Singulärwerte für den Enolase Datensatz.

Lemma 4.5**Parameter Reduktion**

Für das r -te Fragment, gegeben durch $[i_r, j_r]$, und die Spaltungswahrscheinlichkeit p_k mit $i \leq k \leq j$ folgt

$$\frac{\partial F_r}{\partial p_k} = \begin{cases} > 0, & \text{wenn } k \text{ eine Spaltungsstelle ist} \\ < 0, & \text{wenn } k \text{ keine Spaltungsstelle ist.} \end{cases} \quad (4.3)$$

Beweis:

Zur Vereinfachung sei für $F_r = -\log(P([i_r, j_r] = \nu_r))$ $[i_r, j_r] = [i, j]$ und $\nu_r = \nu$.

Für die erste Ableitung von F_r mit

$$\frac{\partial F_r}{\partial p_{ij}} = -\frac{1}{P([i, j])} \frac{\partial P([i, j] = \nu)}{\partial p_{ij}}$$

erhält man

$$\frac{\partial P([i, j] = \nu)}{\partial p_{ij}} = \binom{n}{\nu} q(l, h)^\nu p_{ij}^{\nu-1} (1 - q(l, h)p_{ij})^{n-\nu-1} [\nu - nq(l, h)p_{ij}].$$

Dieser Term besitzt die Nullstellen

$$\text{erste: } p_{ij} = 0, \quad \text{zweite: } p_{ij} = \frac{\nu}{nq(l, h)}, \quad \text{dritte: } p_{ij} = \frac{1}{q(l, h)}.$$

Der zweite Kandidat p_{ij} ist eine Nullstelle, wenn $\frac{\nu}{n} \leq q(l, h)$ und der dritte Kandidat p_{ij} ist eine Nullstelle, wenn $q(l, h) = 1 \Rightarrow p_{ij} = 1$. Die zweite Ableitung ergibt sich zu

$$\frac{\partial^2 P([i, j] = \nu)}{\partial^2 p_{ij}} = \binom{n}{\nu} q(l, h)^\nu p_{ij}^{\nu-2} (1 - q(l, h)p_{ij})^{n-\nu-2} (\nu^2 - \nu + 2\nu q(l, h)p_{ij} - 2\nu nq(l, h)p_{ij} - nq(l, h)^2 p_{ij}^2 + n^2 q(l, h)^2 p_{ij}^2).$$

Setzt man die zweite mögliche Nullstelle $p_{ij} = \frac{\nu}{nq(l, h)}$ der ersten Ableitung ein, so erhält man

$$\frac{\partial^2 P([i, j] = \nu)}{\partial^2 p_{ij}} = \binom{n}{\nu} q(l, h)^2 \left(\frac{\nu}{n}\right)^{\nu-2} (1 - \frac{\nu}{n})^{n-\nu-2} (-\nu + \frac{\nu^2}{n}).$$

In diesem Term ist jeder Ausdruck außer dem letzten positiv. Daher

$$-\nu + \frac{\nu^2}{n} \leq 0 \quad \Rightarrow \quad \frac{\nu^2}{n} \leq \nu \quad \Rightarrow \quad \frac{\nu}{n} \leq 1.$$

Das bedeutet, dass $P([i, j] = \nu)$ seinen maximalen Wert $p_{ij} = \frac{\nu}{nq(l, h)}$ annimmt, wenn $\frac{\nu}{n} \leq q(l, h)$. Andernfalls für $\nu = 0$, $q(l, h) = 0$, oder für $\frac{\nu}{n} > q(l, h)$ $q(l, h) = 1$. So erklärt sich die Konvergenz der Nichtspaltpositionen gegen Null wegen

$$\frac{\partial F_l}{\partial p_k} = -\frac{1}{P([i, j] = \nu)} \frac{\partial P([i, j] = \nu)}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial p_k}$$

mit (angenommen $\nu = 0$)

$$\frac{\partial P([i, j] = \nu)}{\partial p_{ij}} = (1 - q(l, h)p_{ij})^{n-1}(-n)q(l, h)$$

und

$$\frac{\partial p_{ij}}{\partial p_k} = \begin{cases} i) & k = i - 1 & \frac{\partial p_{ij}}{\partial p_{i-1}} = \prod_{l=i}^{j-1} (1 - p_l)p_j \\ ii) & k = j & \frac{\partial p_{ij}}{\partial p_j} = p_{i-1} \prod_{l=i}^{j-1} (1 - p_l) \\ iii) & k \neq i, j & \frac{\partial p_{ij}}{\partial p_k} = -p_{i-1} \prod_{l=i}^{k-1} (1 - p_l) \prod_{l=k+1}^{j-1} (1 - p_l)p_j. \end{cases}$$

Zusammengenommen ergibt sich

$$\frac{\partial F_r}{\partial p_k} = \begin{cases} > 0, & \text{wenn } k \text{ eine Spaltungsstelle ist} \\ < 0, & \text{wenn } k \text{ keine Spaltungsstelle ist.} \end{cases}$$

und rechtfertigt folgende zwei Fälle:

- i oder j kommen in dem Datensatz nie als Anfangs- oder Endposition eines Fragmentes mit $\nu > 0 \Rightarrow p_{i-1} = 0, p_j = 0$ vor, daher gilt $p_{ij} = 0$. Daraus ergeben sich die folgenden Null-Einträge in der ersten Ableitung:

$$DF = \begin{pmatrix} * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \\ 0 & \cdots & 0 \\ * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \end{pmatrix}.$$

- Position k kommt in dem Datensatz nie als Spaltungsstelle vor. $\Rightarrow p_k = 0$. Daraus ergeben sich die folgenden Null-Einträge in der ersten Ableitung:

$$DF = \begin{pmatrix} * & * & 0 & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & 0 & * & * \end{pmatrix}.$$

◇

Bemerkung 4.6

Als Konsequenz dieser Ergebnisse ergibt sich die Struktur der ersten Ableitung wie unten stehend und somit der zugehörige Singulärwert gleich Null.

$$DF_D = \begin{pmatrix} * & \dots & * & 0 & * & \dots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \dots & * & & * & \dots & * \\ 0 & \dots & & 0 & & \dots & 0 \\ * & \dots & * & & * & \dots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \dots & * & 0 & * & \dots & * \end{pmatrix}.$$

Das bedeutet, dass für Fragmente $[i, j]$, wobei i und j nie als Spaltungsstelle im Datensatz auftreten, man zu zeigen hat: $P([i, j] = 0) = 1$. Aus beiden Fällen von oben folgt für $p_{i-1} = 0$ und $p_j = 0$, womit

$$\begin{aligned} \Rightarrow P([i, j] = 0) &= \sum_{m=0}^N \binom{m}{0} q(l, h)^0 (1 - q(l, h))^m \underbrace{\binom{N}{m} 0^m 1^{N-m}}_{=0 \text{ für } m \neq 0} \\ &= \binom{0}{0} q(l, h)^0 (1 - q(l, h))^0 \binom{N}{0} 0^0 1^N = 1 \end{aligned}$$

gilt. Das gleiche Argument trifft für Fragmente zu, bei denen nur eine der beiden Seiten nie eine Spaltungsstelle ist, wobei also nur p_{i-1} oder p_j gleich Null ist.

Diese Überlegungen reduzieren das Modell drastisch. Für die Optimierung müssen nur noch diejenigen Fragmente berücksichtigt werden, bei denen beide Seiten mögliche Spaltungsstellen sind. In dem Datensatz Enolase findet man 114 Spaltungsstellen und somit ergibt sich als Anzahl möglicher Fragmente k

$$k = \sum_{r=1}^{114} r = (1 + 114) \cdot \frac{114}{2} = 6555$$

und als Anzahl der nichttrivialen Parameter

$$\dim(\vec{p}) = 114.$$

Mit den vorangegangenen Definitionen erhält man die zwei Probleme:

1) Das nicht reduzierte Minimierungsproblem

$$\min_{\vec{p}} \|F_D(\vec{p})\|_2$$

mit

$$\vec{p} \in I^{435} \text{ und } F(\vec{p}) : I^{435} \rightarrow \mathbb{R}^{95266}.$$

2) Das reduzierte Minimierungsproblem gekennzeichnet durch \sim

$$\min_{\tilde{\vec{p}}} \left\| \tilde{F}(\tilde{\vec{p}}) \right\|_2$$

mit

$$\tilde{\vec{p}} \in I^{114} \text{ und } \tilde{F} : I^{114} \rightarrow \mathbb{R}^{6555}.$$

Die gesamte Arbeit zum Berechnen der ersten Ableitung DF_D wird reduziert von

$$435 \cdot 95266 = 41440710$$

zu

$$114 \cdot 6555 = 747270$$

Funktionsauswertungen.

4.3 Implementierung und Ergebnisse

Zum Lösen des Minimierungsproblems wurde die Funktion `lsqnonlin` der Optimierungstoolbox von MATLAB verwendet, welche eine Implementierung des Levenberg-Marquardt-Verfahrens ist. Hierfür wurde \vec{p} wie folgt transformiert, um ein unrestringiertes Problem zu erhalten:

$$\vec{p}_i = \frac{1}{1 + \hat{p}_i^2}, \quad \hat{p}_i \in \mathbb{R} \quad \Rightarrow \quad \vec{p}_i \in I.$$

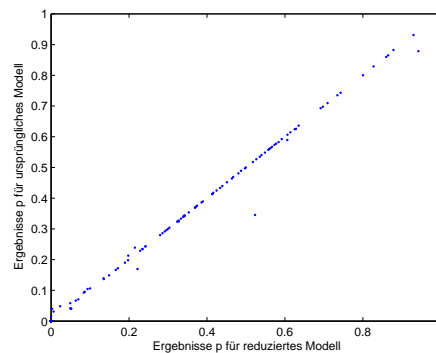


Abbildung 4.4: Vergleich der Parameter, berechnet mit Hilfe des ursprünglichen und des reduzierten Modells.

Die Ergebnisse des ursprünglichen und des reduzierten Modells sind in Abbildung 4.4 mit $(\vec{p}_{alt}, \vec{p}_{neu})$ geplottet, wobei sich nur sieben Parameter signifikant unterscheiden.

Eine genauere Datenanalyse zeigt, dass für diese Parameter sehr wenige Informationen zu Verfügung stehen, d.h. für eine solche Spaltungsstelle ist nur ein einzelnes Fragment mit geringer Häufigkeit gefunden worden. Betrachtet man dahingegen die Einsparung an Rechenzeit, so benötigte ein 1,6 GHz Centrino mehrere Tage für das ursprüngliche Problem, wobei das reduzierte in wenigen Stunden konvergiert.

Das vorgestellte Vorgehen lässt sich nun zu einer bestimmten Methode zusammenfassen, wobei der letzte Schritt dem ursprünglichen Vorgehen entspricht und die Beschleunigung durch die Dimensionsreduzierung erreicht wird.

Es sei gegeben:

Datensatz: Satz von N verschiedenen Dateneinträgen mit identischer Struktur

$$data_set = \{d_r, r = 1, \dots, N\}.$$

Verschiedene Anfangswerte: $\vec{p}_{init,j}$.

Modellfunktion: Funktion, abhängig von m Parametern und einem Datensatzeintrag d_r ,

$$F_r(\vec{p}) := F_{d_r}(\vec{p}) : I^m \rightarrow \mathbb{R}$$

$$F(\vec{p}) := (F_1(\vec{p}), \dots, F_N(\vec{p}))^T.$$

Minimierungsproblem:

$$\min_p \|F(\vec{p})\|_2.$$

Singulärwerte-basierte Methode

1. Modellreduzierung (Lemma 4.3): Vereinfache $F_r(\vec{p})$ (z.B. durch Nutzung der Eigenschaften der Binomialverteilung) $\Rightarrow \tilde{F}_r(\vec{p})$.

2. Parameterreduzierung (Lemma 4.5):

(a) Berechne die Singulären Werte für verschiedene Anfangswerte $p_{init,j}$ von $DF(\vec{p}_{init,j})$

$$DF(\vec{p}_{init,j}) = U_{init,j} \cdot S_{init,j} \cdot V_{init,j}^T.$$

(b) Analysiere die Singulärwerte.

(c) Identifiziere mit Hilfe von $U_{init,j}$ and $V_{init,j}$ unnötige Datensatzeinträge und Parameter $\Rightarrow \tilde{\vec{p}}$ and $\tilde{F}(\tilde{\vec{p}})$ (Abbildung 4.5).

3. Parameter Identifikation: Löse das Minimierungsproblem

$$\min_{\tilde{\vec{p}}} \left\| \tilde{F}(\tilde{\vec{p}}) \right\|_2.$$

Als ein letztes Ergebnis sieht man in Abbildung 4.5 die Einschnitte an den Positionen 114, 135 und 435, vorgegeben durch die SVD Einträge.

$$u_{i,j} = \text{Anteil vom } i\text{-ten Fragment am } j\text{-ten SV}$$

$$v_{i,j} = \text{Anteil vom } i\text{-ten Parameter am } j\text{-ten SV}$$

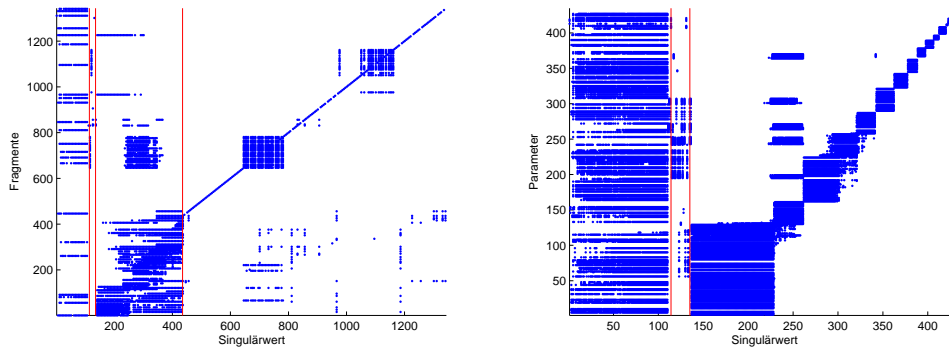


Abbildung 4.5: U (Auszug), V für den Beispieldatensatz Enolase und den Anfangswertsatz *Initial3*, geplottet, wenn $abs(U(i, j)), abs(V(i, j)) > 0.0001$, Markierungen an den Positionen 114, 135 und 435.

Diese Abbildung ermöglicht offensichtlich die Identifikation der essentiellen Parameter und Fragmente - also den Datensatzeinträgen.

4.4 Ausblick

Das langfristige Ziel ist nicht die Bestimmung der Schnittwahrscheinlichkeiten aufgrund eines vorhandenen Datensatzes, sondern die Vorhersage der Schnittwahrscheinlichkeiten für ein vorgegebenes Protein. Die bisher durchgeführte Untersuchung ist somit nur ein erster Schritt. In einem zweiten Schritt muss nun der Wert der Schnittwahrscheinlichkeit in Abhängigkeit ihrer Position innerhalb des Proteins erklärt werden. Hierfür sind einige Vorüberlegungen nötig.

Die natürlich vorkommenden 20 Aminosäuren haben Namen wie Alanin, Cysteine... , welche durch einen Dreibuchstaben-Code bezeichnet werden: Ala, Cys... (Dieser Code darf nicht mit den Dreiergruppen des Genetischen Codes verwechselt werden). Der Dreibuchstaben-Code ist für die computergestützte Berechnung ungünstig und wird daher durch einen Einbuchstaben-Code ersetzt. Folglich lassen sich die Aminosäuren schreiben als A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y. Der Anfang des Proteins wird nun benannt durch den N-N-Terminus und das Endstückende durch den C-C-Terminus. Weiter lassen sich sie Positionen entlang

des Proteins beginnend vom N-N-Terminus durchnummerieren. Somit ergibt sich, dass die Proteine „AC“ und „CA“ als verschieden angesehen werden. Das Auftreten der verschiedenen Aminosäuren in der Natur ist bei weitem nicht gleichverteilt. Die lineare Struktur des Proteins Enolase mit 436 Aminosäuren sieht wie folgt aus:

```
AVSKVYARSVYDSRGNPTVEVELTTEKGVFRSIVPSGASTGVHEALEMRD
GDKSKWMGKGVLHAVKNVNDVIAPAFVKANIDVKDQKAVDDFLISLDGT
ANKSKLGANAILGVSLAASRAAAAENKVPYKHLADLSKSKTSPYVLPVP
FLNVLNGGSHAGGALALQEFMIAPTGAKTFAEALRIGSEVYHNLKSLTKK
RYGASAGNVGDEGGVAPNIQTAEALDLIVDAIKAAGHDGKVKIGLDCAS
SEFFKDGKYDLDFKNPNSDKSKWLTGPQLADLYHSLMKRYPIVSIEDPFA
EDDWEAWSHFFKTAGIQIVADDLTVTNPKRIATAIEKKAADALLKVNQI
GTLSESIKAAQDSFAAGWGMVSHRSGETEDTFIADLVVGLRTGQIKTGA
PARSERLAKLNQLLRIEEEELGDNAVFAGENFHHGDKL
```

Der Spaltungsprozess eines Proteinsubstrats lässt sich also mit der Trennung von Wörtern vergleichen. Bei beiden fällt auf, dass die Trennung stark von den Aminosäuren bzw. Buchstaben, die sich in der Nähe der erwarteten Stelle befinden, abhängt. Der Trennungsprozess ist aber weder streng vorherbestimmt noch rein zufällig. Dennoch hängt die Schnittwahrscheinlichkeit in hohem Maße an den benachbarten Aminosäuren. Folglich ist es wichtig, eine Notation für die Position einer Aminosäuren in der Nachbarschaft (ein Fenster) relativ zur Schnittposition zu haben. Diese Positionen werden durch die Symbole B_i für Positionen nach links (in Richtung zum N-N-Terminus) und von B'_i für Positionen nach rechts derart bezeichnet, dass der Index i von der Schnittposition weg läuft (siehe dazu auch [HAD04]). So ergeben sich die Positionen im Verhältnis zu einer Schnittposition zu

$$B_k, \dots, B_2, B_1 | B'_1, B'_2, \dots, B'_m.$$

Hierbei geben k und m die Breite des berücksichtigten Fensters B an.

Gesucht sind nun Affinitätsparameter AP , die angeben, wie stark die Wahrscheinlichkeit durch eine Aminosäure Θ an einer festen Position B verstärkt oder geschwächt wird. Mit diesen ließe sich z.B. die Wahrscheinlichkeit eines Schnittes an Position i , also nach der i -ten Aminosäure, berechnen zu

$$p_i = \sum_{j=1}^k AP(\Theta_{i+1-j}, B_j) + \sum_{j=1}^m AP(\Theta_{i+j}, B'_j).$$

Zu bestimmen sind also $(k + m)20$ Parameter. In den bisherigen Ansätzen, wie z.B. bei Paproc (siehe [KUT00]), werden Neuronale Netze verwendet, um einen möglichen Schnitt vorherzusagen, ohne jedoch die Wahrscheinlichkeit anzugeben. Die hier vorgestellte Vorgehensweise würde eine begründete Berechnung dieser ermöglichen.

Leider werden hierfür deutlich mehr Datensätze benötigt, als zur Zeit zur Verfügung stehen. Dieses müsste sich aber in den nächsten Jahren im Rahmen immer schnellerer und kostengünstigerer Analyseverfahren ändern. In diesem Kapitel wurde aber schon die Anwendbarkeit gezeigt.

Abbildung 4.6 zeigt für $k = 6$ und $m = 6$ zum einen die Summierung des Auftretens in pmol und zum anderen die einfache Summierung der Wahrscheinlichkeiten einer Aminosäure Θ an Position B .

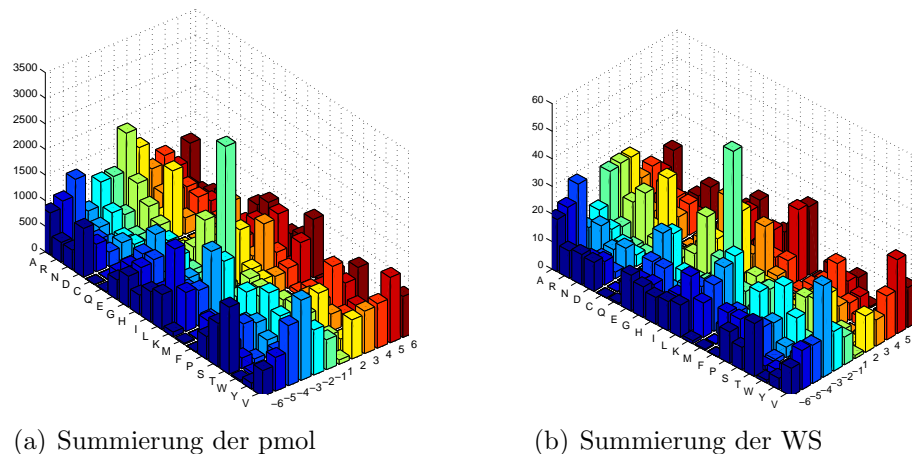


Abbildung 4.6: Vergleich der berechneten cut-Wahrscheinlichkeiten mit den detektierten Fragmentmengen in Abhängigkeit der Stelle der Aminosäuren bzgl. der Schnittstelle.

Der zu erkennende Zusammenhang bekräftigt noch einmal die Vermutung, mit Hilfe der identifizierten Wahrscheinlichkeiten eine Bestimmung der Affinitätsparameter zu ermöglichen.

Kapitel 5

Immunisierungsstrategie gegen Hepatitis in Krankenhäusern

In diesem Kapitel soll das aus einem Normalverteilungsansatz gewonnene Optimierungsproblem mit Ungleichungsrestriktionen gelöst werden. Die Ergebnisse werden anschließend einer Hauptkomponentenanalyse unterzogen, deren Resultate darauf hindeuten, dass es möglich ist, das Modell zu vereinfachen. Ob diese Vermutung zutrifft, müsste durch weiterführende Untersuchungen geklärt werden.

5.1 Modellanalyse

In Kapitel 1 wurde behauptet, dass sich der Antikörpergehalt nach einer Zufallsvariable Z_t verhält, die folgenden Zusammenhang erfüllt:

$$Z_t \sim N(\mu, \sigma^2) \Rightarrow Z_{t+\tau} \sim N(\mu - a\tau, \sigma^2 + b\tau).$$

Ziel ist es nun, diese Zufallswahrscheinlichkeit mit einer vorgegebenen Wahrscheinlichkeit γ über z.B. Null zu halten.

Lemma 5.1

Für

$$Z_t \sim N(\mu, \sigma^2) \text{ mit } (\mu, \sigma^2) \in \mathbb{R} \times \mathbb{R}_+$$

existiert eine Konstante $k_\gamma > 0$, so dass $P(Z < 0) \leq \gamma$ gilt.

Beweis:

$$P(Z_t < 0) \leq \gamma \Leftrightarrow P\left(\frac{Z - \mu}{\sigma} \leq -\frac{\mu}{\sigma}\right) \leq \gamma \Leftrightarrow -\frac{\mu}{\sigma} \leq -k_\gamma \Leftrightarrow \mu^2 \geq k_\gamma^2 \sigma^2.$$

Durch dieses Lemma werden folgende Definitionen sinnvoll.

Definition 5.2

Zu einer gegebenen Wahrscheinlichkeit γ sei die „Erlaubte Region“ Ω definiert als

$$\Omega := \{(\mu, \sigma^2) | \mu^2 > k_\gamma^2 \sigma^2\} \quad \text{mit} \quad P(Z \leq 0) \leq \gamma.$$

Der Rand $\partial\Omega := \Gamma^+ \cup \Gamma^-$ sei definiert als

$$\Gamma^+ := \{(\mu, \sigma^2) | \mu^2 = k_\gamma^2 \sigma^2\}$$

und

$$\Gamma^- := \{(\mu, \sigma^2) | \sigma^2 = 0\}.$$

Die erlaubte Region mit ihren Rändern ist in Abbildung 5.1 (a) schematisch zu erkennen.

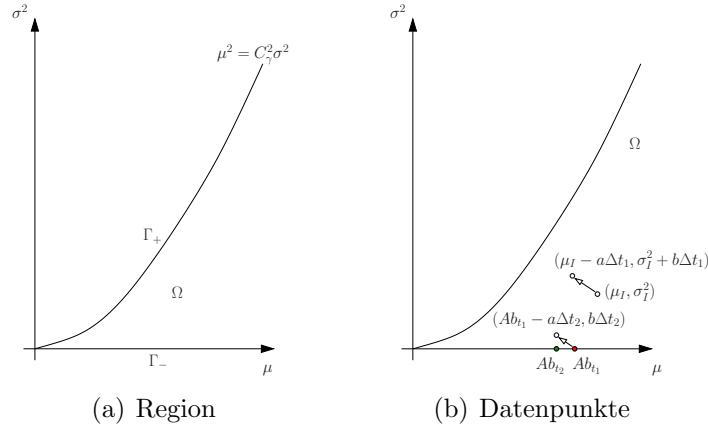


Abbildung 5.1: Erlaubte Region und der Verlauf zweier Datenpunkte in der (μ, σ^2) -Ebene.

In einem ersten Schritt müssen nun für jeden Patienten vier Parameter bestimmt werden, einerseits die Steigungen a und b und andererseits der ausgezeichnete Punkt (μ_I, σ_I^2) , der der Verteilung nach einer Impfung entspricht (siehe Abbildung 5.1 (b)). Hierfür liegen zwei verschiedene Datentypen vor, die sich wie folgt definieren:

Definition 5.3

d_1 sei die Menge aller Messpunkte, die direkt nach einer Impfung ermittelt wurden, d.h. sie bestehen aus der Zeit Δt_i seit der letzten Impfung und dem gemessenen Antikörperwert $Ab_{\Delta t_i}$.

d_2 sei die Menge aller Messpunkte, die aus einer wiederholten Kontrolle stammen, d.h. sie bestehen aus dem Antikörperwert $Ab_{\Delta t_{j-1}}$ der letzten Kontrolle, der Zeit Δt_j seit der letzten Kontrolle und dem gemessenen Antikörperwert $Ab_{\Delta t_j}$ aus der aktuellen Kontrolle.

Für die zwei Arten von Messpunkten gelten die Zusammenhänge zum einen für d_1

$$Ab_{\Delta t_i} \sim N(\mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i)$$

und zum anderen für d_2

$$Ab_{\Delta t_j} \sim N(Ab_{\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j).$$

Mit diesen lässt sich die Likelihood-Funktion wie folgt aufstellen:

Definition 5.4

Die Likelihood-Funktion eines Patientendatensatzes wird definiert durch:

$$L_D(\mu_I, \sigma_I, a, b) = \prod_{i=1}^{n_1} \pi(Ab_{1,\Delta t_i}; \mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i) \prod_{j=1}^{n_2} \pi(Ab_{2,\Delta t_j}; Ab_{2,\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j).$$

Nach üblicher Vorgehensweise ergibt sich damit die Log-Likelihood-Funktion zu:

Definition 5.5

Die Log-Likelihood-Funktion eines Patientendatensatzes wird definiert durch:

$$F_D(\mu_I, \sigma_I, a, b) := - \sum_{i=1}^{n_1} \log(\pi(Ab_{1,\Delta t_i}; \mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i)) - \sum_{j=1}^{n_2} \log(\pi(Ab_{2,\Delta t_j}; Ab_{2,\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j)).$$

Somit ergibt sich für jeden einzelnen Patienten folgende Optimierungsaufgabe:

Gegeben: Datensatz D

$$\min F_D(\mu_I, \sigma_I, a, b)$$

$$\mu_I \in \mathbb{R}, \sigma_I \geq 0$$

$$a \geq 0, b \geq 0$$

5.2 Datenanalyse

Bei jedem Patienten gibt es für jeden Datentyp mindestens einen Datenpunkt. Bei der genaueren Betrachtung des Datenmaterials lassen sich anhand der Anzahl der Datenpunkt n_i des Datentyps d_i vier Patiententypen unterscheiden. In Abbildung 5.2 sind diese durch vier typische Beispiele gezeigt:

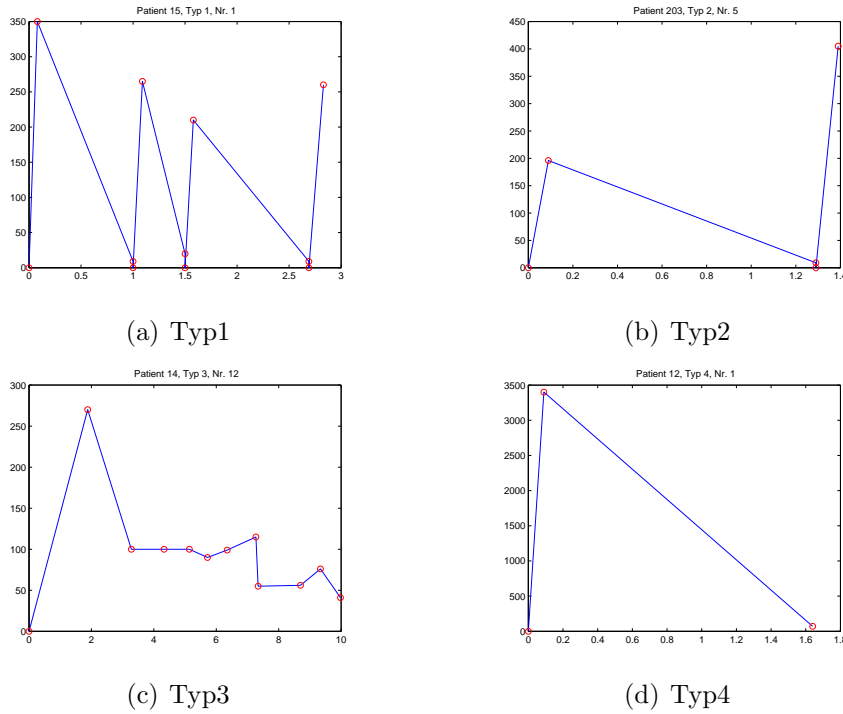


Abbildung 5.2: Typklassifizierung der Datensätze.

Typ1: $n_1 > 1$ und $n_2 > 1$

Typ2: $n_1 > 1$ und $n_2 = 1$

Typ3: $n_1 = 1$ und $n_1 > 1$

Typ4: $n_1 = 1$ und $n_1 = 1$

Anhand dieser sollen die vier Parameter μ_I , σ_I , a , b bestimmt werden. Daher ist es, um kein unterbestimmtes System zu erhalten, sinnvoll, $n_1 + n_2 \geq 4$ zu verlangen. Berücksichtigt man den Zusammenhang

$$Ab_{\Delta t_i} \sim N(\mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i)$$

$$Ab_{\Delta t_j} \sim N(Ab_{\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j),$$

so erkennt man, dass zur Bestimmung der μ_I , σ_I nur Datenpunkte des Datentyps 1 verwendet werden können. Somit benötigt man weiter $n_1 > 1$. Für die Parameter a und b können beide Datentypen verwendet werden, jedoch können mit $n_2 > 1$ diese Parameter unabhängig von μ_I , σ_I bestimmt werden. Daher wird sich im Folgenden nur auf Patienten des Typs 1 bezogen.

5.3 Implementierung und Ergebnisse

Ein Optimierungsproblem ist bezüglich der Wahl der Startwerte meist sensibel. Daher ist es sinnvoll, diese Startwerte vorher zu optimieren. Hierfür lassen sich die empirischen Schätzer für den Erwartungswert (Mittelwert)

$$\mu \approx \frac{1}{n} \sum_{i=1}^n x_i$$

und für die Varianz

$$\sigma^2 \approx \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

verwenden. Im Falle eines Patienten von Typ1 gilt $n_1, n_2 > 1$.

Zuerst berücksichtigt man die Datenpunkte d_2 mit

$$Ab_{\Delta t_j} \sim N(Ab_{\Delta t_{j-1}} - a\Delta t_j, b\Delta t_j) \Rightarrow \frac{Ab_{\Delta t_{j-1}} - Ab_{\Delta t_j}}{\Delta t_j} \sim N(a, b\Delta t_j).$$

Somit lässt sich a_0 als Erwartungswert schätzen zu

$$a_0 = \frac{1}{n_2} \sum_{j=1}^{n_2} \frac{Ab_{\Delta t_{j-1}} - Ab_{\Delta t_j}}{\Delta t_j}.$$

Mit diesem als richtig angenommen a_0 verwendet man für b den Zusammenhang:

$$\frac{Ab_{\Delta t_j} - (Ab_{\Delta t_{j-1}} - a_0\Delta t_j)}{\sqrt{\Delta t_j}} \sim N(0, b).$$

b_0 kann man als Varianz schätzen zu

$$b_0 = \frac{1}{1 - n_2} \sum_{j=1}^{n_2} \frac{(Ab_{\Delta t_j} - (Ab_{\Delta t_{j-1}} - a_0\Delta t_j))^2}{\Delta t_j}.$$

Zur Bestimmung der μ_{I0} und σ_{I0} berücksichtigt man die Datenpunkte d_1 mit:

$$Ab_{\Delta t_i} \sim N(\mu_I - a\Delta t_i, \sigma_I^2 + b\Delta t_i).$$

Mit den zuerst geschätzten a_0 und b_0 erhält man den Zusammenhang:

$$\Rightarrow Ab_{\Delta t_i} + a_0\Delta t_i \sim N(\mu_I, \sigma_I^2 + b_0\Delta t_i).$$

Der Erwartungswertschätzer liefert dann

$$\mu_{I0} = \frac{1}{n_1} \sum_{j=1}^{n_1} Ab_{\Delta t_i} + a_0 \Delta t_i.$$

Mit den bisherigen Startwerten folgt

$$\frac{Ab_{\Delta t_i} - (\mu_{I0} - a_0 \Delta t_i)}{\sqrt{\sigma_I^2 + b_0 \Delta t_i}} \sim N(0, 1).$$

Der schon verwendete Varianzschätzer liefert

$$1 = \frac{1}{1 - n_1} \sum_{i=1}^{n_1} \frac{(Ab_{\Delta t_i} - (\mu_{I0} - a_0 \Delta t_i))^2}{\sigma_{I0}^2 + b_0 \Delta t_i}.$$

Zur Vereinfachung werden nun die Δt_i mit gemitteltem $\bar{\Delta t}_i = \frac{1}{n_1} \sum_{i=1}^{n_1} \Delta t_i$ ersetzt.

Damit lässt sich σ_{I0} schätzen zu:

$$\sigma_{I0} = \frac{1}{1 - n_1} \sum_{i=1}^{n_1} (Ab_{\Delta t_i} - (\mu_{I0} - a_0 \Delta t_i))^2 - \frac{b_0}{n_1} \sum_{i=1}^{n_1} \Delta t_i.$$

Damit die Ungleichungsrestriktionen erfüllt sind, werden noch folgende Transformationen eingeführt:

$$a = \hat{a}^2, \quad b = \hat{b}^2 \quad \sigma_I = \hat{\sigma}_I^2.$$

Zur Lösung des jetzt unrestringierten Minimierungsproblems mit obigen Startwerten und den Parametern $(\hat{a}, \hat{b}, \mu_I, \hat{\sigma}_I)$ wurde die Matlab-Funktion `fminsearch` verwendet.

5.4 Statistische Modellanalyse und Ausblick

Nach der Bestimmung der Parameterwerte wird zunächst versucht, mehr über die Eigenschaften des Modells zu erfahren. Hierfür werden in Abbildung 5.3 die einzelnen Parameter paarweise geplottet. Sind zwei Parameter voneinander unabhängig, würde sich das durch eine gleichmäßige Punktwolke widerspiegeln. Dieses lässt sich annähernd an den Grafen von (a, σ) oder (b, σ) erkennen. Betrachtet man dagegen die Grafen von (μ, σ) oder auch (a, b) , so lässt sich fast ein linearer Zusammenhang vermuten.

Zur Hauptkomponentenanalyse betrachtet man hierzu die Korrelationsmatrix:

$$R = \begin{pmatrix} 1.0000 & 0.5989 & 0.6542 & 0.5316 \\ 0.5989 & 1.0000 & 0.1962 & 0.1650 \\ 0.6542 & 0.1962 & 1.0000 & 0.7600 \\ 0.5316 & 0.1650 & 0.7600 & 1.0000 \end{pmatrix}$$

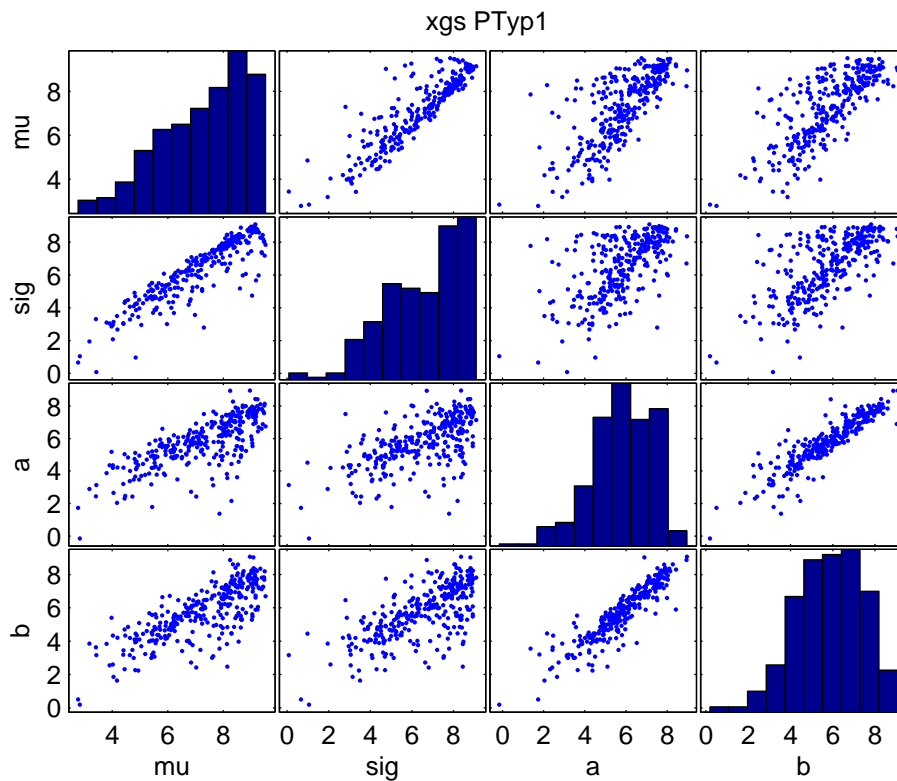


Abbildung 5.3: Korrelationsanalyse der identifizierten Parameter für PTyp1.

Mit Werten, die deutlich über 0.5 liegen, erkennt man hier die Korrelation von a und b , wobei diese relativ unkorreliert (< 0.2) zu σ sind. Auffällig ist hierbei der Zusammenhang dieser drei Parameter mit μ .

Betrachtet man weiter die Eigenwerte

$$\{0.1826, 0.2970, 1.0162, 2.5042\}$$

mit den zugehörigen Eigenvektoren

$$V = \begin{pmatrix} 0.5604 & -0.5533 & -0.2585 & 0.5595 \\ -0.3139 & 0.4032 & -0.7848 & 0.3506 \\ -0.6877 & -0.3105 & 0.3604 & 0.5483 \\ 0.3383 & 0.6594 & 0.4328 & 0.5132 \end{pmatrix},$$

so fällt die deutliche Unterteilung in > 1 und < 1 auf. Dieses legt den Verdacht nahe, dass sich das gesamte Modell auf zwei Parameter reduzieren ließe. Hierfür müssten jedoch weitere Untersuchungen durchgeführt werden, die den Rahmen dieser Arbeit bei weitem übersteigen würden. Daher wird im Folgenden auf das Modell mit vier Parametern aufgebaut.

Kapitel 6

Die Immunisierungsstrategie als Kontrollproblem

In diesem Kapitel erfolgt eine Einführung in die Kontrolltheorie. Die Fallstudie Immunisierungsstrategie wird mit Hilfe der stochastischen Programmierung in ein Kontrollproblem überführt, das im kontinuierlichen Fall auf Konsistenz und Eindeutigkeit untersucht wird. Unter Berücksichtigung einschränkender Annahmen wird der konvergente Fall diskutiert und die Existenz einer optimalen Lösung nachgewiesen. Anschließend wird ein numerisches Verfahren vorgestellt, mit dem sich die optimale Lösung bestimmen lässt. Zum Schluss finden die vorgestellten Verfahren beispielhaft eine Anwendung auf dem konkreten Fall eines Patienten.

6.1 Einführung in die Stochastische Programmierung

Die Einführung folgt den Erläuterungen von [WHI96, CHA04, KAL95].

Ein Problem bei der Modellierung besteht oft in der Unvollständigkeit der vorhandenen Informationen. Meist liegen nicht sämtliche Informationen vor, um eine optimale Entscheidung zu treffen, oder aber das System ist zu komplex, um alle Aspekte zu berücksichtigen. Viele Systeme werden auch von unvorhersagbaren äußeren Eingängen beeinflusst. Ist die Zukunft eines solchen Systems nicht mehr perfekt vorhersagbar, lassen sich die bisherigen Methoden nicht anwenden.

Es gibt nun zwei verschiedene Herangehensweisen, um auf eine natürliche mathematische Formulierung zu kommen. Eine liegt in der Anwendung einer stochastischen Formulierung. Hierbei benutzt man ein Wahrscheinlichkeitsmodell für den Prozess, in dem alle Variablen gemeinsam als Zufallsvariablen definiert sind. Die zu mes-

senden Variablen können dann genutzt werden, um auf die anderen zu schließen. Damit wird die Steuerung dann so gewählt, dass sie die erwarteten Kosten nach einem Optimierungskriteriums minimiert.

Die zweite Herangehensweise liegt im Minimax-Ansatz. Hierbei nimmt man unter Berücksichtigung der gemessenen Variablen für die nicht zu messenden Variablen den schlimmsten Fall bezüglich des Optimierungskriteriums an. Die Betrachtung der bedingten Erwartung wird hierbei durch das bedingte Maximum (der Kosten) ersetzt.

Im Folgenden wird genauer auf den stochastischen Ansatz eingegangen.

Eine spezielle Eigenschaft der Steuerungsoptimierung liegt in ihrer Mehrstufigkeit: Man erzeugt eine Sequenz von Entscheidungen in der Zeit, wobei im Allgemeinen für die späteren Entscheidungen mehr Informationen vorliegen als für die früheren. Daher ist es sinnvoll, zuerst den einstufigen Fall zu betrachten, bei dem nur eine Entscheidung zu treffen ist. Man hat eine Beobachtung y und muss eine Kontrollmaßnahme u wählen. Dann wird eine Kostenfunktion C eingeführt. Ihr Wert hängt von der Entscheidung u und dem unbekanntem „echten“ Zustand des Systems ab. Hierfür schreibt man $C(u)$. $C(u)$ ist eine Zufallsvariable, deren Verteilung von u abhängt. Nun nimmt man ein stochastisches Modell an, in welchem der Wert der Kosten $C(u)$ für verschiedene u und den messbaren y gemeinsam als Zufallsvariable definiert ist. Eine Strategie beschreibt u als Funktion $u(y)$ der gemessenen y ; die Werte der Steuerung sind so zu wählen, dass sie $E[C(u(y))]$ minimiert.

Definition 6.1

Eine Entscheidungsfunktion $\hat{u}(y)$ heißt optimal, wenn

$$E[C(\hat{u}(y))] = \min E[C(u(y))]$$

gilt.

Satz 6.2

Die optimale Entscheidungsfunktion $\hat{u}(y)$ ist bestimmt durch die Wahl von u als minimierender Wert von $E[C(u)|y]$.

Beweis:

Wenn eine Entscheidungsfunktion $u(y)$ gewählt ist, dann gilt für die erwarteten Kosten

$$E[C(u(y))] = E[E[C(u(y))|y]] \geq E[\inf_u E[C(u)|y]]$$

und die untere Grenze wird gerade durch die im Satz vorgegebene Regel erreicht. \diamond

Bemerkung 6.3

Hervorzuheben ist, dass dadurch das ursprünglich eingeschränkte Problem, den Erwartungswert der Kosten $E[C(u(y))]$ unter Berücksichtigung der Funktion $u(y)$ zu

minimieren (wobei also das minimierende u zwingend eine Funktion in Abhängigkeit von y ist), reduziert worden ist auf das freie Problem, nämlich $E[C(u)|y]$ unter Berücksichtigung des Parameters u zu minimieren.

u wie auch y sind Variablen, deren Vorgabe die Wahrscheinlichkeitsverteilung der Kosten $C(u)$ beeinflusst. Die Variable y ist eine Zufallsvariable, deren Beschreibung die Verteilung von C bedingt. Die Variable u ist anfänglich nicht zufällig, ihre Werte können aber durch den Optimierer frei gewählt werden und parametrisieren somit die Verteilung von C .

Nimmt man an, dass das Problem eine Zustandsvariable x , welche den „exakten“ Zustand wiedergibt, zulässt, so sind die Kosten einer deterministischen Funktion $C(x, u)$ abhängig von u und x . Wenn also x bekannt ist, würde man u so wählen, dass es $C(x, u)$ minimiert. Man kennt aber nur ein y , welches man als eine nicht vollständige Messung von x betrachten kann. Die gemeinsame Verteilung von x und y ist unabhängig von u , weil die Werte dieser Zufallsvariablen, gemessen oder nicht, schon vorliegen, bevor die Entscheidung u getroffen wird.

Satz 6.4

Angenommen, das Problem lässt eine Zustandsvariable x zu, $C(x, u)$ sei die Kostenfunktion und $f(x, y)$ die gemeinsame Dichte von x und y in Abhängigkeit des Produktes der Maße $\mu_1(dx)\mu_2(dy)$. Dann ist der optimale Wert von u derjenige, welcher

$$\int C(x, u) f(x, y) \mu_1 dx$$

minimiert.

Beweis:

Nimmt man zur Vereinfachung an, dass x und y diskrete Zufallsvariablen mit einer gemeinsamen Verteilung $P(x, y)$ sind; das formale Verallgemeinern ist dann klar. In diesem Fall gilt:

$$E[C(u)|y] = \sum_x C(x, u) P(x|y) \propto \sum_x C(x, u) P(x, y) = \sum_x C(x, u) P(x) P(y|x),$$

wobei das Proportionalzeichen einem Faktor $P(y)^{-1}$ unabhängig von u entspricht. Der dritte Ausdruck ist nun das Analogon zum Integralausdruck in Satz 6.3. \diamond

Erweitert man die Analyse des einstufigen Falles auf den mehrstufigen Fall, dann erhält man ein Kontrolloptimierungsproblem in diskreter Zeit. Hierfür sind zuerst einige Notationen zu definieren.

Man nehme an, dass der Prozess über eine Zeitspanne $t = 0, \dots, h$ optimiert werden soll. W_0 bezeichne alle verfügbaren Informationen zum Zeitpunkt Null. x_t beschreibe die Werte der Prozessvariablen zum Zeitpunkt t und X_t den bisherigen Prozessverlauf $\{x_1, \dots, x_t\}$. Entsprechend seien y_t die Werte der zu messenden Variablen und

u_t die Steueraktion zum Zeitpunkt t . Ebenso sei analog dazu der Verlauf Y_t und U_t beschrieben. Beschreibt W_t die verfügbaren Informationen zum Zeitpunkt t , dann kann man voraussetzen, dass $W_t = \{W_0, Y_t, U_{t-1}\}$ ist.

Eine realisierbare Steuerung A ist eine, die u_t als Funktion von W_t für $t=1, \dots, h-1$ angibt. Es sei nun C als eine Kostenfunktion angenommen. Man könnte sie als Funktion in Abhängigkeit von X_h und U_{h-1} einführen. Einfacher ist es aber, sie als Zufallsvariable zu betrachten, deren Verteilung gemeinsam mit der Veränderung des Prozesses und den gemessenen Größen durch die Wahl der Kontrollsequenz U_{h-1} parametrisiert ist. Das Ziel ist es dann, A so zu wählen, dass $E_A[C]$ minimal wird.

Definition 6.5

Die Gesamtwert-Funktion $G(W_t)$ sei definiert als die minimal zu erwartenden Kosten in Abhängigkeit der Informationen zum Zeitpunkt t . Es gilt also:

$$G(W_t) = \inf_A E_A[C|W_t].$$

Hierbei ist E_A der Erwartungswertoperator, eingeführt durch die Steuerung A .

Satz 6.6

(Das Dynamische-Programmierungsprinzip)

Die Gesamtwert-Funktion $G(W_t)$ folgt einer Rückwärtsrekursion (der dynamischen Programmierungsgleichung oder Optimalitätsgleichung)

$$G(W_t) = \inf_{u_t} E[G(W_{t+1}|W_t, u_t)] \quad (t = 1, \dots, h-1) \quad (*)$$

mit der abschließenden Bedingung

$$G(W_h) = E[C|W_h]. \quad (**)$$

Der minimierende Wert für u_t von (*) ist der optimale Wert der Steuerung zum Zeitpunkt t .

Für den *Beweis* sei auf [WHI96] verwiesen.

In diesem Satz liegt die Andeutung, dass die Ausdrücke (*) und (**) steuerungsunabhängig sind. Die Begründung hierfür liegt bei (**) darin, dass alle Entscheidungen in der Vergangenheit gemacht wurden. Die Gleichung (*) liefert offensichtlich einen kompakten und allgemeinen Ausdruck für das Dynamische-Programmierungsprinzip, formuliert als Ausdruck der derzeit maximal verfügbaren Informationen W_t .

Mit den folgenden drei Voraussetzungen, welche die Struktur des Prozesses beschreiben, lässt sich der Satz noch spezifizieren.

1. Markov Dynamik:

Die Prozessvariable x habe die Eigenschaft

$$P(x_{t+1}|X_t, U_t) = P(x_{t+1}|x_t, u_t).$$

D.h., die Wahrscheinlichkeit für den Wechsel zum nächsten Zustand hängt nur vom Zustand des Prozesses und der Steuerung zum Zeitpunkt t ab und nicht von ihrer Vergangenheit.

2. *Trennbare Kostenfunktion:*

Die Kostenfunktion lässt sich in die Summe der unmittelbaren und der abschließenden Kosten aufspalten. D.h., es muss gelten:

$$C = \sum_{t=0}^{h-1} c(x_t, u_t, t) + C_h(x_h) = \sum_{t=0}^{h-1} c_t + C_h.$$

3. *Genauere Zustandsmessung:*

Es ist nötig, den genauen Zustand messen zu können. D.h., x_t sollte zum Zeitpunkt t bekannt sein, wenn u_t gesetzt werden soll; also gilt:

$$W_t = (X_t, U_{t-1}).$$

Bevor der Satz formuliert werden kann, sind noch folgende Definitionen sinnvoll.

Definition 6.7

Es sei C_t die Summe der zukünftigen Kosten zum Zeitpunkt t definiert als

$$C_t = \sum_{\tau=t}^{h-1} c_\tau + C_h.$$

Weiter sei die Funktion $F(W_t)$ definiert als

$$F(W_t) = \inf_A E_A[C_t | W_t].$$

Somit lässt sich folgern:

$$G(W_t) = \sum_{\tau=0}^{t-1} c_\tau + F(W_t).$$

Satz 6.8

Unter obigen Voraussetzungen 1.-3. gilt:

(i) $F(W_t)$ ist eine Funktion alleine von x_t und t . Schreibt man $F(x_t, t)$, so gilt für die dynamischen Programmierungsgleichung

$$F(x_t, t) = \inf_{u_t} \{c(x_t, u_t, t) + E[F(x_{t+1}, t+1) | x_t, u_t]\} \quad (t \leq h) \quad (+)$$

mit der abschließenden Bedingung

$$F(x_h, h) = C_h(x_h). \quad (++)$$

- (ii) Der minimierende Wert von u_t in (+) ist der optimale Wert für die Steuerung zum Zeitpunkt t , welche somit auch eine Funktion von x_t und t ist.

Beweis:

Der Wert von $F(W_h)$ ist $C_h(x_h)$, somit ist die behauptete Reduzierung von F zum Zeitpunkt h gültig. Angenommen, die Aussage sei gültig zum Zeitpunkt $t+1$. Dann vereinfacht sich die allgemeine dynamische Programmierungsgleichung zu

$$F(x_t, t) = \inf_{u_t} \{c(x_t, u_t, t) + E[F(x_{t+1}, t+1)|X_t, U_t]\}$$

und der minimierende Wert von u_t ist optimal. Durch Voraussetzung Nr. 1. vereinfacht sich hierbei aber die rechte Seite zur rechten Seite von (+). Alle weiteren Aussagen folgen daraus durch Induktion. \diamond

Mit diesem Satz hat man die Vereinfachung, dass nicht sämtliche Informationen aus der Vergangenheit gespeichert werden müssen. Es reichen also die aktuellen Informationen für die Optimierung aus. Die optimale Steuerungsregel, welche sich aus (+) ableiten lässt, ist wieder in einer geschlossenen Schleife gegeben, da die Steuerung vor dem Zeitpunkt t nicht bekannt sein muss.

Nachdem nun eine Einführung in das Vorgehen der stochastischen dynamischen Programmierung gegeben wurde, sei für den zeitkontinuierlichen Fall auf die Literatur [WHI96, CHA04, KAL95] verwiesen.

Abschließend sind für die Überprüfung der Kostenfunktion auf Konsistenz und Eindeutigkeit noch folgende Definitionen nötig:

Definition 6.9

Unter $L^p(E)$, $1 \leq p < \infty$ versteht man den Raum aller messbaren Funktionen $y : E \rightarrow \mathbb{R}$ mit der Eigenschaft

$$\int_E |y(x)|^p dx < \infty.$$

Dabei werden Funktionen, die sich nur auf einer Menge vom Maß Null unterscheiden, als gleich angesehen. Sie gehören einer gemeinsamen Äquivalenzklasse an. $L^p(E)$ wird versehen mit der Norm

$$\|y\|_{L^p(E)} = \left(\int_E |y(x)|^p dx \right)^{1/p}.$$

Definition 6.10

Mit L^∞ wird der Raum aller fast überall gleichmäßig beschränkten und messbaren Funktionen $y : E \rightarrow \mathbb{R}$ bezeichnet, versehen mit der Norm

$$\|y\|_{L^\infty(E)} = \operatorname{ess\,sup}_{x \in E} |y(x)| := \inf_{|F|=0} \left(\sup_{x \in E \setminus F} |y(x)| \right).$$

6.2 Kontrollproblem

Sind die Parameter für einen Patienten bestimmt, so liegt das eigentliche Ziel in der Bestimmung einer angepassten Kontroll- oder auch Steuerungsstrategie. D.h., zu welchen Zeitpunkten soll der Patient geimpft oder getestet und in welchen Zeiträumen soll abgewartet werden? Wir suchen also eine Steuerstrategie $A(\mu, \sigma, t)$, die uns zu jedem Zeitpunkt t für eine Zustand (μ, σ) angibt, was zu tun ist. Hierfür muss zunächst, wie in der Einführung dargelegt, eine Kostenfunktion definiert werden.

Definition 6.11

Zu einer gegebenen Steuerung $A(\mu, \sigma, t)$ (genügend glatt, d.h. τ_0 in (e) existiert) mit $(\mu, \sigma) \in \bar{\Omega}$ seien die erwarteten Gesamtkosten im Zustand (μ, σ) zum Zeitpunkt t definiert durch

$$\begin{aligned} \text{(a) Zeitpunkt } t = T \quad C(\mu, \sigma^2, T | A(\mu, \sigma, t) = N) &= 0 \\ C(\mu, \sigma^2, T | A(\mu, \sigma, t) = V) &= \alpha \\ C(\mu, \sigma^2, T | A(\mu, \sigma, t) = O) &= \beta. \end{aligned}$$

$$\begin{aligned} \text{(b) Zeitpunkt } t > T \quad A(\mu, \sigma, t) &= N \\ C(\mu, \sigma^2, t | A(\mu, \sigma, t) = N) &= 0. \end{aligned}$$

$$\begin{aligned} \text{(c) „Impfung“, Zeitpunkt } t < T \\ C(\mu, \sigma^2, t | A(\mu, \sigma^2, t) = V) &= \alpha + C(\mu_I, \sigma_I^2, t). \end{aligned}$$

$$\begin{aligned} \text{(d) „Testen“, Zeitpunkt } t < T \\ \text{Sei } \pi(\mu'; \mu, \sigma^2) \text{ die Verteilung von } N(\mu, \sigma^2). \text{ Das bedeutet} \end{aligned}$$

$$P(x_1 < Z < x_2) = \int_{x_1}^{x_2} \pi(\mu'; \mu, \sigma^2) d\mu'.$$

$$\text{Sei weiter } \Phi(x) = \int_{-\infty}^x \pi(\mu'; \mu, \sigma^2) d\mu'.$$

$$\begin{aligned} C(\mu, \sigma^2, t | A(\mu, \sigma^2, t) = O) &= \beta + \Phi(-\mu/\sigma)(\alpha + C(\mu_I, \sigma_I^2, t)) \\ &+ \int_0^\infty C(\mu', 0, t) \pi(\mu'; \mu, \sigma^2) d\mu'. \end{aligned}$$

$$\begin{aligned} \text{(e) „Nichts tun“, Zeitpunkt } t < T \\ \text{Sei } A(\mu, \sigma^2, t) = N \text{ und} \end{aligned}$$

$$\tau_0 = \min\{\tau | A(\mu - a\tau, \sigma^2 + b\tau, t + \tau) \neq N\}.$$

Dann ist

$$C(\mu, \sigma^2, t | A(\mu, \sigma^2, t) = N) = e^{-\lambda\tau_0} C(\mu - a\tau_0, \sigma^2 + b\tau_0, t + \tau_0)$$

mit $\lambda > 0$ als „discount“ Faktor.

Bei dieser Definition wird also davon ausgegangen, dass die Steuerung A schon vorliegt. Der zugrunde liegende Gedanke für C ist folgender: Die Kosten eines Patienten zu einem Zeitpunkt t berechnen sich aus den gerade aufkommenden Kosten und den zukünftig zu erwartenden Kosten. Weiter wird davon ausgegangen, dass ein Patient zu einem Zeitpunkt T die Klinik wieder verlässt. Ab diesem Zeitpunkt entstehen keine Folgekosten mehr. Dieses drücken die Punkte (a) und (b) aus. Wird in einem Zustand geimpft (Punkt (c)), so muss man die Kosten der Impfung α und die daraus resultierenden Kosten, also die Kosten zum Zustand (μ_I, σ_I^2) , tragen. Nach einem Test (Punkt (d)) ist der Zustand genau bekannt. Zuerst sind also die Kosten des Testes β zu berücksichtigen. Anschließend gilt es noch, die Folgekosten zu berechnen, die sich aus dem Produkt der Wahrscheinlichkeit des Folgezustandes und dessen Kosten ergeben. Für die Aktion „Nichts tun“ (Punkt (e)) erhält man abgeschwächt die Kosten aus der nächsten Aktion.

Damit $C_A(\mu, \sigma, t|A = O)$ wohl definiert ist, muss $C_A(\mu, 0, t|A) \in L^\infty$ aufgrund des Integrales in (e) gelten. Bei der weiteren Untersuchung von $C_A(\mu, \sigma, t)$ fällt auf, dass $C_N(\mu, \sigma, t)$ nicht auf ganz $\bar{\Omega}$ definiert ist. Daher führt man eine erweiterte Kostenfunktion ein.

Definition 6.12

Die erweiterten Kostenfunktionen der einzelnen Aktionen seien definiert als

$$\begin{aligned} \hat{C}_V(\mu, \sigma^2, t|A(\mu, \sigma, t)) &= C_V(\mu, \sigma^2, t|A(\mu, \sigma, t)) \\ \hat{C}_O(\mu, \sigma^2, t|A(\mu, \sigma, t)) &= C_O(\mu, \sigma^2, t|A(\mu, \sigma, t)) \\ \hat{C}_N(\mu, \sigma^2, t|A(\mu, \sigma, t)) &= \begin{cases} T_1 & (\mu, \sigma^2) \in \Gamma^+ \\ T_2 & A(\mu, \sigma^2, t) = N \\ T_3 & \begin{cases} A(\mu, \sigma^2, t) \neq N, \\ \tau = \min\{\tau|A(\mu - a\tau, \sigma^2 + b\tau, t + \tau) \neq N\} \\ T_4 & \begin{cases} (\mu - a\tau, \sigma^2 + b\tau) \in \Gamma^+, \\ A(\mu - a\hat{\tau}, \sigma^2 + b\hat{\tau}, t + \hat{\tau}) \neq N, \forall \hat{\tau} \in [0, \tau]. \end{cases} \end{cases} \end{cases} \end{aligned}$$

mit

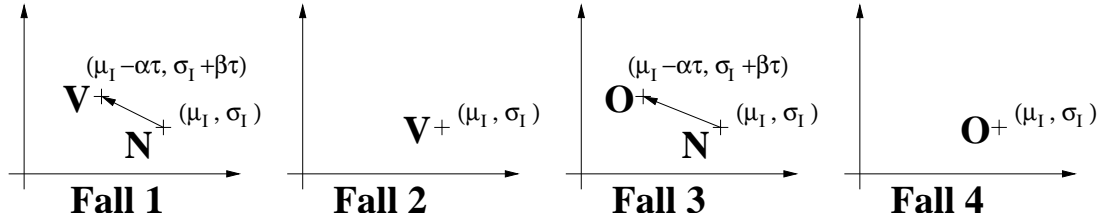
$$\begin{aligned} T_1 &= C(\mu, \sigma^2, t|A(\mu, \sigma, t)) \\ T_2 &= C_N(\mu, \sigma^2, t|A(\mu, \sigma, t)) \\ T_3 &= e^{-\lambda\tau} C_N(\mu - a\tau, \sigma^2 + b\tau, t + \tau|A(\mu, \sigma, t + \tau)) \\ T_4 &= e^{-\lambda\tau} C(\mu - a\tau, \sigma^2 + b\tau, t + \tau|A(\mu, \sigma, t + \tau)). \end{aligned}$$

Zur Überprüfung der Wohldefiniertheit von $C_A(\mu, \sigma)$ bestimmt man die benötigten Konsistenzbedingungen. Hierfür untersucht man zuerst die Existenz von

$$C_{I,A}(t) := C(\mu_I, \sigma_I^2, t|A(\mu, \sigma, t)).$$

Dabei hat man die folgenden Fälle zu berücksichtigen:

Aus der Abbildung 6.1 lassen sich folgende vier Fälle erkennen:

Abbildung 6.1: Fallunterscheidung (Projektion in (μ, σ^2) -Ebene).

Fall 1: Nach dem Impfen nichts tun und dann wieder impfen:

$$A(\mu - a\hat{\tau}, \sigma^2 + b\hat{\tau}, t + \hat{\tau}) = N, \hat{\tau} \in [0, \tau) \text{ und } A(\mu - a\tau, \sigma^2 + b\tau, t) = V.$$

Fall 2: Nach dem Impfen sofort wieder impfen:

$$A(\mu_I, \sigma_I^2, t) = V.$$

Fall 3: Nach dem Impfen nichts tun und dann wieder testen:

$$A(\mu - a\hat{\tau}, \sigma^2 + b\hat{\tau}, t + \hat{\tau}) = N, \hat{\tau} \in [0, \tau) \text{ und } A(\mu - a\tau, \sigma^2 + b\tau, t) = O.$$

Fall 4: Nach dem Impfen sofort testen:

$$A(\mu_I, \sigma_I^2, t) = O.$$

Man beachte hierbei für τ die Abhängigkeit von t .

Für $C_{I,A}(t)$ lassen sich unter Berücksichtigung der Fälle die entsprechenden Konsistenzbedingungen bestimmen:

$$\text{zu Fall 1: } C_{I,A}(t) = e^{-\lambda\tau}(\alpha + C_{I,A}(t + \tau)).$$

$$\text{zu Fall 2: } C_I \text{ divergent.}$$

$$\text{zu Fall 3: } (\hat{\mu}, \hat{\sigma}^2) := (\mu - a\tau, \sigma^2 + b\tau), \hat{\Phi} := \Phi(-\frac{\hat{\mu}}{\hat{\sigma}}), \hat{\pi}(\mu') := \pi(\mu', \hat{\mu}, \hat{\sigma}^2)$$

$$C_{I,A}(t) = e^{-\lambda\tau}(\beta + \hat{\Phi}(\alpha + C_{I,A}(t + \tau))) + \int_0^\infty C(\mu', 0, t + \tau | A(\mu', 0, t + \tau)) \hat{\pi}(\mu') d\mu'.$$

zu Fall 4: Spezialfall von Fall 3 mit $\tau = 0$

$$\Rightarrow C_I = \frac{1}{1 - \hat{\Phi}}(\beta + \hat{\Phi}\alpha + \int_0^\infty C(\mu', 0, t | A(\mu', 0, t)) \hat{\pi}(\mu') d\mu').$$

Dieser Fall 4 wird in Zukunft nicht berücksichtigt, weil nach einer Impfung erst eine Zeit verstreicht, bis sich diese auf den Titer auszuwirken beginnt. Da das Integral in $C(\mu, \sigma^2, t | A(\mu, \sigma^2, t) = O)$ zu Widersprüchen führen kann, genügt zur Überprüfung der Konsistenz nur $C(\mu, 0, t | A(\mu, 0, t))$ zu betrachten. Alle weiteren

$C(\mu, \sigma^2, t|A(\mu, \sigma^2, t))$ sind dadurch und mit $C_{I,A}(t)$ definiert. Hierbei soll für die Steuerung gelten:

$$\mu^* > 0, \sigma^{*2} = b * \tau > 0 \quad \text{mit} \quad (\mu^* - a\tau, \sigma^{*2}) \in \Gamma^+$$

1. $A(\mu, \sigma^2, t) = N \quad \forall (\mu, \sigma^2) \in \Omega, \sigma^2 \leq \sigma^{*2} \forall t$
2. $A(\mu, \sigma^2, t) = V \quad \forall (\mu, \sigma^2) \in \Gamma^+, \sigma^2 \leq \sigma^{*2} \forall t.$

Dieses sind keine gravierenden Einschränkungen. 1. ist sinnvoll, da, sobald der Zustand eines Patienten exakt bekannt ist, erst einmal nichts unternommen werden muss. Dieser Fall liegt bei $(\mu, 0) \in \gamma^-$ vor. 2. bedeutet, dass, sobald der Erwartungswert μ unter einem Wert echt größer Null liegt, nicht mehr getestet sondern sofort geimpft wird, da mit hoher Wahrscheinlichkeit sowieso geimpft werden muss und man sich daher die Kosten des Testens sparen kann.

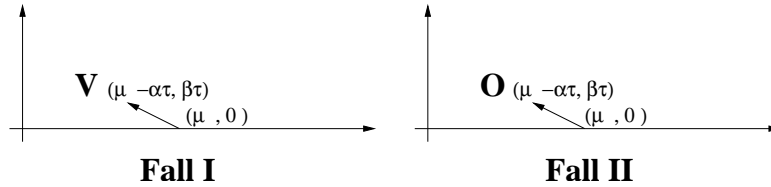


Abbildung 6.2: Fallunterscheidung (Projektion in (μ, σ^2) -Ebene).

Für $C(\mu, 0, t|A(\mu, 0, t))$ können mit obigen Einschränkungen an A nur die 2 folgenden Fälle eintreten (siehe Abbildung 6.2):

$$A(\mu - a\hat{\tau}, \sigma^2 + b\hat{\tau}, t + \hat{t}) = N, \quad \hat{t}, \hat{\tau} \in [0, \tau).$$

Fall I: Nach $(\mu, 0)$ nichts tun, dann nach τ_2 impfen

$$\begin{aligned} A(\mu - a\hat{\tau}, b\hat{\tau}, t + \hat{\tau}) &= N \quad \hat{\tau} \in [0, \tau_2) \\ A(\mu - a\tau_2, b\tau_2, t + \tau_2) &= V. \end{aligned}$$

Fall II: Nach $(\mu, 0)$ nichts tun, dann nach τ_2 impfen

$$\begin{aligned} A(\mu - a\hat{\tau}, b\hat{\tau}, t + \hat{\tau}) &= N \quad \hat{\tau} \in [0, \tau_2) \\ A(\mu - a\tau_2, b\tau_2, t + \tau_2) &= O. \end{aligned}$$

Beachte, τ_2 ist abhängig von μ und beide τ_i sind größer Null und von t abhängig. Es sei $I(\mu) := (\mu - a\tau_2, b\tau_2)$. Daraus ergeben sich die Konsistenzbedingungen zu Fall I:

$$C(\mu, t|A(\mu, 0, t)) = e^{-\lambda\tau_2}(\alpha + C_{I,A}(t + \tau_2)).$$

zu Fall II:

$$C(\mu, t|A(\mu, 0, t)) = e^{-\lambda\tau_2} \left(\beta + \Phi\left(-\frac{\mu - a\tau_2}{\sqrt{b\tau_2}}\right)(\alpha + C_{I,A}(t + \tau_2)) + \int_0^\infty C(\mu', 0, t + \tau_2|A(\mu', 0, t + \tau_2))\pi(\mu'; I(\mu))d\mu' \right).$$

$$A_1(\hat{\tau}) := A(\mu_I - a\hat{\tau}, \sigma_I^2 + b\hat{\tau}, t + \hat{\tau}) \text{ und } A_2(\hat{\tau}) := A(\mu - a\hat{\tau}, b\hat{\tau}, t + \hat{\tau}).$$

Außerdem sei

$$C(\mu, t) = (C_A(\mu, t), C_{I,A}(t)).$$

Man beachte, dass $C_A(\mu, t)$ und $C_{I,A}(t)$ in den Konsistenzbedingungen immer von zukünftigen Werten bestimmt werden und mit der Definition von $C_A(\mu, \sigma, t)$ alle Zeitpunkte $t \geq T$ definiert sind. Damit sind $C_A(\mu, t)$ und $C_{I,A}(t)$ in der Zeit rückwärtig definiert.

Somit erhält man zusammengefasst die Konsistenzbedingung

$$C(\mu, t) = F_1(\mu; A(t)) + T(C(\mu, t); A(t))$$

mit dem Integraloperator

$$T(C(\mu, t); A(t)) := F_2(\mu; A(t)) \int_0^\infty C(\mu', t)\pi(\mu'; I(\mu, t)) d\mu.$$

Insgesamt ergeben sich ausgeschrieben für $C_{I,A}(t)$ und $C_A(\mu, t)$ die Konsistenzbedingungen:

$$C_{I,A}(t) = \begin{cases} e^{-\lambda\tau_1}(\alpha + C_{I,A}(t + \tau_1)) & \text{wenn } A(\mu_I - a\tau_1, \sigma_I^2 + b\tau_1, t + \tau_1) = V, t + \tau_1 < T \\ e^{-\lambda\tau_1}(\beta + \hat{\Phi}(\alpha + C_{I,A}(t + \tau_1)) + \int_0^\infty C(\mu', 0, t + \tau_1|A(\mu', 0, t + \tau_1))\hat{\pi}(\mu')d\mu') & \text{wenn } A(\mu_I - a\tau_1, \sigma_I^2 + b\tau_1, t + \tau_1) = O, t + \tau_1 < T \\ 0 & \text{wenn } A(\mu_I - a\tau_1, \sigma_I^2 + b\tau_1, t + \tau_1) = N, t + \tau_1 \geq T \end{cases}$$

$$C(\mu, t|A(\mu, 0, t)) = \begin{cases} e^{-\lambda\tau_2}(\alpha + C_{I,A}(t)) & \text{wenn } A(\mu - a\tau_2, b\tau_2, t + \tau_2) = V, t + \tau_2 < T \\ e^{-\lambda\tau_2} \left(\beta + \Phi\left(-\frac{\mu - a\tau_2}{\sqrt{b\tau_2}}\right)(\alpha + C_{I,A}(t)) + \int_0^\infty C(\mu', 0, t + \tau_2|A(\mu', 0, t + \tau_2))\pi(\mu'; I(\mu))d\mu' \right) & \text{wenn } A(\mu - a\tau_2, b\tau_2, t + \tau_2) = O, t + \tau_2 < T \\ 0 & \text{wenn } A(\mu - a\tau_2, b\tau_2, t + \tau_2) = N, t + \tau_2 \geq T. \end{cases}$$

Nachdem die Konsistenzbedingungen bestimmt worden sind, geht man nun von allen Kostenfunktionen aus, die diese erfüllen und definiert sich nun alle zulässigen Steuerungen.

Es seien nun folgende Parameter, wie z.B. in Kapitel 5 bestimmt, gegeben:

1. $(\mu_I, \sigma_I) \in \Omega$
2. $a, b > 0$
3. $0 < \mu^* < \mu_I$
4. $\tau_1 > 0$
5. $\lambda > 0$.

Definition 6.13

Eine Steuerung A ist definiert als eine Abbildung

$$A(\mu, \sigma^2, t) \quad (\mu, \sigma^2, t) \in \bar{\Omega} \times \mathbb{R} \rightarrow \{N, V, O\},$$

die folgende Bedingungen erfüllt:

1. $A(\mu, \sigma^2, t) \neq N \quad \forall (\mu, \sigma^2) \in \Gamma^+$
2. $A(\mu, \sigma^2, t) = N \quad \forall (\mu, \sigma^2) \in \Omega, \sigma^2 \leq \sigma^{*2}$
3. $A(\mu, \sigma^2, t) = V \quad \forall (\mu, \sigma^2) \in \Gamma^+, \sigma^2 \leq \sigma^{*2}$
4. $A(\mu_I - a\hat{\tau}, \sigma_I^2 + b\hat{\tau}, t) = N, \quad \forall t \in (0, \tau_1]$
5. $A(\mu, \sigma^2, t) = N$ auf jeder Trajektorie $(\mu - at, \sigma^2 + bt)$ auf endlich vielen halboffenen Intervallen $(t_1, t_2]$
6. $C_A(\mu, t) \in L^\infty(\mu, t)$ und $C_{I,A} \in L^\infty(t)$.

Hierbei bedeutet 1., dass wie vorgegeben auf dem oberen Rand Γ^+ des erlaubten Gebietes Ω eingegriffen werden muss. 2. und 3. waren die einschränkenden Bedingungen, die bei der Bestimmung der Konsistenzbedingungen aufgetreten sind. 4. entspricht dem Abwarten nach einer Impfung, so dass die Impfung erst einmal wirken kann. 5. und 6. sind Glattheitsbedingungen, sowohl an A als auch dem resultierendem C .

Die Menge der erlaubten Steuerungen ist nicht leer, denn sie beinhaltet z.B. die Steuerung

$$A(\mu, \sigma^2, t) = \begin{cases} V & (\mu, \sigma^2) \in \Gamma^+, t \leq T \\ N & \text{sonst} \end{cases}.$$

Jetzt ist zu zeigen, dass durch diese Definition für jede Steuerung A auch genau eine Kostenfunktion $C_A(\mu, t) := C_A(\mu, 0, t) \in L^\infty(\mu, t) = C_A(\mu, 0, t)$ und eine Funktion $C_{I,A}$ definiert ist. D.h. es ist zu zeigen, dass der Integraloperator T eine Kontraktion auf $L^\infty(\mu, t)$ ist.

Lemma 6.14

Für jede wie oben definierte Steuerung A existiert ein eindeutige Kostenfunktion $C(\mu)$.

Beweis (Kontraktionsbeweis):

Für

$$C(\mu, t) = (C_A(\mu, t), C_{I,A}(t))$$

sei die Norm definiert

$$\|C(\mu, t)\|_{L^\infty(\mu, t)} = \max \left\{ \|C_A(\mu, t)\|_{L^\infty(\mu, t)}, \|C_{I,A}(t)\|_{L^\infty(t)} \right\}.$$

Behauptung: Zu gegebenen A gibt es zwei verschiedene Kostenfunktionen, die obige Konsistenzbedingungen erfüllen ($\exists C_1(\mu, t) \neq C_2(\mu, t)$).

$$D(\mu, t) := C_1(\mu, t) - C_2(\mu, t)$$

$$\|D(\mu, t)\|_{L^\infty(\mu, t)} \leq \max \|Konsistenzbedingungen\|_{L^\infty(\mu, t)}$$

Im Folgenden müsste die Norm immer auf denjenigen Raum beschränkt werden, in welchem die einzelnen Fälle eintreten. Eine Vergrößerung des Raumes ist jedoch durch die vollständige Definition von C auf diesen als Abschätzung nach oben erlaubt.

Wenn $A(\mu_I - a\tau_1, \sigma_I^2 + b\tau_1, t + \tau_1) = V$ und $t + \tau_1 < T$ gilt,

$$D_{I,A}(t) = e^{-\lambda\tau_1}(\alpha + D_{I,A}(t + \tau_1)).$$

Also gilt für die Norm

$$\begin{aligned} \|D_{I,A}(t)\|_{L^\infty(t)} &= \|e^{-\lambda\tau_1}(D_{I,A}(t + \tau_1))\|_{L^\infty(t)} \\ &\leq e^{-\lambda\tau_1} \|(D_{I,A}(t))\|_{L^\infty(t)} \\ &\leq e^{-\lambda\tau_1} \|(D(\mu, t))\|_{L^\infty(\mu, t)}. \end{aligned}$$

Wenn $A(\mu_I - a\tau_1, \sigma_I^2 + b\tau_1, t + \tau_1) = O$ und $t + \tau_1 < T$ ist, dann ergibt sich

$$D_{I,A}(t) = e^{-\lambda\tau_1}(\beta + \hat{\Phi}(\alpha + D_{I,A}(t + \tau_1)) + \int_0^\infty D(\mu', 0, t + \tau_1 | A(\mu', 0, t + \tau_1)) \hat{\pi}(\mu') d\mu').$$

Damit gilt für die Norm

$$\begin{aligned}
\|D_{I,A}(t)\|_{L^\infty(t)} &= \left\| e^{-\lambda\tau_1}(\beta + \hat{\Phi}(\alpha + D_{I,A}(t + \tau_1)) \right. \\
&\quad \left. + \int_0^\infty D(\mu', 0, t + \tau_1 | A(\mu', 0, t + \tau_1)) \hat{\pi}(\mu') d\mu' \right\|_{L^\infty(t)} \\
&\leq e^{-\lambda\tau_1} (\hat{\Phi} \|D_{I,A}(t)\|_{L^\infty(t)} + (1 - \hat{\Phi}) \|D_A(\mu, t)\|_{L^\infty(\mu, t)}) \\
&\leq e^{-\lambda\tau_1} \|D(\mu, t)\|_{L^\infty(\mu, t)}.
\end{aligned}$$

Wenn $A(\mu_I - a\tau_1, \sigma_I^2 + b\tau_1, t + \tau_1) = N$ also $t + \tau_1 \geq T$ ist, dann ergibt sich

$$D_{I,A}(t) = 0$$

und somit gilt für die Norm

$$\|D_{I,A}(t)\|_{L^\infty(t)} = 0.$$

Damit ergibt sich unter der Berücksichtigung, dass auch $\tau_1 > 0$ für die ersten beiden Bedingungen ist, für die ersten drei Bedingungen der Zusammenhang:

$$\|D_{I,A}(t)\|_{L^\infty(t)} < \|D(\mu, t)\|_{L^\infty(\mu, t)}.$$

Wenn $A(\mu - a\tau_2, b\tau_2, t + \tau_2) = V$ und $t + \tau_2 < T$ ist (Beachte $\tau_2 > 0$ aber beliebig klein), dann ergibt sich

$$D_A(\mu, t) = e^{-\lambda\tau_2}(D_{I,A}(t + \tau_2)).$$

Also gilt für die Norm

$$\begin{aligned}
\|D_A(\mu, t)\|_{L^\infty(\mu, t)} &= \|e^{-\lambda\tau_2}(D_{I,A}(t))\|_{L^\infty(t)} \\
&< \|D(\mu, t)\|_{L^\infty(\mu, t)}.
\end{aligned}$$

Wenn $A(\mu - a\tau_2, b\tau_2, t + \tau_2) = O$ und $t + \tau_2 < T$ ist (Beachte $\tau_2 > \tau^*$), dann ergibt sich

$$\begin{aligned}
D(\mu, t | A(\mu, 0, t)) &= e^{-\lambda\tau_2} \left(\Phi\left(-\frac{\mu - a\tau_2}{\sqrt{b\tau_2}}\right)(D_{I,A}(t + \tau_2)) \right. \\
&\quad \left. + \int_0^\infty D(\mu', 0, t + \tau_2 | A(\mu', 0, t + \tau_2)) \pi(\mu'; I(\mu)) d\mu' \right).
\end{aligned}$$

Also gilt für die Norm

$$\begin{aligned}
\|D_A(\mu, t)\|_{L^\infty(\mu, t)} &= \left\| e^{-\lambda\tau_2} \left(\Phi\left(-\frac{\mu - a\tau_2}{\sqrt{b\tau_2}}\right) (D_{I,A}(t + \tau_2)) \right. \right. \\
&\quad \left. \left. + \int_0^\infty D(\mu', 0, t + \tau_2 | A(\mu', 0, t + \tau_2)) \pi(\mu'; I(\mu)) d\mu' \right) \right\|_{L^\infty(\mu, t)} \\
&\leq e^{-\lambda\tau_2} \left(\Phi\left(-\frac{\mu - a\tau_2}{\sqrt{b\tau_2}}\right) \|D_{I,A}(t + \tau_2)\|_{L^\infty(t)} \right. \\
&\quad \left. + \left(1 - \Phi\left(-\frac{\mu - a\tau_2}{\sqrt{b\tau_2}}\right)\right) \|D_A(\mu, t)\|_{L^\infty(\mu, t)} \right) \\
&< e^{-\lambda\tau_2} \|D(\mu, t)\|_{L^\infty(\mu, t)}.
\end{aligned}$$

Wenn $A(\mu - a\tau_2, b\tau_2, t + \tau_2) = N$ also $t + \tau_2 > T$ ist, dann ergibt sich

$$D(\mu, t | A(\mu, 0, t)) = 0.$$

Also gilt für die Norm

$$\|D(\mu, t | A(\mu, 0, t))\|_{L^\infty(\mu, t)} = 0.$$

Damit ergibt sich für die zweiten drei Bedingungen die selbe Ungleichung

$$\|D_A(\mu, t)\|_{L^\infty(\mu, t)} < \|D(\mu, t)\|_{L^\infty(\mu, t)}.$$

Und somit müsste gelten:

$$\|D(\mu, t)\|_{L^\infty(\mu, t)} = \max \left\{ \|D_{I,A}(t)\|_{L^\infty(t)}, \|D_A(\mu, t)\|_{L^\infty(\mu, t)} \right\} < \|D(\mu, t)\|_{L^\infty(\mu, t)}$$

$$\Rightarrow \|D(\mu, t)\|_{L^\infty(\mu, t)} = 0$$

und deshalb müssen C_1 und C_2 identisch sein. \diamond

Um nun eine bestmögliche Steuerung zu finden, soll für einen Startzustand mit $Z_{t_0} \sim N(\mu_0, \sigma_0^2)$ eine Kontrollstrategie $A_{(\mu_0, \sigma_0^2, t_0)}^*$ gefunden werden, so dass

$$C(\mu_0, \sigma_0^2, t_0 | A_{(\mu_0, \sigma_0^2, t_0)}^*) \leq C(\mu_0, \sigma_0^2, t_0 | A) \quad \forall A.$$

Aus dem Bellmann-Prinzip kann direkt geschlossen werden, dass diese Steuerung $A_{(\mu_0, \sigma_0^2, t_0)}^*$, wenn sie existiert, so gewählt werden kann, dass sie unabhängig vom Anfangswert (μ_0, σ_0^2, t_0) ist, d.h.

$$C(\mu_0, \sigma_0^2, t_0 | A_{(\mu_0, \sigma_0^2, t_0)}^*) = C(\mu_0, \sigma_0^2, t_0 | A_{(\mu_1, \sigma_1^2, t_1)}^*).$$

Dieses ermöglicht die folgende Definition.

Definition 6.15

Eine Steuerung A^* heißt optimal, wenn $\forall(\mu, \sigma^2) \in \Omega$ gilt:

$$C(\mu_0, \sigma_0^2, t_0 | A^*) \leq C(\mu_0, \sigma_0^2, t_0 | A) \quad \forall A.$$

Für diese optimale Steuerung ergibt sich folgendes:

Monotonie von C**Lemma 6.16**

Die Kostenfunktion A^* wächst monoton, d.h.

$$C(\mu_0, \sigma_0^2, t_1 | A^*) \leq C(\mu_0, \sigma_0^2, t_2 | A^*) \quad t_1 > t_2.$$

Beweis:

Es gilt

$$A(\mu, \sigma^2, t) = A^*(\mu, \sigma^2, t + (t_1 - t_2))$$

und somit lässt sich folgern:

$$C(\mu_0, \sigma_0^2, t_2 | A^*) \geq C(\mu_0, \sigma_0^2, t_1 | A) \geq C(\mu_0, \sigma_0^2, t_1 | A^*).$$

◇

Asymptotisches Verhalten von C**Lemma 6.17**

Die Kostenfunktion A^* konvergiert punktweise

$$\lim_{\tau \rightarrow \infty} C(\mu_0, \sigma_0^2, T - \tau) = C(\mu_0, \sigma_0^2).$$

Beweis:

Wir suchen eine Steuerstrategie für eine obere Grenze der Kosten

$$A(\mu, \sigma^2, t) = \begin{cases} V & (\mu, \sigma^2) \in ((R \times R^+) \setminus \Omega) \cup \Gamma_+ \\ N & \text{sonst} \end{cases}.$$

Dann gilt $C(.,., t | A^*) \leq C(.,., t | A) \leq C(\mu_I, \sigma_I^2, t)$ und

$$C(\mu_I, \sigma_I^2, t) \leq \lim_{\tau \rightarrow \infty} C(\mu_0, \sigma_0^2, T - \tau) = \frac{e^{-\lambda t_I^*}}{1 - e^{-\lambda t_I^*}} \alpha,$$

wobei t_I^* die Zeit ist, welche eine Trajektorie vom Zustand (μ_I, σ_I^2) nach Γ_+ braucht. Da nun $C(.,., T - \tau)$ in τ wächst und beschränkt ist, folgt die Behauptung. ◇

Bemerkung 6.18

Beachte, dieses gilt nur für $\lambda > 0$.

Da möglichst die optimale Steuerung gefunden werden soll, reicht es im Folgenden, nur konvergente Steuerungen zu betrachten. Aus diesem Grund muss man die Abhängigkeit von t nicht mehr berücksichtigen. Ein Problem, das hierbei auftritt, liegt im Verlust der eindeutigen Zuordnung von A und C . Zur Lösung wird A als Äquivalenzklasse zu einem festen C betrachtet.

Zunächst führt man einige Definitionen ein.

Definition 6.19

Eine Steuerung A heißt *stationär*, wenn für alle $(\mu, \sigma^2) \in \Omega$ gilt:

$$C(\mu, \sigma^2) = \min\{C_N, C_V, C_O\}$$

und für alle $(\mu, \sigma^2) \in \Gamma^+$ gilt:

$$C(\mu, \sigma^2) = \min\{C_V, C_O\}.$$

Definition 6.20

Unter *Randsteuerung* versteht man eine Steuerung A , bei der für alle $(\mu, \sigma^2) \in \Gamma^+$ gilt

$$C(\mu, \sigma^2) = \min\{C_V, C_O\}$$

und sonst $\forall (\mu, \sigma^2) \in \Omega$

$$C(\mu, \sigma^2) = C_N.$$

Bemerkung 6.21

Die Randsteuerung ist „optimal“, insoweit, dass gilt:

$$\|C_A(\mu) - C_{A'}(\mu)\|_{L^\infty} \geq 0.$$

Bemerkung 6.22

Die optimale Steuerung muss eine stationäre Steuerung sein.

Satz 6.23

Es gibt ein λ_0 , sodass für jedes $\lambda > \lambda_0$ gilt, dass jede optimale Steuerung eine Randsteuerung ist. Die Randsteuerung ist eine beschränkte Steuerung.

Für den Beweis dieses Satzes sind erst einige Lemmata zu zeigen.

Lemma 6.24

Im Falle von A^* gilt $\forall (\mu, \sigma^2) \in \Omega$

$$C(\mu, \sigma^2) \leq e^{-\lambda\tau} C_V$$

mit $(\mu - a\tau, \sigma^2 + b\tau) \in \Gamma^+$.

Es sei τ mit $(\mu - a\tau, \sigma^2 + b\tau) \in \Gamma^+$ und auf dem Rand gilt:

$$C(\mu - a\tau, \sigma^2 + b\tau) = \min\{C_V, C_O\} \leq C_V;$$

daraus folgt:

$$C(\mu, \sigma^2) \leq e^{-\lambda\tau} \hat{C}(\mu - a\tau, \sigma^2 + b\tau) \leq e^{-\lambda\tau} C_V.$$

Für die Randsteuerung, wie auch für die optimale Steuerung ergeben sich:

1. $C(\mu) = C(\mu, 0)$ ist beschränkt.
2. $\lim_{\mu \rightarrow \infty} C(\mu) = 0$.

Auf dem Rand Γ^+ gilt:

$$C(\mu, \sigma^2) = \min\{C_V, C_O\}$$

und somit $\hat{C}_N = \min\{C_V, C_O\}$. ◇

Betrachte \hat{C} entlang der Trajektorie $(\mu + at, \sigma^2 - bt)$ mit $(\mu, \sigma^2) \in \Gamma^+$ und $(\mu + a\tau, \sigma^2 - b\tau) \in \Gamma^-$.

Nun hat man noch die zwei Möglichkeiten, auf dem Rand zu untersuchen.

Fall 1: $C_V(\mu, \sigma^2) \leq C_O(\mu, \sigma^2)$

Lemma 6.25

$\forall t \in (0, \tau] : \hat{C}_N(t) := \hat{C}_N(\mu + at, \sigma^2 - bt) < \hat{C}_V(\mu + at, \sigma^2 - bt) =: C_V(t)$

Beweis:

In $t = 0$ gilt $\hat{C}_N(0) = \hat{C}_V(0)$ wegen Definition 6.12.

$$\hat{C}'_N(t) = \frac{d\hat{C}_N(\mu + at, \sigma^2 - bt)}{dt} = -\lambda e^{-\lambda t} C_V(\mu, \sigma^2)$$

$$\hat{C}'_V(t) = \frac{d\hat{C}_V(\mu + at, \sigma^2 - bt)}{dt} = 0$$

$$-\lambda e^{-\lambda t} C_V(\mu, \sigma^2) < 0 \Rightarrow \hat{C}_N(\mu, \sigma) \leq \hat{C}_V(\mu, \sigma).$$

◇

Fall 2: $C_O(\mu, \sigma^2) \leq C_V(\mu, \sigma^2)$

Lemma 6.26

Es existiert ein λ_0 , so dass $\forall \lambda > \lambda_0$ gilt $\forall t \in (0, \tau] :$

$$\hat{C}_N(t) := \hat{C}_N(\mu + at, \sigma^2 - bt) < \hat{C}_O(\mu + at, \sigma^2 - bt) =: C_O(t)$$

Beweis:

In $t = 0$ gilt $\hat{C}_N(0) = \hat{C}_O(0)$ wegen Definition 6.12.

$$\hat{C}'_N(t) = \frac{d\hat{C}_N(\mu + at, \sigma^2 - bt)}{dt} = -\lambda e^{-\lambda t} C_0(\mu, \sigma^2)$$

$$\hat{C}'_O(t) = \frac{d\hat{C}_O(\mu + at, \sigma^2 - bt)}{dt}$$

Somit ist nur folgender Term zu betrachten:

$$\begin{aligned} \left| \frac{d}{dt} \int_{\mathbb{R}} C(\mu') \pi(\mu'; \mu + at, \sigma^2 - bt) d\mu' \right| &\leq \left| \int_{\mathbb{R}} C(\mu') \frac{d}{dt} \pi(\mu'; \mu + at, \sigma^2 - bt) d\mu' \right| \\ &\leq \|C\|_{\infty} \int_{\mathbb{R}} \left| \frac{d}{dt} \pi(\mu'; \mu + at, \sigma^2 - bt) \right| d\mu' \\ &= \|C\|_{\infty} \int_{\mathbb{R}} \left| \frac{d}{dt} \pi(\mu' - \mu; at, \sigma^2 - bt) \right| d\mu' \\ &= \|C\|_{\infty} \int_{\mathbb{R}} \left| \frac{d}{dt} \pi(\mu''; at, \sigma^2 - bt) \right| d\mu'' \\ &\leq \|C\|_{\infty} (Const + Const/\sigma^2) \\ &\leq \|C\|_{\infty} Const. \end{aligned}$$

Der letzte Schritt ist möglich, da σ nach unten durch σ^* von Null weg beschränkt ist.

Daher existiert ein λ_0 mit

$$\forall t \in (0, \tau] : \hat{C}_N(t) := \hat{C}_N(\mu + at, \sigma^2 - bt) < \hat{C}_O(\mu + at, \sigma^2 - bt) =: C_O(t).$$

◇

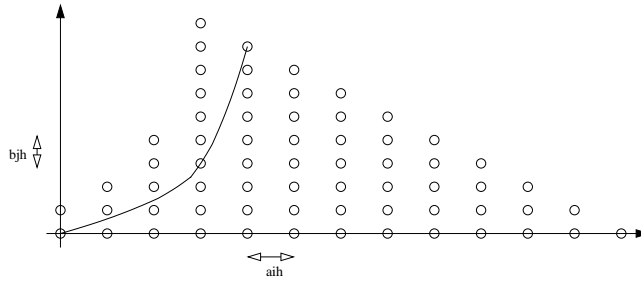
Satz 6.22 folgt direkt aus den Lemmata 6.23-6.25.

6.3 Berechnung einer Steuerstrategie

Zur Berechnung der Randsteuerung bietet sich eine Lineare Dynamische Programmierung an. Hierfür muss zuerst die Unendlichkeit eingeschränkt werden, was man durch die Wahl eines endlichen Gebietes erreicht, beschränkt von Γ_- , $\{(0, \sigma^2)\}$ und $\{(\mu_0 - at, \sigma^2 + bt) | t \geq 0\}$.

Im nächsten Schritt diskretisiert man diese Region und die Zeit wie folgt (siehe Abbildung 6.3):

- Wähle Zeitdiskretisierung h und betrachte das System zu den Zeitpunkten $t_n = T - nh$.

Abbildung 6.3: Diskretisierung des Gebietes $\bar{\Omega}$.

- Diskretisiere die Region entlang den Linien $\{(\mu_0 - at, \sigma^2 + bt)\}$ und betrachte das System an den Punkten $\mu_i = aih, \sigma_j^2 = bjh$.

Nun initialisiert man das System mit folgenden Schritten:

- Identifiziere (μ_I, σ_I^2) mit dem nächsten Gitterpunkt (i_I, j_I) .
- Identifiziere Gitterpunkte I_1 in $\Omega \cup \Gamma^-$.
- Identifiziere Gitterpunkte I_2 außerhalb von $\Omega \cup \Gamma^-$ (linker Rand).
- Initialisiere die diskrete Kostenfunktion $c_{i,j}^n$ für $n = 0$,

$$c_{i,j}^n|_n = 0 = 0.$$

Abschließend iteriert man die folgenden Schritte:

- Berechne mit Hilfe von $c_{i,j}^{n-1}$ die Kostenfunktion für den Zeitschritt n :
 - für „N“ $c_{i,j}^{N,n} = e^{-\lambda h} c_{i-1,j+1}^{n-1}$.
 - für „V“ $c_{i,j}^{V,n} = \alpha + e^{-\lambda h} c_{i_I,j_I}^{n-1}$.
 - für „O“ $c_{i,j}^{O,n} = \beta + \alpha \Phi\left(-\frac{ia}{jb}\right) + \sum_i e^{-\lambda h} c_{i-1,1}^{n-1} \pi_i$
mit $\pi_i = \int_{ih}^{(i+1)h} \pi(\mu', ia, jbh) d\mu'$.
 - $c_{i,j}^n = c_{i,j}^{N,n}$ für $(i,j) \in I_1$.
 - $c_{i,j}^n = \min\{c_{i,j}^{N,n}, c_{i,j}^{V,n}, c_{i,j}^{O,n}\}$ für $(i,j) \in I_2$.

Das Ergebnis eines solchen Vorgehens wird im folgenden Unterkapitel vorgestellt.

6.4 Beispiel Patient 35

Exemplarisch werden an Patient 35 in diesem Unterkapitel sämtliche Schritte durchgeführt.

Der folgende Datensatz ist gegeben, wobei t in Jahren angegeben ist (siehe auch Abbildung 6.4):

		$Ab_{\Delta t_{j-1}}$	$Ab_{\Delta t_j}$	Δt_j
$d_1 =$	$Ab_{\Delta t_i} \quad \Delta t_i$	2800	120	2.89
	2800	120	20	1.09
	4100	670	2.67	
		670	430	0.67
		430	390	1.40

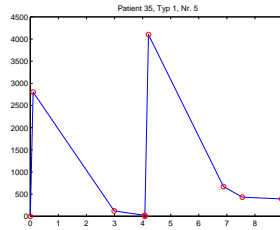


Abbildung 6.4: Patient 35, PTyp1, Nr.5.

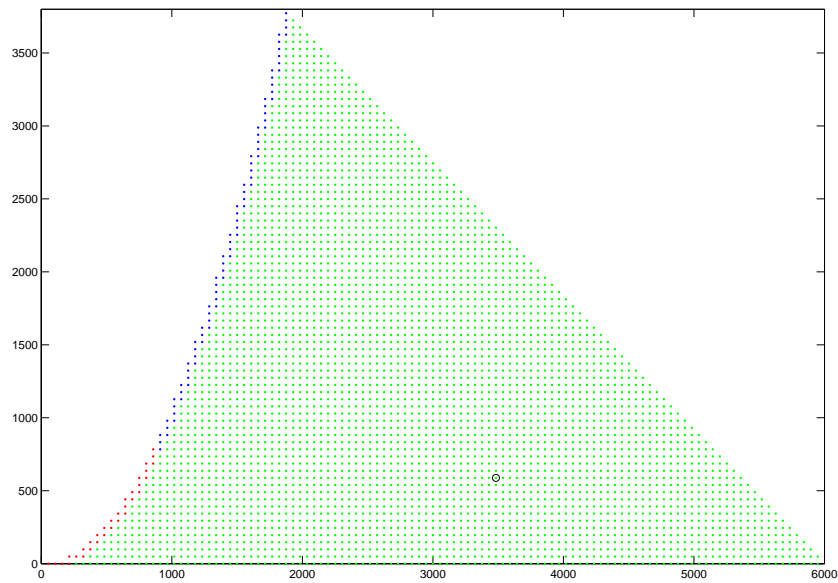
Mit diesen angegebenen Werten berechnen sich die geschätzten Startwerte zu

$$x_0 = \begin{pmatrix} \mu_{I0} \\ \sigma_{I0} \\ a_0 \\ b_0 \end{pmatrix} = \begin{pmatrix} 3511.88 \\ 890.74 \\ 538.10 \\ 632056.51 \end{pmatrix}.$$

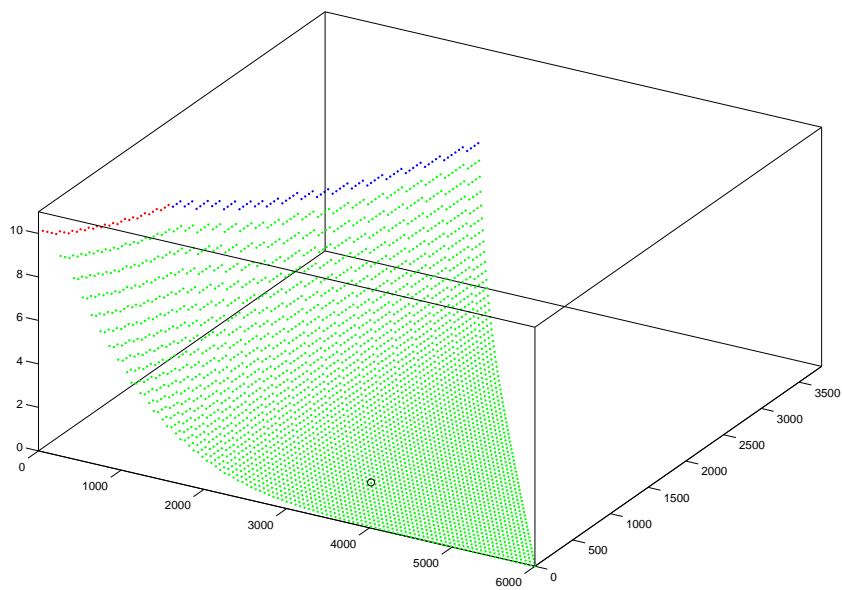
Die identifizierten Parameter lauten dann:

$$\hat{x} = \begin{pmatrix} \hat{\mu}_I \\ \hat{\sigma}_I \\ \hat{a} \\ \hat{b} \end{pmatrix} = \begin{pmatrix} 3496.93 \\ 601.57 \\ 535.91 \\ 489.93 \end{pmatrix}.$$

Führt man mit diesen Werten und den noch zu setzenden Parametern $\alpha = 10$, $\beta = 3$, $\lambda = 1$, $\gamma = 0.5e - 200$ und $\Delta t = 0.05$ die Bestimmung der optimalen Steuerstrategie durch, so erhält man als optimale Steuerung Abbildung 6.5.



(a) Steuerung



(b) Steuerung mit z-Achse = Kosten

Abbildung 6.5: optimale Steuerstrategie für Patient 35, PTyp1, Nr.5., rot = Impfen, blau = Testen, grün = Nichts tun, schwarz = (μ_I, σ_I^2) .

Zusammenfassung

Ziel dieser Arbeit war es, das Potenzial der Mathematik in der immunologischen Forschung aufzuzeigen und anhand dreier Fallbeispiele zu verdeutlichen. In allen Fallstudien diente die Mathematik als Abstraktionswerkzeug. Infolgedessen war es möglich, auf bereits bekannte mathematische Verfahren aus anderen Bereichen zurückzugreifen und diese entsprechend der neuen Anforderungen zu modifizieren und zu ergänzen.

Bei der ersten Fallstudie wurde das Grundmodell der Populationsgenetik auf die „Entwicklung von Mutationen in prekanzerösen Zellen“ übertragen. Die zahlreichen Erkenntnisse über das Populationsmodell konnten so auch für die Untersuchungen zu diesem immunologischen Prozess genutzt werden. Weiterhin war es mit der Scoring-Funktion möglich, die Vielzahl der beobachteten Mutationen für die weiteren Untersuchungen automatisiert und objektiv zu reduzieren und somit auf das derzeitige Leistungsvermögen der Rechner anzupassen. Mit dem Nelder-Mead-Verfahren, das üblicherweise in der Biologie angewendet wird, konnten anschließend die Werte der Modellparameter identifiziert werden.

Bei der zweiten Fallstudie wurde der immunologische Prozess der „Proteinspaltung durch das Enzym Proteasom“ mittels eines Binomialen Ansatzes, der vor allem im Finanz- und Ingenieurwesen zur Anwendung kommt, in ein mathematisches Problem übertragen. Nach Verwendung der Singulärwertzerlegung, die hauptsächlich in der Bildverarbeitung eingesetzt wird, zeigte sich ein direkter Zusammenhang der Singulärwerte sowohl mit den Eingabedaten als auch mit den gesuchten Parametern, der in einem Lemma bewiesen wurde. In der Folge war es möglich, die Daten und Parameter zu reduzieren. Abschließend wurden die Werte der Modellparameter mittels des Levenberg-Marquardt-Verfahrens bestimmt.

Bei der dritten Fallstudie „Immunisierungsstrategie bei Hepatitis in Krankenhäusern“ wurde ein in der Statistik gebräuchliches Verfahren zur Problemformulierung genutzt. Es folgte eine Datenreduzierung, die bisher heuristisch vorgenommen wurde, nun aber mathematisch begründet werden konnte. Hier wurden die Werte der Modellparameter mittels eines Trust-Region-Newton-Verfahrens bestimmt. Die Hauptkomponentenanalyse der identifizierten Parameter deutete darauf hin, dass das Modell weiter vereinfacht werden kann. Die Fallstudie wurde anschließend in ein Kon-

trollproblem überführt. Dieses wurde mittels stochastischer Programmierung, welche ursprünglich aus dem Finanzwesen stammt, analysiert und gelöst. Zum Schluss folgte eine Simulation am konkreten Fall eines Patienten.

Um weiterführende Erkenntnisse zu gewinnen, benötigt man bei den beiden ersten Fallstudien zusätzliche Datensätze, die während der Erstellung dieser Arbeit noch nicht zur Verfügung standen. Bei der dritten Fallstudie steht noch die Verifizierung des Modells aus. Die Bearbeitung aller drei Fallstudien kann als Grundlage für die numerische Simulation herangezogen werden. Die gewonnenen Erkenntnisse wurden von den beteiligten Wissenschaftlern mit Interesse aufgenommen.

Anhang

Anhang: Mutation

Auftretende Mutationen

1. t.X.8.	2. t.X.8.15.	3. der.X.t.X.13.	4. der.X.t.X.15.
5. del.1q.	6. der.1.t.1.19.	7. der.1.t.X.1.t.X.9.	8. der.2.
9. del2p	10. der.2.t.2.3.	11. der.2.t.2.17.	12. der.2.t.2.19.
13. der.3.	14. i.3p.	15. del.3q.	16. der.3.t.3.6.
17. der.4.mar.	18. del.4q.	19. der.4.t.3.4.	20. der.4.t.4.17.
21. der.4.t.4.18.	22. del.5q.	23. der.5.t.4.5.	24. der.5.t.5.17.
25. der.5.t.5.19.	26. der.5.t.5.20.	27. del.6p.	28. del.6q.
29. der.6.t.3.6.	30. der.6.t.6.17.	31. del.8q.	32. der.8.t.X.8.t.X.7.
33. der.8.t.X.8.t.X.8.	34. der.8.t.6.8.	35. der.8.t.8.22.	36. del.9q.
37. der.10.t.X.10.	38. dup.11q.	39. del.11q.	40. der.12.t.3.12.
41. del.13q.	42. der.16.	43. del.16q.	44. der.16.t.16.19.
45. der.17.t.X.17.t.X.8.	46. der.18.t.5.18.	47. del.19p.	48. der.22.t.8.22.

Datensatz (Auszug)

time	cell	t.X.8.	t.X.8.15.	der.X.t.X.13.	der.X.t.X.15.	del.1q.	der.1.t.1.19.	der.1.t.X.1.t.X.9.	der.2.
1	C01	0	0	0	0	0	0	0	0
1	C02	0	0	0	0	0	0	0	0
1	C03	0	0	0	0	0	0	0	0
1	C04	0	0	0	0	0	0	0	0
1	C05	0	0	0	0	0	0	0	0
1	C06	0	0	0	0	0	0	0	1
1	C07	0	0	0	0	0	0	0	0
1	C08	0	0	0	0	0	0	0	0
1	C09	0	0	0	0	0	0	0	1
1	C10	0	0	0	0	0	0	0	0
1	C11	0	0	0	0	0	0	0	0
1	C12	0	0	0	0	0	0	0	0
1	C13	0	0	0	0	0	0	0	0
2	C14	0	0	0	0	0	0	0	0
2	C15	0	1	0	0	0	0	0	0
2	C16	0	0	0	0	0	0	0	0
2	C17	0	0	0	0	0	0	0	0
2	C18	0	0	0	0	0	0	0	0
2	C19	0	0	0	0	0	0	0	0
2	C20	1	0	0	0	0	0	0	0
2	C21	1	0	0	0	0	0	0	0
2	C22	0	0	0	0	0	0	0	0
2	C23	0	0	0	0	0	0	0	0
2	C24	1	0	0	0	0	0	0	0
2	C25	1	0	0	0	0	0	0	0
2	C26	0	0	0	0	0	0	0	0
2	C27	0	0	0	0	0	0	0	0
2	C28	0	0	0	0	0	0	0	0
2	C29	1	0	0	0	0	0	0	0

Anhang: Proteasom

Datensatz Enolase (Auszug)

Fragment Ende	Fragment Anfang	Aminosäurenpositionen	Fragment	Häufigkeit [pmol]
5	0	1-5	AVSKV	70
7	0	1-7	AVSKVYA	100
8	0	1-8	AVSKVYAR	80
9	0	1-9	AVSKVYARS	10
10	0	1-10	AVSKVYARSV	20
10	5	6-10	YARSV	50
23	8	9-23	SVYDSRGNPTVEVEL	15
19	10	11-19	YDSRGNPTV	40
20	10	11-20	YDSRGNPTVE	15
23	10	11-23	YDSRGNPTVEVEL	50
31	23	24-31	TTEKGVFR	50
38	31	32-38	SIVPSGA	30
44	31	32-44	SIVPSGASTGVHE	50
46	37	38-46	ASTGVHEAL	25
48	37	38-48	ASTGVHEALEM	25
48	38	39-48	STGVHEALEM	10
56	48	49-56	RDGDKSKW	50
62	56	57-62	MGKGVV	20
63	56	57-63	MGKGVVH	40
79	64	65-79	VKNVNDVIAPAFVKA	5
75	66	67-75	NVNDVIAPA	25
78	66	67-78	NVNDVIAPAFVK	25
79	66	67-79	NVNDVIAPAFVKA	20

Datensatz Prion (Auszug)

Fragment Anfang	Fragment Ende	Häufigkeit [pmol]
24	0	775
24	28	1
24	32	15
24	36	115
24	37	40
24	41	95
24	48	5
24	52	420
24	60	120
24	68	260
31	0	50
31	55	80
31	59	30
33	41	1
34	56	5
35	0	70
35	41	35
35	44	3
35	46	3
35	52	100
37	52	20
38	0	20
42	0	100
42	52	40

Anhang: Immunisierungsstrategie

Datensatz (Auszug)

Patientnummer	Datum1	Impfung	Datum2	Titer
1	11.02.1993	1	11.02.1993	.
1	11.02.1993	0	17.03.1993	1900
1	11.02.1993	0	19.10.1994	300
1	11.02.1993	0	14.11.1996	60
2	02.02.1989	1	02.02.1989	.
2	02.02.1989	0	06.03.1989	4100
2	02.02.1989	0	15.03.1993	130
2	02.02.1989	0	06.03.1995	90
3	22.04.1993	1	22.04.1993	.
3	22.04.1993	0	25.05.1993	330
3	22.04.1993	0	11.04.1994	54
3	22.04.1993	0	07.11.1994	13
4	30.09.1993	1	30.09.1993	.
4	30.09.1993	0	03.11.1993	370
4	30.09.1993	0	25.04.1994	35
4	30.09.1993	0	12.11.1997	65
5	28.06.1994	1	28.06.1994	.
5	28.06.1994	0	27.07.1994	500
5	28.06.1994	0	03.08.1995	98
5	28.06.1994	0	14.03.1996	96
5	28.06.1994	0	07.11.1996	50
6	25.11.1993	1	25.11.1993	.
6	25.11.1993	0	04.01.1994	1600
6	25.11.1993	0	19.09.1994	70
6	25.11.1993	0	04.10.1995	53
7	06.05.1991	1	06.05.1991	.
7	06.05.1991	0	06.06.1991	1300
7	06.05.1991	0	27.03.1992	140
7	06.05.1991	0	06.12.1993	30
7	06.05.1991	0	21.11.1995	25
7	06.05.1991	0	08.04.1997	15
8	02.12.1993	1	02.12.1993	.
8	02.12.1993	0	11.01.1994	600
8	02.12.1993	0	02.01.1996	66
8	02.12.1993	0	07.10.1996	54
8	02.12.1993	0	04.08.1997	66
9	08.02.1994	1	08.02.1994	.
9	08.02.1994	0	31.08.1994	200
9	08.02.1994	0	04.12.1995	113
9	08.02.1994	0	19.03.1997	78
10	17.02.1992	1	17.02.1992	.
10	17.02.1992	0	17.03.1992	430
10	17.02.1992	0	24.06.1992	2000
10	17.02.1992	0	19.10.1995	55
10	17.02.1992	0	03.06.1996	110

Weiterer Ansatz zur Bestimmung einer Kontrollstrategie

Das Bestimmen der optimalen Steuerung lässt sich auch als Partielle Differentialgleichung schreiben, wodurch sich weitere Analysemöglichkeiten bieten. In dieser Arbeit wird jedoch nur der Ansatz vorgestellt und kein Versuch hinsichtlich weiterführender Untersuchungen unternommen (siehe hierfür [TRO05]).

PDE für die optimale beschränkte Steuerung:

Satz 6.27

Die Kostenfunktion für die optimale beschränkte Steuerung erfüllt in Ω

$$(a\partial_\mu - b\partial_{\sigma^2} + \partial_\tau)C(\mu, \sigma^2, T - \tau) = -\lambda C(\mu, \sigma^2, T - \tau)$$

mit der Startbedingung $C(\mu, \sigma^2, T) = 0$ und den Randbedingungen

$$C(\mu, \sigma^2, T - \tau) = \min \left\{ \begin{array}{l} \alpha + C(\mu_I, \sigma_I^2, T - \tau), \\ \beta + \Phi(-\mu/\sigma)(\alpha + C(\mu_I, \sigma_I^2, T - \tau)) \\ + \int_0^\infty C(\mu', 0, T - \tau)\pi(\mu'; \mu, \sigma^2)d\mu' \end{array} \right\}$$

für $(\mu, \sigma^2) \in \Gamma_+$.

Satz 6.28

Es gibt ein λ_0 , so dass für $\lambda > \lambda_0$ diese Gleichung eine eindeutige Lösung hat.

Satz 6.29

Es gibt ein λ_0 , so dass für $\lambda > \lambda_0$ eine optimale beschränkte Steuerung existiert.

Bemerkung 6.30

Diese Sätze wurden durch die Sätze in Kapitel 6 bewiesen.

Für die Analyse der PDE stehen einem nun die folgenden zwei Arten zur Verfügung:

1.Art:

$$\begin{aligned} (a\partial_\mu - b\partial_{\sigma^2} + \partial_\tau)C(\mu, \sigma^2, T - \tau) &= -\lambda C(\mu, \sigma^2, T - \tau) \\ ((a)\frac{\partial}{\partial\mu} + (-b)\frac{\partial}{\partial\sigma^2} + (1)\frac{\partial}{\partial\tau})C(\mu, \sigma^2, T - \tau) &= (-\lambda)C(\mu, \sigma^2, T - \tau) \\ \frac{dC}{dt} = \left(\frac{\partial\mu}{\partial t}\frac{\partial}{\partial\mu} + \frac{\partial\sigma^2}{\partial t}\frac{\partial}{\partial\sigma^2} + \frac{\partial\tau}{\partial t}\frac{\partial}{\partial\tau}\right)C(\mu, \sigma^2, T - \tau) &= (-\lambda)C(\mu, \sigma^2, T - \tau). \end{aligned}$$

Die Charakteristiken hierzu sind:

$$\begin{aligned} \dot{\mu} = a &\Rightarrow \mu = a \cdot t \\ \dot{\sigma} = -b &\Rightarrow \sigma = -b \cdot t \\ \dot{\tau} = 1 &\Rightarrow \tau = t \\ \dot{C} = -\lambda C &\Rightarrow C = C_0 e^{-\lambda t}. \end{aligned}$$

Bemerkung 6.31

Die Zeit geht hier in die Vergangenheit.

2.Art:

$$\begin{aligned} (-a\partial_\mu + b\partial_{\sigma^2} - \partial_\tau)C(\mu, \sigma^2, T - \tau) &= \lambda C(\mu, \sigma^2, T - \tau) \\ ((-a)\frac{\partial}{\partial\mu} + (b)\frac{\partial}{\partial\sigma^2} + (-1)\frac{\partial}{\partial\tau})C(\mu, \sigma^2, T - \tau) &= (\lambda)C(\mu, \sigma^2, T - \tau) \\ \frac{dC}{dt} = \left(\frac{\partial\mu}{\partial t}\frac{\partial}{\partial\mu} + \frac{\partial\sigma^2}{\partial t}\frac{\partial}{\partial\sigma^2} + \frac{\partial\tau}{\partial t}\frac{\partial}{\partial\tau}\right)C(\mu, \sigma^2, T - \tau) &= (\lambda)C(\mu, \sigma^2, T - \tau). \end{aligned}$$

Die Charakteristiken hierzu sind:

$$\begin{aligned} \dot{\mu} = -a &\Rightarrow \mu = -a \cdot t \\ \dot{\sigma} = b &\Rightarrow \sigma = b \cdot t \\ \dot{\tau} = -1 &\Rightarrow \tau = -t \\ \dot{C} = \lambda &\Rightarrow C = C_0 e^{\lambda t}. \end{aligned}$$

Bemerkung 6.32

Die Zeit geht hier in die Zukunft.

Glossar

Folgende Erläuterungen und Definitionen wurden [WIK06] entnommen.

A

Aminosäuren

Aminosäuren bzw. Aminocarbonsäuren sind eine Klasse kleiner organischer Moleküle mit mindestens einer Carboxylgruppe ($-COOH$) und mindestens einer Aminogruppe ($-NH_2$). Der Begriff Aminosäure (veraltet Amidosäure) wird synonym für die proteinogenen Aminosäuren verwendet, die die Bausteine der Proteine sind. Von den proteinogenen Aminosäuren sind bisher 23 bekannt. Das Spektrum der Klasse der Aminosäuren geht aber weit über diese hinaus. So sind bisher 250 nicht-proteinogene Aminosäuren bekannt, die biologische Funktionen haben (Stand: Februar 2006). Die Anzahl der synthetisch erzeugten und die der theoretisch möglichen Aminosäuren ist noch erheblich größer.

C

Chromosom

Ein Chromosom (v. griech.: Chroma= Farbe und Soma=Körper, also Farbkörperchen oder anfärbbares Körperchen) ist die Verpackungseinheit für ein Molekül der DNA und somit die stoffliche Grundlage für die Gene. Das Material, aus dem die Chromosomen bestehen, wird auch als Chromatin bezeichnet.

Cytogenetik

Cytogenetik ist die Forschungsrichtung, welche sich mit den (menschlichen) Chromosomen beschäftigt.

Cytotoxische T-Zellen

Cytotoxische T-Zellen (CTL), auch T-Killerzellen oder nur kurz $CD8^+$ -Zellen, tragen typischerweise $CD8\alpha\beta$ -Heterodimere an ihrer Oberfläche und erkennen Antigene, die ihnen von allen kernhaltigen Zellen auf MHC-I-Molekülen dargeboten werden. Daher spielen sie vor allem in der Erkennung und Beseitigung von viral infizierten Zellen eine Rolle.

E

Enolase

Die neuronenspezifische Enolase (NSE), engl.: neuron-specific enolase, ist ein Enzym (Biokatalysator) des Glukose-Stoffwechsels. Sie kommt in verschiedenen Isoformen in den Nervenzellen (Neuronen) des Gehirns und des peripheren Nervengewebes sowie in neuroendokrinen Geweben, v.a. in den sog. APUD-Zellen, vor.

F

FISH (Fluoreszenzinsituhybridation)

Eine Variante der In situ-Hybridisierung, bei der die Sonde mit Hilfe eines fluoreszierenden Farbstoffes nachgewiesen wird.

G

Genotyp

Der Genotyp oder das Erbbild eines Organismus repräsentiert seine exakte genetische Ausstattung, also den individuellen Satz von Genen, den er im Zellkern in sich trägt. Der Begriff Genotyp wurde 1909 von dem dänischen Genetiker Wilhelm Johannsen geprägt.

H

Hepatitis

Als Hepatitis (Plural: Hepatitiden; von griech. Leber) wird eine Entzündung der Leber bezeichnet, für die zahlreiche Ursachen angeführt werden können. Gibt es eine andere zugrundeliegende Erkrankung zu dieser Leberentzündung, wird von einer sogenannten Begleithepatitis gesprochen.

HPLC

HPLC steht für High Performance Liquid Chromatography (in den Anfangszeiten dieser Technik auch für High Pressure Liquid Chromatography), eine analytische Methode in der Chemie. Die HPLC ist eine Methode, mit der man nicht nur Substanzen trennt sondern diese auch über Standards identifizieren und quantifizieren (die genaue Konzentration bestimmen) kann. Das deutsche Wort Hochleistungs-Flüssigchromatographie wird nur sehr selten verwendet.

I

In situ-Hybridisierung

Die In situ-Hybridisierung (ISH) ist ein Verfahren, um unter anderem die mRNA in einem Gewebe sichtbar zu machen. Es werden bei einer solchen Färbung nur diejenigen Zellen angefärbt (hybridisiert), in denen das zu untersuchende Gen aktiv ist und in denen daher die mRNA im Cytoplasma vorliegt. Dieses Färbeverfahren findet besonders in der Entwicklungsbiologie Anwendung. Hier ist es von besonderem In-

teresse, die Aktivität eines Gens beispielsweise während der Embryogenese in situ zu verfolgen. Das embryonale oder auch adulte Gewebe muss für die Färbung zunächst fixiert werden; die Aktivität kann daher nicht in Echtzeit verfolgt werden sondern ist nur eine Momentaufnahme des Zustands, in dem sich das Gewebe befand, als es fixiert wurde.

M

M-FISH

Eine verfeinerte Variante von FISH.

MHC-I-Molekülen

Der Haupthistokompatibilitätskomplex (Abk. MHC von engl.: Major Histocompatibility Complex) umfasst eine Gruppe von Genen bei Wirbeltieren, die für Proteine codiert sind, welche für die Immunerkennung, die Gewebeverträglichkeit (Histokompatibilität) bei Transplantationen und die immunologische Individualität wichtig sind. MHC-Regionen finden sich in allen Wirbeltieren ab den Knorpelfischen (Haie, Rochen). Beim Menschen sind diese Gene auf dem kurzen Arm von Chromosom 6 zu finden. Die Genprodukte, die MHC-Moleküle, sind körpereigene Antigene auf der Oberfläche jeder Körperzelle, die immunologische Vorgänge regulieren. Diese Moleküle kennzeichnen die Zellen als zum Körper gehörig und sind auf weißen Blutkörperchen (Leukozyten) leicht nachweisbar. Daher kommt auch die Bezeichnung HLA-System (Human Leucocyte Antigen) für das Regulationssystem der Immunabwehr, dessen wichtigster Bestandteil der MHC ist.

Unter anderem werden im MHC-Komplex die MHC-Klasse-I- und MHC-Klasse-II-Moleküle codiert, die für die einwandfreie Funktion des Immunsystems eine zentrale Rolle spielen.

P

Prion

Bei Prionen (von engl. proteinaceous infectious particle) handelt es sich um bestimmte Proteine, welche im menschlichen wie auch im tierischen Organismus natürlicherweise vorkommen. Einige Formen, welche nicht durch den Organismus selbst korrigiert werden können, sind jedoch schädlich für diesen. Sie sind zum Beispiel für die Creutzfeldt-Jakob-Krankheit beim Menschen, BSE (Rinderwahnsinn) beim Rind oder Scrapie bei Schafen verantwortlich. Das pathogene Prion gelangt entweder durch kontaminierte Nahrung (dieser Infektionsweg ist am wahrscheinlichsten; andere Infektionswege wie etwa die Schmierinfektion konnten aber noch nicht ausgeschlossen werden) in den Körper (z.B. bei BSE, Kuru) oder es entsteht durch die spontane Umfaltung körpereigener Prione (z.B. Creutzfeldt-Jakob-Krankheit, familiäre Schlaflosigkeit). Grundsätzlich sind pathogene Prione von anderen Krankheitserregern wie Viren, Bakterien oder Pilzen zu unterscheiden, da sie kein eigenes Erbmateriale enthalten. Sie sind nicht nur von großem wissenschaftlichen Interesse

sondern hatten durch die BSE-Krise und die neue Variante der Creutzfeldt-Jakob-Krankheit auch starke Auswirkungen auf Gebiete wie Landwirtschaft, Verbraucherschutz, Medizin und Politik.

Proteasom

Das Proteasom ist ein 700 kDa schwerer Proteinkomplex, der im Cytoplasma und im Zellkern Proteine zu Fragmenten abbaut.

T**Titer (Medizin)**

Der Titer ist in der Biologie und Medizin eine Konzentrationsangabe z.B. eines Antikörpers, Antigens oder eines Virus, die durch eine fortlaufende Verdünnung einer Ausgangslösung bestimmt wird. Auf die Verdünnungsreihen wird dann ein bestimmter Test (z.B. Immunoassay) angewendet und jene Verdünnung, bei der noch eine Reaktion nachweisbar ist, als Titer angegeben. Die Ermittlung des Titers war früher eine übliche Methode um, z.B. eine Immunität nach einer Impfung oder den Anstieg von Antikörpern während einer akuten Infektionskrankheit zu beurteilen.

T-Lymphozyten

T-Lymphozyten oder kurz T-Zellen sind eine für die Immunabwehr wichtige Gruppe von Blutzellen. Es handelt sich bei ihnen um eine Subpopulation der weißen Blutkörperchen (Leukozyten). T-Lymphozyten sind neben B-Lymphozyten an der adaptiven Immunantwort beteiligt.

Literaturverzeichnis

- [ALT02] Alt, W. *Nichtlineare Optimierung*, Wiesbaden: Vieweg, 2002.
- [AND99] Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide, Third Edition*, Philadelphia: SIAM, 1999.
- [ARN00] Arnold, B.: *Singular Value Decomposition Image Compression System for Automatic Object Recognition*, University of Canterbury, Department of Mathematics and Statistics, Report, 2000.
- [BOC98] Bock, R. K., Krischer, W.: *The Data Analysis BriefBook*, New York: Springer Verlag, 1998.
- [BOG89] Bogacki, P. and L. F. Shampine: *A 3(2) pair of Runge-Kutta formulas*, Appl. Math. Letters, **2**: 1 – 9, 1989.
- [BUE00] Bürger, R., *The Mathematical Theory of Selection, Recombination, and Mutation.*, Chichester, New York: Wiley Verlag, 2000.
- [CHA04] Chang, F.-R.: *Stochastic Optimization in Continuous Time*, Cambridge: Cambridge University Press, 2004.
- [CLO00] Clote, P., Backofen, R.: *Computational Molecular Biologie*, Chichester, New York: Wiley Verlag, 2000.
- [CRO70] Crow, J. F. and Kimura, M., *An Introduction to Population Genetics Theory*. Harper & Row, 1970.
- [FAH97] Fahrmeir, L., Künstler, R., Pigeot, I., Tutz, G.: *Statistik, Der Weg zur Datenanalyse*, Berlin: Springer, 1997.
- [FAU04] Fauth, C., O'Hare, MJ., Lederer, G., Jat, PS., Speicher, MR.: *Order of genetic events is critical determinant of aberrations in chromosome count and structure*, Genes Chromosomes Cancer, **40**: 298 – 306, 2004.

- [GIL96] Gilks, W.R., Richardson, S., Spiegelhalter, D.J., *Markov chain Monte Carlo in practice*, London: Chapman und Hall, 1996.
- [GOL96] Golub G. H., van Loan C. F.: *Matrix Computations*, Baltimore, London: The Johns Hopkins University Press, 1996.
- [HAD04] Hadeler, K.P., Kuttler, C., Nussbaum, A. K.: *Cleaving proteins for the immune system*, *Mathematical Biosciences*, **188**: 63 – 79, 2004.
- [HER02] Hermisson, J. Redner, O., Wagner, H. and Baake, E.: *Mutation–Selection Balance: Ancestry, Load, and Maximum Principle*, *Theor. Pop. Biol.*, **62**: 9 – 46, 2002.
- [HIG02] Higham, D. J.: *Nine Ways to Implement the Binomial Method for Option Valuation in MATLAB*, *SIAM Review* 44, **4**: 661 – 677, 2002.
- [HOF85] Hofbauer, J.: *The selection mutation equation*, *J. Math. Biol*, **23**: 41 – 53, 1985.
- [KAL95] Kall, P., Wallace S.W.: *Stochastic Programming*, Chichester: John Wiley and Sons, 1995.
- [KUH80] Kuhlmann, W.: *Parameterschätzung unter Eingleichungsmodellen im unbeschränkten Parameterraum*, aus „Arbeiten zur Angewandten Statistik“ von K. - A. Schäffer, Würzburg: Physica Verlag, 1980.
- [KUT00] Kuttler, C., Nussbaum, A. K., Dick, T. P., Rammensee, H.-G., Schild, H., Hadeler, K.-P.: *An algorithm for the prediction of proteasomal cleavages*, *J. Mol. Biol.*, **298**: 417 – 429, 2000.
- [LAG98] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright. *Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions*, *SIAM Journal of Optimization*, **9**: 112 – 147, 1998.
- [LOA97] van Loan, C. F.: *Introduction to scientific computing*, New Jersey: MATLAB Curriculum Series, 1997.
- [MAT03] Matlab 6 R13 Documentation: *MATLAB functions + Optimization Toolbox*, 2003.
- [MUE02] Müller, J., Schönfisch, B., Nussbaum, A. K.: *Modeling Proteasomal Protein Degradation*, Preprint GSF, PP02 – 033, 2002.
- [NUS98] Nussbaum, A. K., Dick, T. P., Keilholz, W., Schirle, M., Stevanovic, S., Dietz, K., Heinemeyer, W., Groll, M., Wolf, D. H., Huber, R., Rammensee, H.-G., Schild, H.: *Cleavage motifs of the yeast 20S proteasome β subunits deduced from digests of enolase*, *Proc. Natl. Acad. Sci, USA*, **95**: 12504 – 12509, 1998.

- [PAM98] Pamer, E., Cresswell, P.: *Mechanisms of MHC class I-restricted antigen processing*, Annu. Rev. Immunol. **16**: 323 – 358, 1998.
- [RAM93] Rammensee, H.G., Falk, K., Rotschke, O.: *Peptides naturally presented by MHC class I molecules*, Annu. Rev. Immunol., **11**: 213 – 244, 1993.
- [SAL96] Sall, J., Lehman, A.: *JMP, Start Statistic*, SAS Institute Inc., 1996.
- [SPE93] Spellucci, P.: *Numerische Verfahren der nichtlinearen Optimierung*, Berlin: Birkhäuser, 1993.
- [SPE96/2] Speicher, M.R., Ballard, S.G., Ward, D.C.: *Computer image analysis of combinatorial multi-flour FISH*, Bioimages, **4**: 52 – 64, 1996.
- [SPE96] Speicher, M.R., Ballard, S.G., Ward, D.C.: *Carrotying human chromosomes by combinatorial multi-flour FISH*, Nature Genet, **12**: 368 – 375, 1996.
- [TOE01] Toes, R. E. M., Nussbaum, A. K., Degermann, S., Schirle, M., Emmerich, N., Kraft, M., Laplace, C., Zwinderman, A., Dick, T., Müller, J., Schönfish, B., Schmid, C., Fehling, H.-J., Stevanovic, S., Rammensee, H.-G., Schild, H.: *Discrete cleavage motifs of constitutive and immunoproteasomes revealed by quantitative analysis of cleavage products*, J. Exp. Med., **194**: 1 – 12, 2001.
- [TRO05] Tröltzsch, F.: *Optimale Steuerung partieller Differentialgleichungen*, Wiesbaden: Vieweg, 2005.
- [WHI96] Whittle, P.: *Optimal Control, Basics and Beyond*, Chichester: John Wiley and Sons, 1996.
- [WIK06] Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 9. November 2006, 10:42 UTC. URL: <http://de.wikipedia.org>.

