

**Bachelorarbeit**

**Fancy thesis template**

Vorname Nachname  
Matrikelnummer: 0123456  
Angewandte Informatik (Bachelor)

**UNIVERSITÄT**  
**D U I S B U R G**  
**E S S E N**

Fachgebiet Verteilte Systeme, Abteilung Informatik  
Fakultät für Ingenieurwissenschaften  
Universität Duisburg-Essen

21. Februar 2023

**Erstgutachter:** Prof. Dr-Ing. Torben Weis  
**Zweitgutachter:** Hier Zweitgutachter eintragen  
**Zeitraum:** 1. September 2042 - 1. Januar 2043



# Abstract

An abstract is a brief summary of a research article, thesis, review, conference proceeding, or any in-depth analysis of a particular subject and is often used to help the reader quickly ascertain the paper's purpose. When used, an abstract always appears at the beginning of a manuscript or typescript, acting as the point-of-entry for any given academic paper or patent application. Abstracting and indexing services for various academic disciplines are aimed at compiling a body of literature for that particular subject.

<sup>1</sup>

---

<sup>1</sup>Wikipedia: [https://en.wikipedia.org/wiki/Abstract\\_\(summary\)](https://en.wikipedia.org/wiki/Abstract_(summary))



# Contents

<b>1 Grundlagen</b>	<b>1</b>
1.1 Markov Prozesse . . . . .	1
1.2 Hidden Markov Modelle . . . . .	2
1.3 Baum-Welch Algorithmus . . . . .	4
1.4 Underflow und Overfitting . . . . .	5
1.5 Metaheuristische Algorithmen . . . . .	5
1.6 Funktionsweise . . . . .	6
1.6.1 Kritik an Metaheuristischen Algorithmen . . . . .	6
1.7 Genetische Algorithmen . . . . .	7
1.7.1 Selektionsoperator . . . . .	8
1.7.2 Crossoveroperator . . . . .	9
1.7.3 Mutationsoperator . . . . .	11
1.7.4 Ersetzen . . . . .	11
1.7.5 Ablauf eines genetischen Algorithmus . . . . .	11
<b>2 Trainieren von HMMs mit einem GA</b>	<b>15</b>
2.1 Konstruktion eines GAHMM . . . . .	15
2.2 Literaturübersicht . . . . .	16
<b>3 Evaluation</b>	<b>17</b>
3.1 Vergleich genetischer Operatoren . . . . .	17
3.2 Berechnungskost eines GA . . . . .	17
<b>Bibliography</b>	<b>19</b>



# Chapter 1

## Grundlagen

### 1.1 Markov Prozesse

Markovketten wurden 1906 von Andrei Andreyvich Markov eingeführt und genießen seit jeher große Beliebtheit in vielen Wissenschaftlichen Disziplinen wie Biologie, Sozialwissenschaften, Psychologie und Physik . [5] Das besondere an Markovketten ist, dass sie trotz ihrer simplen Struktur in der Lage sind sehr komplexe Prozesse zu modellieren. Im folgenden werden wir uns auf zeitdiskrete homogene Markovketten beschränken.

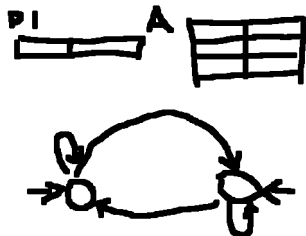
Ein zeitdiskreter homogener Markov Prozess modelliert ein zeitdiskretes Signal mit einer endlichen Menge an Zuständen. Dieser Prozess wird beschrieben durch einen Startwahrscheinlichkeitsvektor  $\pi$  und eine Transitionsmatrix  $A$ .  $\pi_i$  ist die Wahrscheinlichkeit sich in Zeitpunkt  $t = 0$  in Zustand  $S_i$  zu befinden.  $A_{i,j}$  ist die Wahrscheinlichkeit, dass der Prozess sich im nächsten Zeitpunkt in Zustand  $S_j$  befindet, wenn er sich im gegenwärtigen Zeitpunkt im Zustand  $S_i$  befindet. In anderen Worten ist  $A_{i,j}$  also die Wahrscheinlichkeit einer Transition von Zustand  $S_i$  nach Zustand  $S_j$ . Man beobachtet, dass die Transitionswahrscheinlichkeiten unabhängig von  $t$  sind, so dass der nächste Zustand einzig und allein vom gegenwärtigen Zustand des Prozesses abhängt. Diese Eigenschaft ist bekannt als Markov-Eigenschaft oder Markov-Bedingung. Das Ereignis in irgendeinem Zustand zu starten, so wie das Ereignis von einem gegebenen Zustand in irgendeinen anderen Zustand zu wechseln sind sichere Ereignisse. Somit gelten die Bedingungen, dass die Summe der Startwahrscheinlichkeiten  $\pi$  und die Summe der ausgehenden Transitionen für jeden Zustand  $A_i$  1 ergeben müssen.

$$\sum_{i=0}^N \pi_i = 1 \quad \sum_{j=0}^N A_{i,j} = 1$$

Um das ganze an einem Beispiel zu verdeutlichen stellen wir uns einen Markovprozess vor, welcher die durchschnittliche jährliche Temperatur modelliert. Um das Beispiel übersichtlich zu halten beschränken wir uns auf die Zustände Heiß und Kalt.

Die Startwahrscheinlichkeiten  $\pi$  entsprechen den Wahrscheinlichkeiten, dass ein zufälliger Tag heiß beziehungsweise kalt ist. Die Transitionswahrscheinlichkeiten seien gegeben

durch  $A$ . Solch ein Markov Prozess kann anschaulich als gerichteter Transitionsgraph representiert werden.



Mit diesem Modell können wir nun zum Beispiel berechnen, was die Wahrscheinlichkeit ist die Temperaturfolge Heiß, Kalt, Heiß, Kalt, Kalt zu beobachten unter der Voraussetzung, dass die gegenwärtige Temperatur Heiß ist.  $O = \{S_1, S_2, S_1, S_2, S_2\}$   $P(O \mid Model, q_1 = S_1) = P(q_1 = S_1 \mid Model) \wedge P(S_1 \mid S_1) \wedge P(S_2 \mid S_1) \wedge P(S_1 \mid S_2) \wedge P(S_2 \mid S_1) \wedge P(S_2 \mid S_2) = \pi_1 \cdot A_{1,1} \cdot A_{1,2} \cdot A_{2,1} \cdot A_{1,2} \cdot A_{2,2} = 0.8 \cdot 0.4 \cdot 0.3 \cdot 0.7 \cdot 0.5 \cdot 0.3 \cdot 0.9 = 0.009072$

Allgemeiner können wir eine beliebige Observations-Sequenz  $O$  berechnen mit  $O = \{O_1, O_2, \dots, O_{T-1}, O_T\}$   $P(O \mid Model) = \pi_{O_1} \cdot \prod_{t=2}^T A_{O_{t-1}, O_t}$

Mit einer Markovkette können wir jedoch nur Prozesse modellieren welche wir messen können. Wenn wir einen Prozess  $X$  haben den wir nicht beobachten können und einen beobachtbaren Prozess  $Y$  welcher von  $X$  beeinflusst wird, können wir  $X$  anhand der von  $Y$  generierten Beobachtungen mit einem Hidden Markov Model modellieren.

## 1.2 Hidden Markov Modelle

Angenommen wir möchten die durchschnittlichen jährlichen Temperaturen im Mittelalter modellieren, also für die Jahre 500 bis 1500. In diesem Zeitraum gab es noch keine Temperaturaufzeichnungen und wir können die Temperatur somit nicht direkt beobachten. Es ist jedoch bekannt, dass sich die Temperatur auf das Wachstum von Bäumen auswirkt. Warme Jahre führen in der Regel zu breiteren Jahresringen und kalte Jahre zu schmaleren. Die Jahresringe von Bäumen für den gewünschten Zeitraum können wir beobachten. Seien  $\Sigma = \{S, M, L\}$  die Symbole für die verschiedenen Breiten an Jahresringen welche beobachtet werden können.  $S$  steht für dünne,  $M$  für mittlere und  $L$  für breite Jahresringe. Unser Modell aus dem vorherigen Abschnitt wird nun um eine Emissionsmatrix  $B$  erweitert.  $B_{i,k}$  ist die Wahrscheinlichkeit in Zustand  $S_i$  Emissions-Symbol  $k$  zu beobachten. Zum Beispiel gibt  $B_{1,2}$  die Wahrscheinlichkeit an in in einem Heißen Jahr Jahresringe mittlerer Breite zu beobachten. Analog zu  $A$  und  $\pi$  sind auch die Reihen von  $B$  stochastisch.  $\sum_{j=0}^N B_{i,j} = 1$  Die Werte der Parameter  $\pi$ ,  $B$  und  $A$  gewinnen wir aus den Temperaturaufzeichnungen und Jahresringen der Jahre 1900 bis 2000.



Ein Mögliches Problem welches wir für dieses Hidden Markov Model stellen können ist die Berechnung der Wahrscheinlichkeit einer Observationssequenz unter einem gegebenen Modell. Zum Beispiel Was ist die Wahrscheinlichkeit 5 aufeinanderfolgende Jahre mit dicken Baumringen zu beobachten? Formell formuliert  $P(L, L, L, L, L \mid \lambda)$ . Wenn wir die Abfolge der Zustände  $Q = q_1, q_2, q_3, q_4, q_5$  bereits kennen lässt sich  $P(O \mid \lambda)$  sehr einfach berechnen. Die Wahrscheinlichkeit ein Symbol  $k$  zu beobachten unter der Bedingung in Zustand  $S_i$  zu sein ist nichts anderes als  $B_{i,k}$ . Die Wahrscheinlichkeit einer Observationssequenz für ein gegebenes Modell und eine gegebene Abfolge von Zuständen ist also  $P(O \mid Q, \lambda) = \prod_{t=1}^T b_{q_t}(O_t)$ . Wenn wir die Abfolge der Zustände jedoch nicht kennen gestaltet sich die Berechnung etwas komplizierter. Die Wahrscheinlichkeit von  $P(O \mid \lambda)$  ist equivalent zu der Summe der bedingten Wahrscheinlichkeiten aller möglichen Zustandsfolgen.  $P(O \mid \lambda) = \sum_{all Q} P(O \mid Q, \lambda)$ . Es ist möglich  $P(O \mid \lambda)$  so zu berechnen, jedoch wächst die Anzahl der Möglichen Zustandsabfolgen  $Q$  exponentiell in Abhängigkeit von der Länge der Observationssequenz  $T$ . Präzise gesagt gilt  $|Q| = 2T \cdot N^T$ . Selbst wenn  $N$  und  $T$  kleine Werte einnehmen ist die benötigte Rechenleistung untragbar. Für  $N=5$  und  $T=100$  müssten bereits  $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$  Berechnungen durchgeführt werden. Um diese Zahl in Relation zu stellen,  $10^{72}$  ist mindestens 3 mal mehr als 1000 und 1000 ist schon ziemlich groß. Zum Glück gibt es eine effiziente Methode um  $P(O \mid \lambda)$  zu berechnen, die Forwardvariable.

## Die Forwardvariable

Sei  $\alpha_t$  die Forwardvariable folgendermaßen definiert  $\alpha_t(i) = P(O_1 O_2 \dots O_t \wedge q_t = S_i \mid \lambda)$ .  $\alpha_t(i)$  Beschreibt also die Wahrscheinlichkeit die partielle Observationssequenz  $O_1 O_2 \dots O_t$  zu beobachten und in Zeitpunkt  $t$  in Zustand  $i$  zu sein.  $\alpha_t(i)$  kann folgendermaßen induktiv berechnet werden. Für  $t = 0$  lässt sich  $\alpha_0(i)$  aus  $\pi$  und  $B$  berechnen, da noch keine Transition stattgefunden hat.  $\alpha_0(i) = \pi_i \cdot b_i(O_0)$ . Für  $t > 0$  gilt  $\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) \cdot a_{i,j} \right] \cdot b_j(O_{t+1})$ . Der Casus Knacksus, welcher eine effiziente Berechnung von  $\alpha$  ermöglicht ist die Markov-Eigenschaft. Diese besagt, dass der Zustand in Zeitpunkt  $t + 1$  nur vom Zustand in Zeitpunkt  $t$  abhängt. Für jeden Zeitpunkt und jeden Zustand gibt es also genau  $N$  Zustände in denen sich der Prozess zuvor befinden haben kann. Da der Rechenaufwand für jeden Zeitpunkt gleich ist hängt die Komplexität von  $\alpha$  mit  $N^2 \cdot T$  nur noch linear von  $T$  ab und nicht exponentiell wie in dem vorherigen Ansatz.

## Motivation Baum Welch Algorithmus

Wir sind nun also in der Lage  $P(O \mid \lambda)$  effizient zu berechnen. Was ist jedoch wenn die Parameter des Modells  $\lambda$  unbekannt sind? Wir erinnern uns, dass in unserem Beispiel

die Parameter  $\lambda = \pi, B, A$  aus den Jahren 1900 bis 2000 gewonnen wurden, da die Temperatur im Mittelalter nicht direkt beobachtbar ist. Es ist jedoch gut möglich, dass für die Jahre 500 bis 1500 eine andere Belegung von  $\lambda$  existiert, welche die beobachteten Jahresringe besser erklärt. Gesucht wird also eine Belegung  $\lambda'$  welche die Wahrscheinlichkeit der Observationssequenz  $P(O \mid \lambda')$  maximiert. Dieses Problem ist bekannt als Parameterannäherung und wird gelöst durch den Baum-Welch Algorithmus. Für den Baum-Welch Algorithmus werden zwei weitere Variablen  $\beta$  und  $\gamma$  eingeführt

Die Rückwärtsvariable *beta* funktioniert analog zu  $\alpha$  und ist wie folgt definiert.  $\beta_t(i) = P(O_{t+1}, O_{t+2} \dots O_T \mid q_t = S_i, \lambda)$  Somit ist  $\beta_t(i)$  die Wahrscheinlichkeit in Zeitpunkt  $t$  in Zustand  $S_i$  zu sein und ausgehend von  $S_i$  die partielle Observationssequenz  $O_{t+1}, O_{t+2} \dots O_T$  zu beobachten. Für  $t = T$  gilt  $\forall_i 1 \leq i \leq N \mid \beta_T(i) = 1$  Für  $t < T$  gilt  $\beta_t(i) = \sum_{j=1}^N a_{i,j} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)$

Mit den Variablen  $\alpha$  und  $\beta$  gewappnet können wir nun eine weitere Variable definieren.  $\gamma_t(i) = P(q_t = S_i \mid O, \lambda)$  Diese beschreibt die Wahrscheinlichkeit beim Beobachten von  $O$  in Zeitpunkt  $t$  in Zustand  $i$  zu sein.  $\gamma_t(i)$  wird berechnet durch

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \cdot \beta_t(i)} \quad (1.1)$$

Die letzte Variable, welche wir für den Baum-Welch Algorithmus benötigen ist  $\xi_t(i, j)$ , die vorwärts-rückwärts-Variable. Diese gibt die Wahrscheinlichkeit an in Zeitpunkt  $t$  in Zustand  $S_i$  zu sein und im nächsten Zeitpunkt  $t + 1$  in Zustand  $S_j$  zu sein.

$$\xi_t(i, j) = \frac{\alpha_t(i) \cdot a_{i,j} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)}{P(O \mid \lambda)} = \frac{\alpha_t(i) \cdot a_{i,j} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \cdot a_{i,j} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)} \quad (1.2)$$

## 1.3 Baum-Welch Algorithmus

Der Baum Welch ist ein erwartungsmaximierender Algorithmus. Er basiert auf dem Prinzip, dass wir aus den Variablen  $\xi$  und  $\gamma$  Erwartungswerte für die Anzahl an Transitionsübergängen und Symbolemissionen berechnen können. Mit dem Wissen wie oft diese Ereignisse erwartungsgemäß eintreffen können wir die Parameter des Models  $\lambda = \pi, B, A$  neu berechnen. Wir erinnern uns

$\gamma_t(i)$  = Wahrscheinlichkeit in Zeitpunkt  $t$  in Zustand  $S_i$  zu sein

$\xi_t(i, j)$  = Wahrscheinlichkeit in Zeitpunkt  $t$  in Zustand  $S_i$  und in  $t + 1$  in  $S_j$  zu sein

Wenn wir  $\gamma_{\text{gamma}_t(i)}$  über alle  $t$  exklusive  $t = T$  summieren erhalten wir die zu erwartende Anzahl an Ausgehenden Transitionen von  $S_i$

$\sum_{t=1}^{T-1} \gamma_t(i)$  = Zu erwartende Anzahl der von  $S_i$  ausgehenden Transitionen

Summieren wir  $\gamma_t(i)$  über alle  $t$  inklusive  $t = T$  erhalten wir die zu erwartende Anzahl an Aufenthalten in Zustand  $S_i$

$\sum_{t=1}^T \gamma_t(i)$  = Zu erwartende Anzahl an Aufenthalten in  $S_i$

Die zu erwartende Anzahl der Transitionen von  $S_i$  zu einem bestimmten Zustand  $S_j$  erhalten wir ähnlich, durch summieren von  $\xi_t(i, j)$  über alle  $0 < t < T$ .

$\sum_{t=1}^{T-1} \xi_t(i, j)$  = Zu erwartende Anzahl an Transitionen von  $S_i$  nach  $S_j$

Mit den Erwartungswerten für Anzahlen an Zustandsübergängen, Zustandsaufenthalten und Zustandsemissionen können wir neue Werte für die Parameter des HMM berechnen.

$$\pi_i = \gamma_1(i) b_{i,j} = \frac{\sum_{t=1}^T \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} a_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Eine Neuberechnung der Parameter kann beliebig oft durchgeführt werden. In der Praxis wird als Abbruchbedingung ein Mindestwert für die Verbesserung des Modells festgelegt.

## 1.4 Underflow und Overfitting

## 1.5 Metaheuristische Algorithmen

Metaheuristische Algorithmen werden angewendet auf "Harte" Optimierungsprobleme. In diese Kategorie fallen unter anderem Probleme für die kein Algorithmus bekannt ist, der ein globales Optimum in endlicher Zeit findet. [6] Das bestimmen optimaler Parameter eines Hidden Markov Models ist demnach ein hartes Problem.

Heuristischer Algorithmus ist salopp gesagt schlaues Raten.

Vorteile Gegenüber klassischen iterativen Algorithmen - - Metaheuristische Optimierungsverfahren können vielversprechend sein, wenn wir keine hochoptimierte Lösung suchen, sondern eine Lösung welche "gut genug" ist, aber einfach zu berechnen ist. [7]

- Eine Metaheuristik ist nicht ein bestimmter Algorithmus sondern vielmehr eine Ansammlung von Ideen, Konzepten und Operatoren, welche verwendet werden können um einen heuristischen Algorithmus zu erstellen. [4]

Eine Metaheuristik bietet ein Problemunabhängiges Framework, aus welchem man sich einen Heuristischen Algorithmus für ein spezifisches Problem bastelt. So muss man nicht ständig das Rad neu erfinden.

## 1.6 Funktionsweise

- heuristische Algorithmen lassen eine Verschlechterung einer Lösung zu in der Hoffnung dadurch lokalen Minima zu entkommen. [6]

Metaheuristische Algorithmen in letzter Zeit am stärksten

Ein Metaheuristischer Algorithmus ist inspiriert durch die Dynamiken in einem biologischen, physikalischen oder auch sozialen System.

Bekannte Beispiele sind Particle Swarm Optimization, Genetic Algorithm und

Key Unterschied ob es sich um einen Single-Solution oder Population based handelt

Key Komponente ist Exploration und Exploitation Manchmal auch Intensivierung und Diversifikation genannt [missing quote]

Exploitation beschreibt das Verbessern einer gegebenen Lösung Exploration beschreibt das Finden neuer Lösungen

Es ist wichtig eine gute Balance zu finden, denn wenn die nur Exploitation ohne Exploration ist äquivalent zu lokaler Suche

Nur Exploration ohne Exploitation ist äquivalent zu random Search

### 1.6.1 Kritik an Metaheuristischen Algorithmen

In den letzten Jahren wurde das Feld der Optimierung regelrecht überschwemmt mit "neuen" Metapher-basierten Algorithmen. Ob Mikroflodermas, Jazz-Musiker, Schwarze Löcher oder auch intelligente Wassertropfen. Jedes erdenkliche Konzept kann in einen Optimierungsalgorithmus verwandelt werden. Oft stellt sich jedoch heraus, dass das einzige, was diese "neuen" Algorithmen zum Feld beitragen, eine Umbenennung eines bereits etablierten Algorithmus ist. [1] So ist zum Beispiel Harmony Search, ein Suchalgorithmus, der auf dem Prinzip von Jazz-Musikern funktioniert, nichts weiter als eine Umbenennung eines speziellen Falles des Genetischen Algorithmus. [8] Trotz mangelnder Innovation verzeichnet eine Suche nach "harmony search" auf Google Scholar laut Weyland im Jahre 2010 586 Einträge. Im Jahre 2023 ist diese Zahl auf stolze 57.500 Einträge gestiegen, wovon 7.840 Einträge nach 2022 erschienen. Die Flut solcher vermeintlich "neuen" Algorithmen ist sehr nachteilig für das Feld der Optimierung, denn die elaborierten Metaphern für bereits existierende Konzepte führen zu Verwirrung und tatsächlich innovative Ansätze werden übersehen. [4]

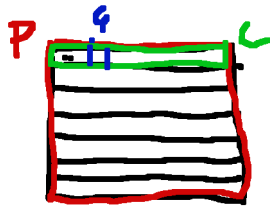
## 1.7 Genetische Algorithmen

Genetische Algorithmen nehmen Inspiration von der Evolutionstheorie von Charles Darwin. Dass angepasste Individuen erfolgreicher darin sind ihre genetischen Informationen zu verbreiten.

Die zugrundeliegende Idee ist, dass angepasste Individuen eine höhere Chance haben Nachkommen zu erzeugen. [2]

Wie die meisten stochastischen Algorithmen haben genetische Algorithmen keine Garantie ein globales Optimum zu finden [2]

Das Besondere an genetischen Algorithmen ist, dass diese nicht mit Werten aus dem Suchraum des Problems selbst arbeiten, sondern mit Chromosomrepräsentationen. [3] Ein Chromosom ist also eine Vektorrepräsentation einer Lösung aus dem Suchraum. Die einzelnen Werte eines Chromosoms nennt man Gene und eine Menge an Genen bilden eine Population. [Siehe Grafik 01] Die Chromosomrepräsentation kann Binär, Hexadezimal, mit Fließkommazahlen oder auch anderen Symbolen geschehen. Welche Werte die Gene einnehmen können hängt von der Problemdomäne ab.



Die Hauptbestandteile eines genetischen Algorithmus, welche von jeder Variante implementiert werden sind Repräsentationsoperator, Fitnessfunktion und genetische Operatoren.

Der Repräsentationsoperator übersetzt eine Lösung aus dem Suchraum in eine Chromosomrepräsentation.

Die Fitnessfunktion einem gegebenen Chromosom einen Fitnesswert zu, anhand welcher die Chromosome verglichen werden können. Die Fitness eines Chromosoms entspricht dem Wert der objektiven Funktion an der Stelle welche das Chromosom repräsentiert. [3]

Die genetischen Operatoren sind aufgeteilt in Selektionsoperator, Mutationsoperator und Crossoveroperator. Im folgenden werde ich die einzelnen Operatoren erläutern und beliebige Implementierungen aufführen.

### 1.7.1 Selektionsoperator

Der Selektionsoperator wählt eine Menge von Eltern aus der Population, aus welchen Kinder für die nächste Generation erzeugt werden. Typischerweise hat jedes Kind-Chromosom zwei Eltern-Chromosome. Es gibt jedoch auch Crossover-Operatoren welche mit mehr als zwei Eltern arbeiten. Ganz nach der Evolutionstheorie sollen hier Chromosome mit einer höheren Fitness auch eine höhere Chance haben als Elternteil gewählt zu werden. Wie stark Chromosome mit einer höheren Fitness bevorzugt werden wird als Selektionsdruck bezeichnet. Selektionsschemata können in zwei Klassen unterteilt werden, proportionale Selektion und ordinale Selektion. [3] Eine proportionale Selektion gewichtet Chromosome anhand ihrer Fitness. Ordinale Selektion gewichtet Chromosome anhand ihres Ranges. Der Selektionsdruck Bei einer proportionalen Selektion ist der Selektionsdruck hoch und es besteht das Risiko einer verfrühten Konvergenz. Denn wenn es ein Chromosom gibt, welches weitaus fitter als der Rest der Population ist wird dieses einen proportionalen Selektionsprozess dominieren und somit die genetische Diversität der Population senken. Andererseits führt ein geringer Selektionsdruck zu langsamer Konvergenz. [3] Die Auswahl des Selektionsoperators sollte also wohlüberdacht sein.

#### Roulette Rad Selektion

Roulette Rad Selektion ist einer der traditionellen proportionalen Selektionsoperatoren. Stellen wir uns ein Roulette Rad vor, welches in  $N$  Segmente unterteilt ist, wobei  $N$  die Anzahl der Chromosome in einer Population ist. Die Länge eines Segmentes  $s_i$  ist proportional zu der normalisierten Fitness des korrespondierenden Chromosoms.

$$s_i = 2\pi \cdot \frac{fitness(i)}{\sum_{j=1}^N fitness(j)} \quad (1.3)$$

Nun wird das Roulette Rad gedreht und das Chromosom auf wessen Feld man landet wird in die Menge der Eltern aufgenommen. Diese Prozedur wird so oft wiederholt bis man die gewünschte Anzahl an Eltern gesammelt hat.

#### Zufällige Selektion

Der zufällige Selektionsoperator ist der simpelste. Jedes Chromosom hat die gleiche Wahrscheinlichkeit in die Elternmenge aufgenommen zu werden.

## Rang Selektion

Die Rang Selektion ist eine Abwandlung der Roulette Rad Selektion. Die Länge eines Segmentes ist jedoch nicht proportional zu der Fitness sondern proportional zu dem Rang des Chromosoms.

$$s_i = 2\pi \cdot \frac{N - \text{rank}(i)}{\sum_{j=1}^N \text{rank}(j)} \quad (1.4)$$

## Elitismus

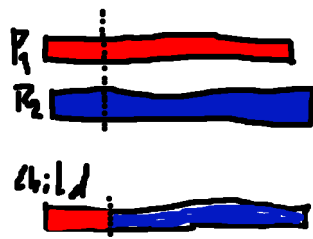
Die Selektionsoperatoren garantieren nicht, dass die besten Chromosome ausgewählt werden. Darüber hinaus kann es vorkommen, dass die besten Chromosome durch Crossover und Mutation verschlechtert werden, sodass die Fitness der folgenden Generation geringer als die vorherige ist. Um eine Abnahme der Fitness zu verhindern kann Elitismus angewendet werden. Die  $n$  besten Chromosome einer Population, die sogenannten Eliten werden auf jeden Fall in die nächste Generation aufgenommen ohne durch Crossover und Mutation verändert zu werden.

### 1.7.2 Crossoveroperator

Crossover bezeichnet das Rekombinieren von typischerweise zwei Eltern-Chromosomen zu einem Kind-Chromosom. Der Vollständigkeit halber sei angeführt, dass auch Crossoveroperatoren existieren, welche mit mehr als zwei Eltern akzeptieren oder mehr als ein Kind produzieren. In dieser Arbeit beschränken wir uns jedoch auf Crossoveroperatoren von zwei Eltern zu einem Kind. Beim Crossover werden keine neuen Informationen in den Genpool eingeführt sondern es werden nur die vorhandenen Gene der Eltern rekombiniert. Es folgt eine Aufzählung gängiger Crossoveroperatoren.

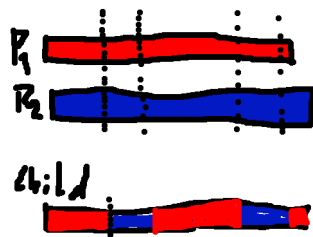
#### Single Point Crossover

Beim Single Point Crossover wird ein Cutpoint entlang der Länge der Eltern gewählt. Beide Eltern werden dann an diesem Cutpoint geschnitten und das Kind-Chromosom setzt sich zusammen aus der ersten Hälfte des ersten Elternteils und der zweiten Hälfte des zweiten Elternteils.



### N-Point Crossover

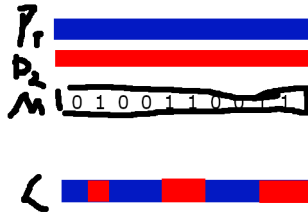
Der N-Point Crossover ist eine Generalisierung des Single Point Crossovers. Es werden  $n$  Crossover Points entlang der Länge der Eltern gewählt. Anhand dieser Crossover Points werden die Eltern in  $n + 1$  Segmente unterteilt. Das Kind erhält alle Segmente mit geradem Index vom ersten Elternteil und alle Segmente mit ungeradem Index vom zweiten Elternteil. Im Allgemeinen führen mehr cutpoints jedoch zu einer geringeren Effizienz des genetischen Algorithmus. [3]



### Uniform Crossover

Bei einem uniformen Crossover wird eine Crossovermaske  $m$  mit gleicher Länge zu den Eltern erstellt. Das Kind erhält Gene der Eltern nach dieser Crossover Maske, wobei  $m_i$  angibt, von welchem Elternteil das  $i$ -te Gen bezogen wird. Für jedes Elternpaar wird eine neue Crossovermaske erstellt. Typischerweise gilt  $P(m_i = 1) = P(m_i = 0) = 0.5$ . Die Wahrscheinlichkeit, dass ein Gen von einem Elternteil bezogen wird kann jedoch auch gewichtet werden anhand der Fitness oder Ränge, so dass  $P(m_i = 0) = w$  und  $P(m_i = 1) = 1 - w$





### 1.7.3 Mutationsoperator

Der Crossover Schritt verringert unweigerlich die genetische Diversität der Population. Um dem entgegenzuwirken muss es einen Mechanismus geben, welcher neues genetisches Material hinzufügt. Dieser Mechanismus ist die Mutation, welche ein Chromosom zufällig verändert.

TODO: Include Examples for Real valued GA crossover

#### Mutationschance

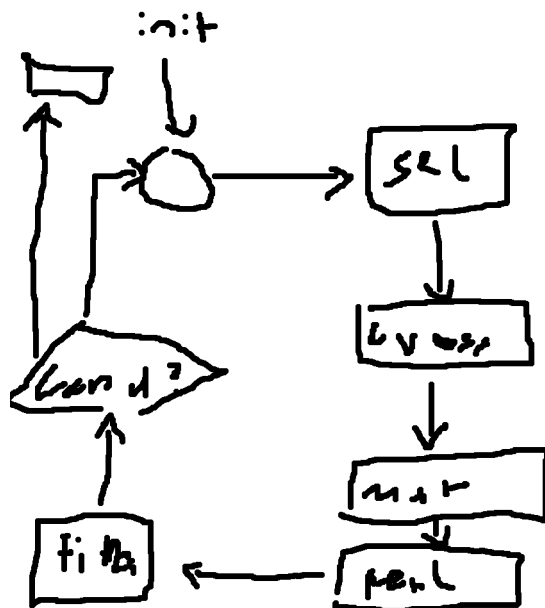
Die Mutationschance bestimmt wie viele Gene eines Chromosoms durch die Mutation verändert werden und wie viele Unverändert bleiben. Bei einer Mutationschance von 100% werden alle Gene eines Chromosoms verändert und der genetische Algorithmus ist equivalent zu einer zufälligen Suche. [3]

### 1.7.4 Ersetzen

TODO: Ersetzen moped schmoped

### 1.7.5 Ablauf eines genetischen Algorithmus

Ein Iterationsschritt des klassischen genetischen Algorithmus besteht aus Vier Schritten. Die Abbruchbedingung eines Genetischen Algorithmus kann zum Beispiel vom Mittelwert, der Summe oder dem Maximalwert der Fitness abhängen oder aber auch einfach nach einer vorher festgelegten Anzahl an Iterationen terminieren. Der Ablauf einer Iteration ist in Grafik so und so beschrieben.



Population Size Wählt man diese Zu gering kann es zu einer verfrühten Konvergenz und somit auch einer schlechten Lösung kommen Gleichzeitig ist die Population size ein Maßgebender Faktor in der Rechenzeit und eine zu große Population könnte Rechenzeit verschwenden [GAs]

Im folgenden werde ich bekannte implementationen der genetischen Operatoren eingehen und beschreiben worauf man bei der implementation achten muss

### Mutationsoperatoren

Random Uniform Mutation: Random Unifor mutation

### Parameter eines genetischen Algorithmus

- Welche Crossover Funktion (evtl. Parameter der Crossover Funktion) - Welche Mutations Funktion (evtl. Parameter der Mutations Funktion) - Populationsgröße - Rekombination - Selektionsfunktion -
- Populationsgröße Die globale Suchkapazität eines genetischen Algorithmus hängt stark von der gewählten Populationsgröße ab. Eine größere Population ist hier von Vorteil. Jedoch benötigt eine große Population auch mehr Rechenleistung, Speicher und Zeit. [3]
- Auswahl des Mutationsoperators. Mutationsoperatoren unterscheiden sich stark in der benötigten Rechenleistung. Der Single Point Crossover-Operator benötigt zum Beispiel

nur ein einziges Zufallsereignis, wohingegen ein Uniformer Crossover-Operator so viele Zufallsereignisse wie es Gene gibt benötigt. Bei einer Chromosomrepresentation mit vielen Genen macht sich dieser Unterschied bemerkbar.



## Chapter 2

# Trainieren von HMMs mit einem GA

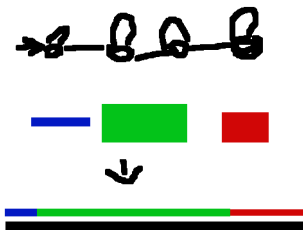
### 2.1 Konstruktion eines GAHMM

In diesem Kapitel werde ich beschreiben wie man einen Genetischen Algorithmus für Hidden Markov Modelle erstellen kann und dann auf die in meiner Arbeit verwendeten konkreten Implementationen der genetischen Operatoren eingehen.

#### Representation

Ähnlich wie in den Papers [hier papers einfügen] Ist die Chromosonale Representation eines HMMs ein Vektor welcher die Reihen aller Matrizen hintereinander enthält.

Um die Struktur von nicht ergodischen Hidden Markov Modellen im Mutations-schritt nicht zu Verändern muss man die Gene Welche Startwahrscheinlichkeiten und Transitionswahrscheinlichkeiten beschreiben von der Mutation ausschließen oder man kann eine Maske definieren welche eine Mutation Gene welche initial 0 oder 1 sind verhindert.



#### Fitness Operator

Die Fitness eines Chromosoms ist die durchschnittliche Log Wahrscheinlichkeit des Hidden Markov Models, welchen das Chromosom representiert. Als Observations Sequenzen werden Äußerungen der Zahl 0 aus dem Free Spoken Digit Dataset verwendet

## Mutationsoperatoren

Mutationsoperatoren wird dies das annanas gewählt

Belegung der Parameter:

Hidden Markov Parameter Bei den Hidden Markov Modellen welche von den genetischen Algorithmus optimiert werden handelt es sich ausschließlich um Left-right Modelle mit 4 Zuständen und 128 Observationssymbolen. Diese Werte sind relativ Arbiträr gewählt und orientieren sich an existierender Literatur zu Spracherkennung mit Hidden Markov modellen.

Als Observations-Sequenzen werden Äußerungen der Zahl 0 aus der Free Spoken Digit Database gewählt. Welche mittels eines k-means algorithmus mit  $k=128$  quantisiert wurden.

Genetischer Algorithmus Parameter:

Die Populationsgröße wird auf 50 beschränkt und die Anzahl der Generationen auf 100. Das Rational für diese Entscheidung ist, dass ein Ablauf des Genetischen Algorithmus mit diesen Parametern auf meinem Klapprechner in unter 5 minuten abläuft und 100 Generationen meißt ausreichen um den Genetischen Algorithmus konvergieren zu lassen Die Anzahl der Observationssequenzen ist 10

## Erweiterung des GA

- Der genetische Algorithmus muss um einen Normalisierungsschritt erweitert werden, falls die Crossover oder Mutationsoperatoren keine Reihenstochastizität garantieren.

## 2.2 Literaturübersicht

TODO: in diesem Chapter gebe ich eine übersicht über die unternommenen Ansätze das Training von Hidden Markov Modellen mit hybriden metaheuristischen Algorithmen.

## **Chapter 3**

### **Evaluation**

#### **3.1 Vergleich genetischer Operatoren**

#### **3.2 Berechnungskost eines GA**

- Wallah Teuer - Lohnt net





# Bibliography

- [1] *Grey Wolf Firefly and Bat Algorithms Three Widespread Algorithms that Do Not Contain Any Novelty*. 2020.
- [2] Sivanandam. *Genetic Algorithms*. In: *Introduction to Genetic Algorithms*. Springer. 2008.
- [3] Sivanandam. *Terminologies and Operators of GA*. In: *Introduction to Genetic Algorithms*. Springer. 2008.
- [4] Kenneth Sorensen. *Metaheuristics—the metaphor exposed*. 2012.
- [5] Springer. *Genetic Algorithms*. 2005.
- [6] Springer. *Metaheuristics*. 2008.
- [7] Springer. *Metaheuristics and Evolutionary Computation: Algorithms and Applications*.
- [8] Dennis Weyland. *A rigorous analysis vong dem harmony moped*. 2010.



## **Versicherung an Eides Statt**

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit - mit Ausnahme der Anleitung durch die Betreuer  
- selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides Statt nach §156 und §163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

Duisburg, 21. Februar 2023  

---

(Ort, Datum)

---

(Vorname Nachname)