

**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

15. 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</b>	<b>Related Work</b>	<b>1</b>
1.1	Markov Prozesse . . . . .	1
1.2	Hidden Markov Modelle . . . . .	2
1.3	Baum Welch Algorithmus . . . . .	4
1.4	Genetische Algorithmen . . . . .	4
1.5	Metaheuristische Algorithmen . . . . .	6
1.6	Genetischer Algorithmus für HMMs . . . . .	6



# Chapter 1

## Related Work

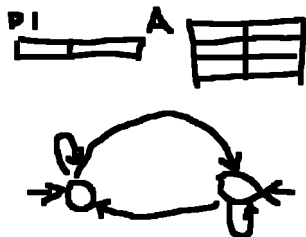
### 1.1 Markov Prozesse

Markov Ketten wurden 1906 von Andrei Andreyvich Markov eingeführt. Ein Markov Prozess modelliert ein zeitdiskretes Signal mit einer endlichen Menge an Zuständen. Dieser Prozess wird beschrieben durch einen Startwahrscheinlichkeitsvektor  $\pi$  und eine Transitionsmatrix  $A$ .  $\pi$  ist die Wahrscheinlichkeit sich in Zeitpunkt  $t = 0$  in Zustand  $i$  zu befinden und  $A_{i,j}$  ist die Wahrscheinlichkeit dass der Prozess sich in Zeitpunkt  $t$  in Zustand  $j$  befindet unter der Bedingung, dass sich der Zustand in Zeitpunkt  $t-1$  in Zustand  $i$  befand. Man beobachte, dass die Transitionswahrscheinlichkeiten unabhängig von  $t$  sind, so dass der nächste Zustand einzig und allein vom gegenwärtigen Zustand des Prozesses anhängt. 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 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  $O = \{O_1, O_2, \dots, O_{T-1}, O_T\}$   $P(O \mid Model) = \pi_{O_1} \cdot \prod_{t=2}^T A_{O_{t-1}, O_t}$

Ein Prozess kann also nur mit einer Markovkette modelliert werden, wenn wir diesen Prozess beobachten können. Es kann jedoch sein, dass wir den Prozess selbst nicht beobachten können dann benötigen wir ein Hidden Markov Model um diesen zu modellieren.

## 1.2 Hidden Markov Modelle

Angenommen wir möchten die durchschnittlichen jährlichen Temperaturen modellieren für einen Zeitraum bevor es Temperaturaufzeichnungen gab. Wir können die Temperatur in diesem Fall 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 S, M und L die verschiedenen Breiten an Jahresringen welche beobachtet werden können. S steht für dünne, M für mittlere und L für breite Jahresringe. Unser Modell wird nun um eine Emissionsmatrix B erweitert.

Die Wahrscheinlichkeit für eine gegebene Temperatur irgend eine dicke zu Beobachten ist ebenfalls ein sicheres ereignis und somit gilt auch für die Reihen von B analog zu A.

$B_{i,k}$  ist die Wahrscheinlichkeit in Zustand i Emissions-Symbol k zu beobachten.  $B_{1,2}$  gibt die Wahrscheinlichkeit an in in einem Heißen Jahr Jahresringe mittlerer Breite zu beobachten.



Mit diesem erweiterten Modell kann man nun folgende Fragen stellen Problem 1: Was ist die Wahrscheinlichkeit einer gegebenen Observationssequenz unter einem gegebenen Modell?

Problem 2: Gegeben eine Observationssequenz und ein Model. Aus welcher Sequenz an Zuständen ist die Observationssequenz am Wahrscheinlichsten entstanden?

Problem 3: Gegeben ein Modell und eine Observationssequenz. Wie kann man die Wahrscheinlichkeit  $P(O | \lambda)$  maximieren?

### Problem 1

Wie berechnet man die Wahrscheinlichkeit einer Observationssequenz unter einem Modell? Wenn wir die Abfolge der Zustände haben lässt sich  $P(O | \lambda)$  sehr einfach berechnen da die Wahrscheinlichkeit ein Symbol  $k$  zu beobachten unter der Bedingung in Zustand  $i$  zu sein nichts anderes ist als  $B_{i,k}$   $P(O | Q, \lambda) = \prod_{t=1}^T b_{q_t}(O_t)$

Die Wahrscheinlichkeit von  $P(O | \lambda)$  ist equivalent zu der Summe der bedingten Wahrscheinlichkeiten aller möglichen Zustandsfolgen.  $P(O | \lambda) = \sum_{all Q} P(O | Q, \lambda)$  Es ist möglich  $P(O | \lambda)$  so zu berechnen, jedoch wächst die Anzahl der Möglichen Zustandsabfolgen  $Q$  exponentiell in Abhängigkeit von  $T$ . Präzise gesagt gilt  $|Q| = 2^T \cdot N^T$ . Selbst wenn  $N$  und  $T$  kleine Werte einnehmen ist die benötigte Rechenleistung untragbar. Für  $N=5$  und  $T=100$  müssten  $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$  Berechnungen durchgeführt werden. Um diese Zahl in Perspektive zu packen  $10^{72}$  ist mindestens 3 mal mehr als 1000 und 1000 ist schon ziemlich groß.

### Forward Backward Procedure

Sei  $\alpha_t$  die Forwärtvariable folgendermaßen definiert

$$\alpha_t(i) = P(O_1 O_2 \dots O_t \wedge q_t = S_i | \lambda)$$

$\alpha_t(i)$  Beschreibt also die Wahrscheinlichkeit die partielle Observationssequenz  $O_1 O_2 \dots O_t$  zu beobachten und in Zeitpunkt  $t$  in 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.  $\alpha_0(i) = \pi_i \cdot b_i(O_0)$

$$\text{Für } t > 0 \text{ gilt } \alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) \cdot a_{i,j} \right] \cdot b_j(O_{t+1})$$

Eine effiziente berechnung von  $\alpha$  ist möglich aufgrund der Markov-Eigenschaft, welche besagt, dass der Zustand in Zeitpunkt  $t+1$  nur vom Zustand in Zeitpunkt  $t$  abhängt. Für

jeden Zeitpunkt und jeden Zustand gibt es genau  $N$  Zustände in denen sich der Prozess zuvor befunden haben kann. Somit ergibt sich eine Rechenkomplexität von  $N^2 \cdot T$

Wahrscheinlichkeit die Partielle Observationssequenz zu beobachten hängt von der Wahrscheinlichkeitsverteilung ab in welchem Zustand sich der Prozess befand nach dem die Partielle Observationssequenz  $t-1$  beobachtet wurde. Da für jeden Zeitpunkt  $t$  sich der Prozess in  $t-1$

Nach der Markov-Eigenschaft gilt, dass der Zustand in Zeitpunkt  $t + 1$  einzig und allein vom Zustand in Zeitpunkt  $t$  abhängt. Daher ist eine effiziente Berechnung von  $\alpha$  möglich, da für

Der Prozess kann sich nach Beobachtung von  $o \dots t$  nur in  $N$  Zuständen befinden. Da wir für jeden Zeitpunkt und für jeden Zustand

Wichtig ist anzumerken, dass  $\alpha$  nicht die Wahrscheinlichkeit ist in Zeitpunkt  $t$  in Zustand  $i$  zu sein sondern nur die Wahrscheinlichkeit angibt  $i$  zu sein

Wir führen die Rückwärtsvariable  $\beta$  ein, welche wie folgt definiert ist.  $\beta_t(i) = P(O_{t+1}, O_{t+2} \dots O_T \mid q_t = S_i, \lambda)$  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

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

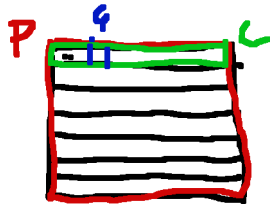
## 1.3 Baum Welch Algorithmus

In den vorherigen Beispiel waren die Werte für die Parameter  $\pi$ ,  $A$  und  $B$  gegeben. Häufig sind diese Werte jedoch unbekannt und es liegen lediglich Observationssequenzen vor.

## 1.4 Genetische Algorithmen

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

Ein Genetischer Algorithmus arbeitet nicht mit den möglichen Lösungen selbst sonder mit Chromosom Representationen der Lösungen. [GAs] Ein Chromosom ist also ein Vektor Representation einer Lösung des Problems Die einzelnen Werte eines Chromosoms nennt man Gene und eine Menge an Genen bilden eine Population. [Siehe Grafik 01]



Jeder Genetische Algorithmus implementiert eine representations Operator Fitness Operator und Genetische Operatoren [missing quote] Selection Operator Mutation Operator Crossover Funktion

Der Representationsoperator übersetzt eine Lösung aus dem Search space in ein Chromosom.

Die Fitness Funktion weist jedem Individuum einen Wert zu anhand welchem man die Individuen vergleichen kann

Der Selection Operator wählt Chromosomepaare aus. Dies kann zufällig geschehen, meist wird der selection Operator jedoch anhand der Fitness oder Ränge ausgewählt

Der Crossover Operator erhält eine Menge an Chromosomen und rekombiniert diese zu Kindern. Typischerweise werden je zwei Eltern zu einem Kind rekombiniert.

Der Mutationsoperator verändert ein Chromosom zufällig

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

## 1.5 Metaheuristische Algorithmen

Diggahhhhh was zum fiiiickkkkkk????

Metaheuristische Algorithmen in letzter zeit am stabbo

Ein Metaheuristischer Algorithmus ist inspiriert durch die dynamiken in einem Biologischen, Physikalischen oder auch Sozialen System.

Bekannte Beipsiele 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 intensivication und diversification gennant [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 equivalent zu Lokaler Suche

Nur Exploration ohne Exploitation ist equivalent zu random Search

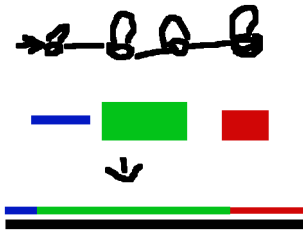
## 1.6 Genetischer Algorithmus für HMMs

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 Transitions-wahrscheinlichkeiten 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 Modells, welchen das Chromosom repräsentiert. 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

was zum fick?







## **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 vorge-  
nommen 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, 15. Februar 2023  

---

(Ort, Datum)

---

(Vorname Nachname)