

Master in Human Centered Computing
„Formale Methoden“

SS2015

Prof. Dr.-Ing. habil. Natividad Martínez

- Abschlussausarbeitung -

Das Hidden Markov Model im Machine Learning Kontext

vorgelegt von:

Paul Pasler

1. Semester

Contact addresses:

paul.pasler@student.reutlingen-university.de

Submitted on: 20.06.2015

Inhaltsverzeichnis

Abbildungsverzeichnis

1	Einführung	1
1.1	Machine Learning	1
1.2	Wahrscheinlichkeitsrechnung	3
2	Markov Kette	4
2.1	Definition	4
2.2	Beispiel	5
3	Hidden Markov Model	7
3.1	Definition	7
3.2	Beispiel	9
3.3	Funktionsweise	9
4	Zusammenfassung	12
4.1	Fazit	12
4.2	Vergleich mit anderen Machine Learning Ansätzen	12

Literatur

Anhang	1
A Anhang 1	1

Abbildungsverzeichnis

1.1	Birne und Apfel unterscheiden sich durch Farbe, Form etc. - einen Stiel haben jedoch beide	1
1.2	Vereinfachte Darstellung eines neuronalen Netzwerkes mit 3 Merk- maldimensionen.	3
1.3	Daten lassen sich nicht immer linear trennen.	3
2.1	Einfaches Zustandsdiagramm einer Markov Kette (Quelle: de. wikipedia.org/wiki/Markow-Kette)	5
2.2	Gerichteter Zustandsgraph der modellierten Wetter-Markov Ket- te	6
3.1	Hidden Markov Model für das Beispiel des Gefangen im Verlies .	9

1 Einführung

Das Hidden Markov Model hat im Bereich des maschinellen Lernens viele Anwendungsfälle. In der vorgelegten Arbeit wird die Funktionsweise des Hidden Markov Model erklärt werden (Kapitel 3). Die dafür notwendigen Grundlagen werden in den nachfolgenden Abschnitten und im Kapitel 2 beleuchtet. Kapitel 4 befasst sich mit einer Einschätzung des Hidden Markov Models und dem Vergleich mit anderen Ansätzen im Machine Learning Kontext.

1.1 Machine Learning

Machine Learning befasst sich mit der modellierung des Lernvorgangs auf einem Computer [Mar09]. Es wird versucht ein “künstliche” Generierung von Wissen aus Erfahrung zu erzeugen. Dabei wird anhand von Beispielen “gelernt”, sodass nicht nur die selben Daten wieder erkannt werden können, sondern auch ähnliche bzw. unbekannte Daten klassifiziert werden. Diese Transferleistung nennt man Generalisierung und ist auch beim Menschen eine wichtige Eigenschaft im Lernvorgang.

So können wir Äpfel von Birnen (Siehe Abbildung 1.1 ¹) unterscheiden, egal, ob wir genau diese Frucht schon einmal gesehen haben. Wir entscheiden anhand gelernter Merkmale, um welche Frucht es sich vermutlich handelt. Merkmale sind bspw. Größe, Form, Farbe, Geruch etc.



Abb. 1.1: *Birne und Apfel unterscheiden sich durch Farbe, Form etc. - einen Stiel haben jedoch beide*

Die Extraktion signifikanter Merkmale ist ein wichtiger Teil von Machine Learning. Viele Eigenschaften eines Objektes sind nicht geeignet es von anderen zu

¹Quelle: <http://www.lifeline.de/img/abnehmen/origs76797/7656955923-w830-h830/Birne-und-Apfel.jpg>

unterscheiden. Im Apfel-Birnen-Beispiel würde das Merkmal “Stiel” nicht zu einer Unterscheidung führen.

Der nächste Schritt ist das Training des Systems. Hierbei wird zwischen drei algorithmischen Ansätzen unterschieden:

- Überwachtes Lernen
- Unüberwachtes Lernen
- Bestärkendes Lernen

Die häufigste menschliche Lernform, ist das bestärkende Lernen, hier wird mit “Belohnung” und “Bestrafung” gearbeitet. Im Machine Learning Bereich ist jedoch Überwachtes und Unüberwachtes Lernen sehr viel häufiger zu finden. Beim überwachten Lernen, werden mehrere Eingaben und Lösungen an den Algorithmus überreicht und nach einigen Durchgängen sollte er in der Lage sein Assoziationen herzustellen. Je nach Algorithmus werden hierzu Funktionen und Gewichtungen angepasst.

Das Hidden Markov Model ist im Bereich des Unüberwachten Lernens beheimatet. Aus der Menge der Eingaben wird ein Modell erzeugt, das Vorhersagen ermöglichen soll. Mit einem Expectation-Maximization-Algorithmus (EM-Algorithmus) wird versucht, die vorliegenden Daten in Kategorien einzuteilen, sodass die Daten optimal erklärt werden. Eine Form des EM-Algorithmus kommt auch beim Hidden Markov Model zum Einsatz.

Weitere Machine Learning Ansätze sind

- Neuronale Netze
- Support Vector Machine
- K-Means

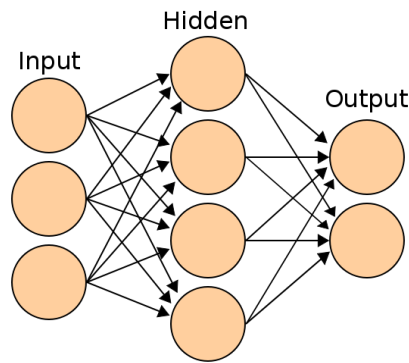


Abb. 1.2: Vereinfachte Darstellung eines neuronalen Netzwerkes mit 3 Merkmaldimensionen.

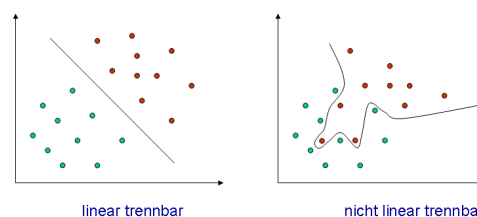


Abb. 1.3: Daten lassen sich nicht immer linear trennen.

Neuronale Netze versuchen das menschliche Gehirn mit seinen Neuronen und Synapsen nachzubauen [MP88]. Für jede Dimension des Merkmalsvektors sind Neuronen vorhanden, welche wiederum mit anderen Neuronen verschaltet sind. Beim Training werden die Gewichtungen der einzelnen Verschaltungen verändert. Abbildung 1.2² zeigt ein vereinfachtes neuronales Netz mit drei Inputs, vier weiteren Neuronen und zwei Outputs.

Die Support Vector Machine (Stützvektormaschine) versucht die Daten durch lineare Trennung zu Klassifizieren (Siehe Abbildung 1.3³) [BGV92]. Es wird versucht, den Stützvektor möglichst weit von den beiden Klassen entfernt zu erstellen (Large-Margin-Classifer).

Im Abschnitt 4.2 werden die vorgestellten Ansätze mit dem Hidden Markov Model verglichen.

1.2 Wahrscheinlichkeitsrechnung

Notwendige mathematische Grundlagen

²Quelle: http://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Artificial_neural_network.svg

³Quelle: <http://upload.wikimedia.org/wikipedia/de/a/a0/Diskriminanzfunktion.png>

2 Markov Kette

Grundlage des Hidden Markov Model war die vom russischen Mathematiker Andrej Andrejewitsch Markov (1856 - 1922, siehe [Mar13]) entwickelte Markov Kette. Zu Beginn des 20. Jahrhunderts beschäftigte er sich als erster mit einer statistischen Beschreibung von Zustands- und Symbolfolgen. Er führte eine statistische Analyse der Buchstabenfolge des Textes “Eugene Onegin” von Alexander Pushkin durch.

2.1 Definition

Eine Markov Kette beschreibt einen zeit-diskreten Prozess $(X_t)_{t \in \mathbb{N}_0}$ mit m abzählbaren Zuständen S [KHW13]. Weiterhin wird sie als stationär bezeichnet, wenn alle Wahrscheinlichkeiten unabhängig von der Zeit sind. Da die Verteilung der Zufallsvariablen nur von den vergangenen Zuständen abhängt, gilt eine Markov Kette als kausal [Fin03, 48]. Wichtig für eine Markov Kette ist die sog. Markov-Eigenschaft:

$$\begin{aligned} P(X_{t+1} = s_{t+1} | X_0 = s_0, \dots, X_{t-1} = s_{t-1}, X_t = s_t) \\ = P(X_{t+1} = s_{t+1} | X_t = s_t) \end{aligned}$$

Genügt eine Markov Kette dieser Eigenschaft, wird sie als “einfach” oder Markov Kette 1. Ordnung bezeichnet. Anders ausgedrückt beschreibt die Markov-Eigenschaft die Gedächtnislosigkeit des Prozesses, da der Folgezustand nur vom direkten Vorgänger abhängt.

Als Übergangswahrscheinlichkeit bezeichnete man die bedingte Wahrscheinlichkeit $P(X_{t+1} = s_{t+1} | X_t = s_t)$, sodass auf den aktuellen Zustand s_t der Nachfolgezustand s_{t+1} folgt. Diese Wahrscheinlichkeiten werden üblicherweise zu einer Übergangsmatrix zusammengefasst:

$$A = [a_{ij}] = \begin{bmatrix} a_{00} & \cdots & a_{0m} \\ \vdots & \ddots & \vdots \\ a_{m0} & \cdots & a_{mm} \end{bmatrix} \forall i, j \in S$$

Da es sich um Wahrscheinlichkeiten handelt, muss sich die Summe jeder Reihe zu Eins addieren.

Weiterhin benötigt der Prozess einen Vektor für den Anfangszustand $t = 0$:

$$\Pi = [\pi_i] = [P(X_0 = i)], i \in S$$

So lässt sich eine Markov-Kette durch Zustandsraum S , den Übergangsmatrix A und einen Anfangszustand Π definieren. Veranschaulichen lässt sich eine Markov Kette als gerichtetes Zustandsdiagramm (Abb. 2.1) mit den Zuständen S und mit den Übergangswahrscheinlichkeiten X_i an den Kanten

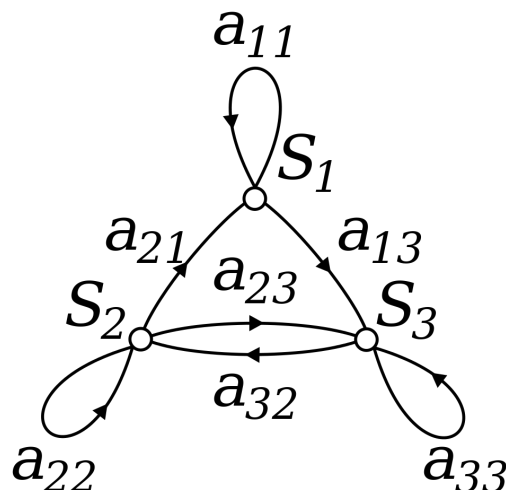


Abb. 2.1: Einfaches Zustandsdiagramm einer Markov Kette (Quelle: de.wikipedia.org/wiki/Markov-Kette)

Die Wahrscheinlichkeit für k -Schritte lässt sich so ausrechnen:

$$X_k = X_{k-1}A = X_0A^k$$

2.2 Beispiel

Markov Kette für das Wetter ¹

Im folgenden Beispiel soll aufgrund des aktuellen Wetters auf das Wetter der

¹Quelle: en.wikipedia.org/wiki/Examples_of_Markov_chains

folgenden Tage geschlossen werden. Das Wetter kann entweder “sunny” oder “rainy” sein, zu Beginn (Tag 0, $t = 0$) des Experiments ist es “sunny”. Die Wahrscheinlichkeit, dass auf “sunny” wieder “sunny” folgt, liegt bei 90% (“rainy” = $1 - \text{“sunny”} = 10\%$). Nach “rainy” liegt die Wahrscheinlichkeit jeweils bei 50% (siehe Abb. 2.2).

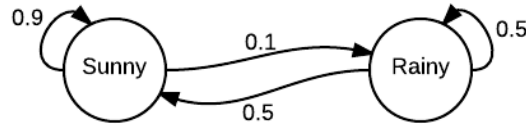


Abb. 2.2: Gerichteter Zustandsgraph der modellierten Wetter-Markov Kette

Zustände : $S = [\text{“sunny”}, \text{“rainy”}]$

Anfangszustand : $\Pi = X_0 = [1, 0]$

Übergangsmatrix : $A = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$

Nun kann die Wahrscheinlichkeit für das Wetter an Tag 1 berechnet werden über:

$$X_1 = X_0 * A = [1, 0] \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} = [0.9, 0.1]$$

Für Tag 2:

$$X_2 = X_1 A = X_0 A^2 = [1, 0] \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}^2 = [0.86, 0.14]$$

Verallgemeinert für Tag k bedeutet das:

$$X_k = X_{k-1} A = X_0 A^k = [1, 0] \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}^k$$

3 Hidden Markov Model

Das Hidden Markov Model ist ein stochastisches Modell für sequentielle Daten und wird vor allem in der Spracherkennung und in der Bioinformatik eingesetzt. Der amerikanische Mathematiker Leonard E. Baum (* 1931) und andere Autoren entwickelten auf Basis der Markov Kette Ende der sechziger Jahre das Hidden Markov Model [BP66]. Erste Hidden Markov Model-Applikationen wurden zur Spracherkennung und später auch in der Bioinformatik zur Analyse von Nukleotid- und Proteinsequenzen eingesetzt.

3.1 Definition

Ein Hidden Markov Model erweitert eine Markov Kette um einen weiteren Zufallsprozess und ist somit ein zweistufiger stochastischer Prozess [Fin03, 67]. Hierfür wird jedem Zustand der Markov Kette eine Ausgabe bzw. Emission zugeordnet, dessen Wahrscheinlichkeitsverteilung einzig vom aktuellen Zustand abhängig ist. Die Emissionen sind die einzigen beobachtbaren Zustände des Hidden Markov Model. Der Rest ist sozusagen 'versteckt', woher sich auch der Name des Modells ableitet. Eine Folge von Emissionen wird auch Observationsfolge genannt.

Das Hidden Markov Model wird definiert durch [Fin03, 68]:

$$\lambda = (S; V; A; B; \pi)$$

- Endlich Menge von Zuständen

$$S = \{s | 1 \leq s \leq N\}$$

- Alphabet der Emissionen

$$V = \{v | 1 \leq v \leq M\}$$

- Matrix der Zustandsübergangswahrscheinlichkeiten

$$A = \{a_{ij} | a_{ij} = P(S_t = j | S_{t-1} = i)\}$$

- Matrix der Emissionsverteilung

$$B = \{b_{jk} | b_{jk} = P(O_t = o_k | S_t = j)\} \text{ bzw. } B = \{b_j(x) | b_j(x) = p(x | S_t = j)\}$$

- Vektor von Zustandsstartwahrscheinlichkeiten

$$\pi = \{\pi_i | \pi_i = P(S_1 = i)\}$$

Die Emissionsmodellierung ist hierbei vom Kontext der Problemstellung abhängig. Wird das Hidden Markov Model zum Beispiel bei der Analyse von biologischen Sequenzen, sprich einem diskreten Symbolinventar, angewendet, wird ein diskretes Emissionsmodel genutzt. Man spricht hierbei auch von einem diskreten Hidden Markov Model. Wenn dieses Model zur Verarbeitung von Signalen verwendet werden soll erfordert dies in der Vorverarbeitung der Daten einen Quantisierer der die kontinuierlichen Merkmale in eine diskrete Observationsfolge überführt.

Gängiger ist es hierfür kontinuierliche Hidden Markov Model's zu nutzen. Hierbei wird eine Emissionsmodellierung auf Basis kontinuierlicher Dichtefunktionen genutzt die kontinuierliche Observationen im \mathbb{R}^n verarbeitet.

$$B = \{b_j(x) | b_j(x) = p(x | S_t = j)\}$$

Zur Behandlung kontinuierlicher Verteilungen mit mehreren komplexen Häufigkeitsgebieten werden approximative Verfahren genutzt. Die verbreitetste Technik besteht aus der Verwendung von Mischverteilungen auf der Basis von Gauß-Dichten (Gaussian Mixture Model). Man kann nämlich zeigen, dass sich jede allgemeine kontinuierliche Verteilung $p(x)$ durch eine Linearkombination von i.a. unendlich vielen Basis-Normalverteilungen beliebig genau approximieren lässt [Fin03, 69]:

$$\begin{aligned} p(x) &\hat{=} \sum_{k=1}^{\infty} c_k N(x | \mu_k, K_k) \\ &\approx \sum_{k=1}^M c_k N(x | \mu_k, K_k) \end{aligned} \quad (3.1)$$

Der Approximationsfehler lässt sich hierbei über eine geeignete Anzahl von M Basisverteilungen klein halten. Somit ergibt sich für die Beschreibung der Emissionsverteilung eines Zustands des Hidden Markov Model folgende Formel:

$$b_j(x) = \sum_{k=1}^M c_{jk} g_{jk}(x) \quad (3.2)$$

Die Anzahl der Basisverteilungen eines Gaussian Mixture Model kann hierbei für die einzelnen Zustände des HMM variieren.

3.2 Beispiel

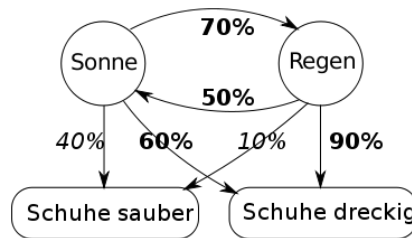


Abb. 3.1: *Hidden Markov Model für das Beispiel des Gefangenen im Verlies*

Abbildung 3.1 soll ein Hidden Markov Model an dem Beispiel des Gefangenen im Verlies darstellen¹. Ein Gefangener im Kerkerverlies möchte das aktuelle Wetter herausfinden. Er weiß, dass auf einen sonnigen Tag zu 70 % ein Regentag folgt und dass auf einen Regentag zu 50 % ein Sonnentag folgt. Weiß er zusätzlich, dass die Schuhe der Wärter bei Regen zu 90 % dreckig, bei sonnigem Wetter aber nur zu 60 % dreckig sind, so kann er durch Beobachtung der Wärterschuhe Rückschlüsse über das Wetter ziehen (das heißt, er kann die Wahrscheinlichkeit für Regenwetter gegenüber sonnigem Wetter abschätzen). Sonne und Regen sind in diesem Fall die versteckten Zustände. Die Emissionen bzw. die Observation die der Gefangene machen kann sind nur der Verschmutzungsgrad der Schuhe der Wärter.

3.3 Funktionsweise

Das Konzept des Hidden Markov Model kann laut [Rab89] in drei Problemstellungen eingeteilt werden:

- Evaluierungsproblem: Bestimme die Wahrscheinlichkeit für ein Model mit der dieses eine gegebene Observationsfolge erzeugt.
- Dekodierungsproblem: Finde interne Abläufe für eine gegebene Observationsfolge
- Trainingsproblem: Finde Modellparameter für gegebene Beispieldaten

¹Quelle: http://de.wikipedia.org/wiki/Hidden_Markov_Model

Evaluierung

In der Evaluierung soll die Wahrscheinlichkeit bestimmt werden mit der eine betrachtete Observationsfolge in einer beliebigen Zustandsfolge von einem gegebenen Hidden Markov Model λ generiert wird. Diese Wahrscheinlichkeit wird Produktionswahrscheinlichkeit genannt. Diese wird mit dem Forward-Algorithmus berechnet. Der Algorithmus nutzt hierfür die geltende Markov Eigenschaft aus das nur die Speicherung eines internen Zustandes erlaubt. Hierfür definiert man als Vorwärtsvariable $\alpha_t(i)$ die Wahrscheinlichkeit, bei gegebenem Model λ den Anfang der betrachteten Observationsfolge O_t zu erzeugen und zum Zeitpunkt t den Zustand i zu erreichen.

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, s_t = i | \lambda) \quad (3.3)$$

Die Vorwärtsvariable lässt sich nun mit den folgenden Schritten rekursiv berechnen um die Gesamtwahrscheinlichkeit des Models zu erhalten.

1. Initialisierung

$$\alpha_1(i) := \pi_i b_i(O_1)$$

2. Rekursion

für alle Zeitpunkte $t, t = 1 \dots T - 1$

$$\alpha_{t+1}(j) := \sum_i \{\alpha_t(i) a_{ij}\} b_j(O_{t+1})$$

3. Rekursionsabschluss

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

Dekodierung

Bei der Dekodierung soll die optimale, bzw. wahrscheinlichste Zustandsfolge s^* aus der Menge der Zustände ermittelt werden die eine gegebene Observationsfolge erzeugt. Zur Ermittlung der optimalen Zustandsfolge bedient man sich des Viterbi-Algorithmus, einem induktiven Verfahren das dem Forward Algorithmus sehr ähnlich ist. Zu Beginn werden erneut die Wahrscheinlichkeiten $\delta_t(i)$ für partiell optimale Pfade definiert, die das Anfangssegment der Observationsfolge bis O_t mit maximaler Wahrscheinlichkeit erzeugen und in Zustand i enden.

$$\delta_t(i) = \max_{s_1, s_2, \dots, s_t} P(O_1, \dots, O_t, s_1, \dots, s_t = i | \lambda) \quad (3.4)$$

Der Algorithmus entspricht weitgehend dem Forward-Algorithmus jedoch werden anstatt der Summe im Rekursionsabschluss, die Maximalen über die in den Vorgängerzuständen vorliegenden Wahrscheinlichkeiten gebildet.

1. Initialisierung

$$\delta_1(i) := \pi_i b_i(O_1)$$

$$\phi_1(i) := 0$$

2. Rekursion

für alle Zeitpunkte $t, t = 1 \dots T - 1$

$$\delta_{t+1}(j) := \max_i \{ \delta_t(i) \alpha_{ij} \} b_j(O_{t+1})$$

$$\phi_{t+1}(j) := \arg \max_i \{ \lambda_t(i) \alpha_{ij} \}$$

3. Rekursionsabschluss

$$P^*(O|\lambda) = (P(O, s^*|\lambda) = \max_i \lambda_T(i) \alpha_T(i)$$

4. Rückverfolgung des Pfades

für alle Zeitpunkte $t, t = 1 \dots T - 1$

$$s_t^* = \phi_{t+1}(s_{t+1}^*)$$

Mit $\phi_t(j)$ wird ein “Rückwärtszeiger” definiert der für jedes entsprechende $\delta_t(j)$ entlang der partiellen Pfade den jeweils optimalen Vorgängerzustand speichert.

Training

Je nach Problemstellung müssen unterschiedliche Modelle eines HMM’s gewählt werden. Es ist bisher kein Verfahren bekannt das aufgrund einer Stichprobe ein Optimales Modell generieren kann. Die Anzahl der Zustände, die Wahl der Emissionsverteilungen sowie deren initialer Parameterwerte müssen nach eigenen Erfahrungen gewählt werden. Wenn dies geschehen ist kann das Modell in einem iterativen Prozess trainiert werden. Hierbei werden die Parameter einer Wachstumstransformation unterworfen. Ziel ist es das die Modellparameter so verändert werden das die Bewertung des veränderten Modells besser als die des Ausgangsmodells ist.

Zum trainieren eines Hidden Markov Model existieren diverse Algorithmen. Sie unterscheiden sich im wesentlichen durch die Verwendeten Qualitätsmaße zur Bewertung der Modellierungsgüte. Beim Baum-Welch-Algorithmus [Rab89] wird die Produktionswahrscheinlichkeit $P(O|\lambda)$ zur Bewertung genutzt. Beim Viterbi-Algorithmus [Vit06] und dem eng verwandten Segmental-k-means Algorithmus [JR90] nur die Wahrscheinlichkeit $(P(O, s^*|\lambda))$ der jeweils optimalen Zustandsfolge betrachtet [Fin03].

4 Zusammenfassung

4.1 Fazit

Wie schlägt sich ein HMM in der Praxis

4.2 Vergleich mit anderen Machine Learning Ansätzen

Wie arbeiten Neuronale Netze, SVM, k-Means und Co. im Vergleich?

Literatur

- [BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, pages 144–152, New York, NY, USA, 1992. ACM.
- [BP66] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Statist.*, 37(6):1554–1563, 12 1966.
- [Fin03] Gernot A. Fink. *Mustererkennung mit Markov-Modellen*. B. G. Teubner Verlag, 2003.
- [JR90] Biing-Hwang Juang and L. R. Rabiner. The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:1639–1641, 1990.
- [KHW13] U.M. Stocker K.-H. Waldmann. *Stochastische Modelle - Eine anwendungsorientierte Einführung*. Springer-Verlag, Berlin Heidelberg, 2013.
- [Mar13] A.A. Markov. Example of a statistical investigation of the text of eugene onegin illustrating the dependence between samples in chain. ein beispiel statistischer forschung am text eugen onegin zur verbindung von proben in ketten. eine vorlesung vor der physikalischmathematischen fakultät der königlichen akademie der wissenschaften st. petersburg, 23. januar 1913. Übersetzt von alexander y. nitussov, lioudmila voropai und david link, 1913.
- [Mar09] Stephen Marsland. *Machine Learning - An Algorithmic Perspective*. Chapman I& Hall, 2009.

- [MP88] Warren S. McCulloch and Walter Pitts. Neurocomputing: Foundations of research. chapter A Logical Calculus of the Ideas Immanent in Nervous Activity, pages 15–27. MIT Press, Cambridge, MA, USA, 1988.
- [Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.
- [Vit06] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor.*, 13(2):260–269, September 2006.

Anhang

A Anhang 1

Erklärung zur Abgabe einer Prüfungsleistung

Ich versichere, dass ich

- den „Leitfaden für gute wissenschaftliche Praxis im Studiengang MKI“¹ kenne und achte,
- die von mir eingereichten Dokumente und Artefakte selbständig ohne Hilfe Dritter verfasst habe,
- alle benutzten Quellen und Hilfsmittel - dazu zählen auch sinngemäß übernommene Inhalte, leicht veränderte Inhalte sowie übersetzte Inhalte - in Quellenverzeichnissen, Fußnoten oder direkt bei Zitaten angegeben habe,
- alle wörtlichen und sinngemäßen Zitate von Textstücken, Tabellen, Grafiken, Fotos, Quellcode usw. aus fremden Quellen als solche gekennzeichnet und mit seitengenaue Quellenverweisen versehen habe,
- die von mir eingereichten Dokumente und Artefakte noch nicht in dieser oder ähnlicher Form in einem anderen Kurs vorgelegt worden sind und ich
- alle nicht als Zitat gekennzeichneten Inhalte selbst erstellt habe.

Mir ist bekannt, dass unmarkierte und unbelegte Zitate und Paraphrasen Plagiate sind und nicht als handwerkliche Fehler, sondern als eine Form vorsätzlicher Täuschung der Prüfer gelten, da fremde Gedanken als eigene Gedanken vorgetäuscht werden mit dem Ziel der Erschleichung einer besseren Leistungsbewertung.

Mir ist bekannt, dass Plagiarismus

- die Standards guter wissenschaftlicher Praxis,
- den Leitfaden für gute wissenschaftliche Praxis im Studiengang MKI,
- die Studien- und Prüfungsordnung der Hochschule Reutlingen (§10 Täuschung und Ordnungsverstoß) sowie
- das Landeshochschulgesetz von Baden-Württemberg (§3 Wissenschaftliche Redlichkeit Abs. 5, §62 Exmatrikulation Abs. 3)

missachtet und seine

studienrechtlichen Folgen vom Nichtbestehen bis zur Exmatrikulation

reichen.

Nachname: _____

Vorname: _____

Matrikelnummer: _____

abgegeben zur Lehrveranstaltung: _____

für das Semester: _____

Datum, Ort: _____

Unterschrift: _____

¹ <https://bscwserv.reutlingen-university.de/bscw/bscw.cgi/d2871027/GWP.pdf>