

Formale Modelle: Hidden Markov Model

Paul Pasler, Reutlingen University

Zusammenfassung—Es geht um versteckte Ketten.

Keywords—Machine Learning, Markov Kette.

1 EINFÜHRUNG

HIER beginnt der Spaß!

1.1 Machine Learning

Was ist eigentlich Machine Learning?

1.2 Wahrscheinlichkeitsrechnung

Notwendige mathematische Grundlagen

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 "Eugen Onegin" von Alexander Pushkin durch.

Eine Markov Kette beschreibt einen zeitdiskreten 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. 1) mit den Zuständen S und mit den Übergangswahrscheinlichkeiten X_i an den Kanten

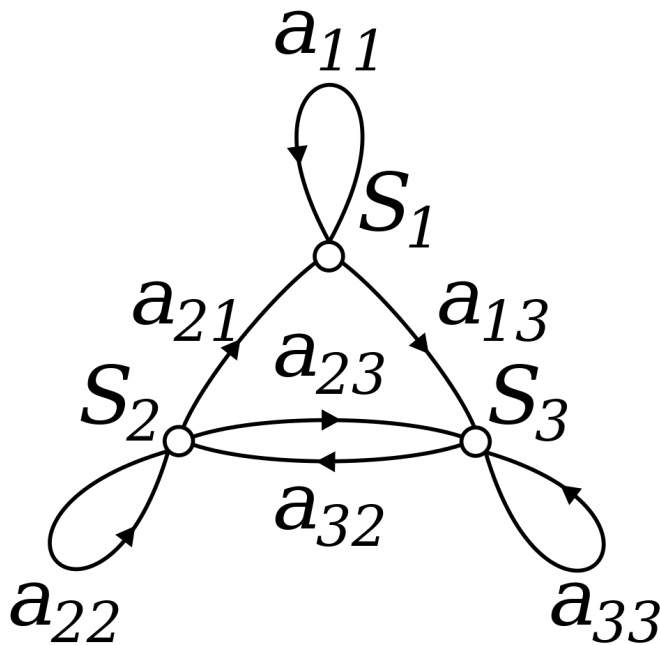


Abbildung 1. Einfaches Zustandsdiagramm einer Markov Kette (Quelle: de.wikipedia.org/wiki/Markov-Kette)

Die Wahrscheinlichkeit für k -Schritte lässt sich so ausrechnen:

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

Beispiel: Markov Kette für das Wetter ¹

Im folgenden Beispiel soll aufgrund des aktuellen Wetters auf das Wetter der 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 - "sunny" = 10%). Nach "rainy" liegt die Wahrscheinlichkeit jeweils bei 50% (siehe Abb. 2).

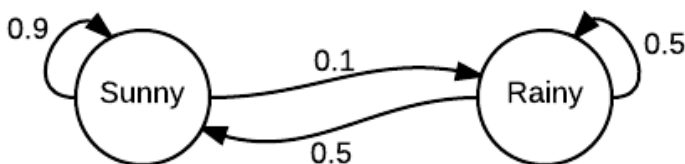


Abbildung 2. Gerichteter Zustandsgraph der modellierten Wetter-Markov Kette

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

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:

$$X_k = X_{k-1}A = X_0A^k = [1, 0] \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}^k$$

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.

In den folgenden beiden Abschnitten werden die Entstehung und Konzepte der MK und des HMM aufgezeigt. Im letzten Abschnitt werden die Ergebnisse evaluiert und ein Fazit mit Ausblick gezogen.

3.1 Hidden Markov Model und Gaussian Mixture Model

Der amerikanischen Mathematiker Leonard E. Baum (* 1931) und andere Autoren entwickelten auf Basis der Markov Kette Ende der sechziger Jahre das Hidden Markov Model. Erste Hidden Markov Model-Applikationen wurden zur Spracherkennung und später auch in der Bioinformatik zur Analyse von Nukleotid- und Proteinsequenzen eingesetzt.

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, deren 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 Models ableitet. Eine Folge von Emissionen wird auch Observationsfolge genannt.

3.2 Beispiel

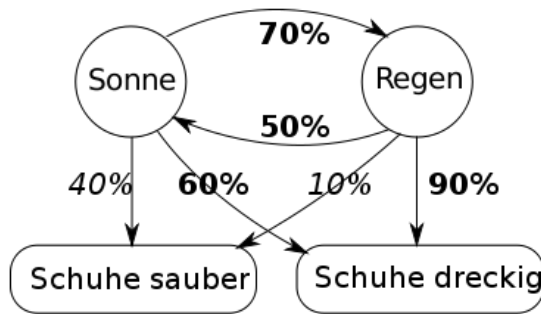


Abbildung 3. Hidden Markov Model für das Beispiel des gefangenen im Verlies

Abbildung 3 soll ein Hidden Markov Model an dem Beispiel des Gefangenen im Verlies darstellen². 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 Definition

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

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

- Endlich Menge von Zuständen
 $S = \{s | 1 \leq s \leq N\}$

- Alphabeth 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
 $B = \{b_{jk} | b_{jk} = P(O_t = o_k | S_t = j)\}$ 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 Observations 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]:

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

Der Approximationsfehler lässt sich hierbei über eine geeignete Anzahl von M Basisver-

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

teilungen klein halten. Somit ergibt sich für die Beschreibung der Emissionsverteilung eines Zustands des Hidden Markov Model folgende Formel:

$$b_j(x) = \sum_{k=1}^M c_{jk} g_{jk}(x) \quad (2)$$

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

3.4 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

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.

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, s_t = i | \lambda) \quad (3)$$

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.

$$\delta_t(i) = \max_{s_1, s_2, \dots, s_t} P(O_1, \dots, O_t, s_1, \dots, s_t = i | \lambda) \quad (4)$$

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 $t, t = 1 \dots T - 1$
 $\delta_{t+1}(j) := \max_i \{\delta_t(i) a_{ij}\} b_j(O_{t+1})$
 $\phi_{t+1}(j) := \arg \max_i \{\lambda_t(i) a_{ij}\}$
- 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 Models besser als die des Ausgangsmodels 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 [?] und dem eng verwandten Segmental-k-means Algorithmus [?] nur die Wahrscheinlichkeit ($P(O, s^*|\lambda)$) der jeweils optimalen Zustandsfolge betrachtet [?].



Michael Shell Biography text here.

John Doe Biography text here.

4 EVALUATION UND FAZIT

LITERATUR

- [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.
- [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.
- [KHW13] U.M. Stocker K.-H. Waldmann. *Stochastische Modelle - Eine anwendungsorientierte Einführung*. Springer-Verlag, Berlin Heidelberg, 2013.
- [Fin03] Gernot A. Fink. *Mustererkennung mit Markov-Modellen*. B. G. Teubner Verlag, 2003.

Jane Doe Biography text here.