Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Hidden-Markov-Modelle

Tobias Scheffer
Michael Großhans
Paul Prasse

Hidden-Markov-Modelle: Wozu?

- Spracherkennung:
  - Akustisches Modell.
- Geschriebene Sprache:
  - Part-of-Speech-Tagging,
  - Informationsextraktion.
- Biologie:
  - Finden von Genen in der DNA.

Markov-Prozesse

- $X_1, \ldots, X_n$: Zufallsvariablen.
- Gemeinsame Wahrscheinlichkeit:

$$P(X_1,\ldots,X_n) = P(X_1)\prod_{i=2}^{n} P(X_i \mid X_{i-1},\ldots,X_1)$$

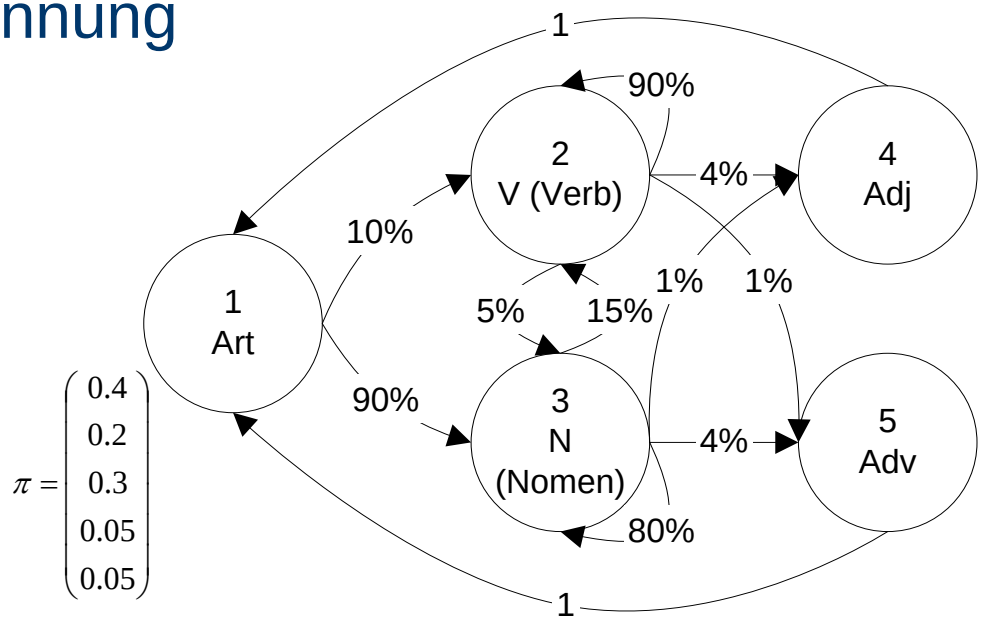
- $X_1, \ldots, X_n$ bilden Markovkette, wenn:
  - $P(X_1,\ldots,X_n) = P(X_1)\prod_{i=2}^{n} P(X_i \mid X_{i-1})$
  - Jede Variable X_i nur vom Vorgänger X_{i-1} abhängig.

- Markov-Modell:
  - Probabilistischer endlicher Automat.
  - Folge der Zustände ist Markov-Kette.

Markov-Modell

- Zustände $\Omega = \{1, \ldots, N\}$, Zustand zur Zeit t: q_t
- Transitionswahrscheinlichkeiten $A = \{a_{ij}\}$ mit $\sum_{j=1}^{N} a_{ij} = 1$
- Startwahrscheinlichkeiten $\pi = \{\pi_i = P(q_1 = i)\}$

- Z.B. Wortartenerkennung („Part-of-Speech Tagging“)

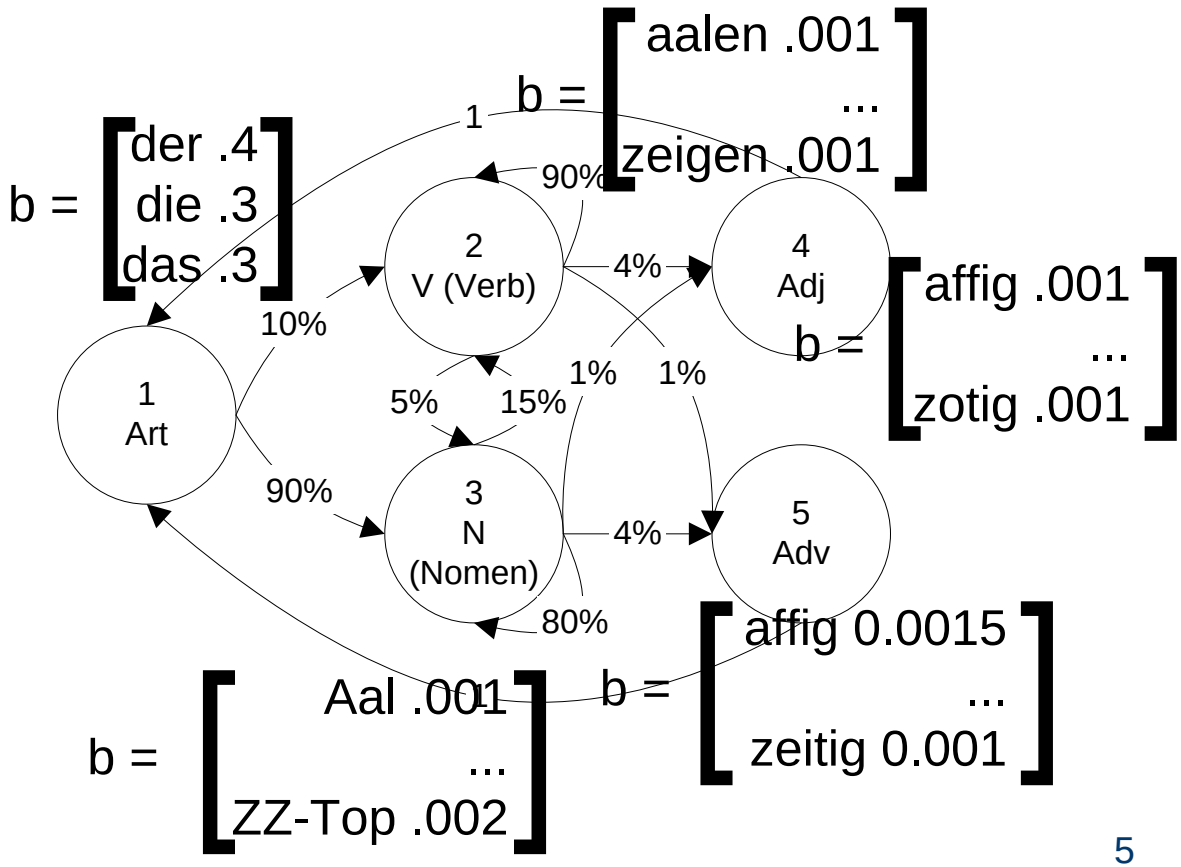


$$\pi = \begin{pmatrix} 0.4 \\ 0.2 \\ 0.3 \\ 0.05 \\ 0.05 \end{pmatrix}$$

- $P(q_1 = Art, q_2 = N, q_3 = V) = 0,4 \cdot 0,9 \cdot 0,15 = 0,054$

Hidden-Markov-Modell

- Folge der Zustände ist nicht sichtbar.
- Statt dessen: Zustände emittieren Beobachtungen O_t (mit Wahrscheinlichkeit $b_i(O_t)$).



Hidden-Markov-Modell

- Zustände $\Omega = \{1,\ldots,N\}$, Zustand zur Zeit t: q_t
- Transitionswahrscheinlichkeiten $A = \left\{a_{ij}\right\}$ mit $\sum_{j=1}^{N} a_{ij} = 1$
- Startwahrscheinlichkeiten $\pi = \{\pi_i = P(q_1 = i)\}$
- Beobachtungswahrscheinlichkeiten:
  $B = \{b_i(O_t) = P(O_t \mid q_t = i)\}$

- HMM definiert durch Parameter $\theta = (A, B, \pi)$

- Markov-Annahme für Zustandsfolgen:
  $P(q_t \mid q_{t-1},\ldots,q_1) = P(q_t \mid q_{t-1})$
- Markov-Annahme für Beobachtungen:
  $P(O_t \mid q_t, q_{t-1},\ldots,q_1) = P(O_t \mid q_t)$

Drei Basisprobleme

- Problem 1: Likelihood einer Beobachtungsfolge:
  - Berechne $P(O_1,...,O_T \mid \theta)$
- Problem 2: Optimale Zustandskette finden:
  - „Welche Zustandskette hat die Beobachtung am wahrscheinlichsten erzeugt?“
  - Berechne $\arg\max_{q_1,...,q_T} P(q_1,...,q_T \mid O_1,...,O_T,\theta)$
- Problem 3: Lernproblem
  - „Gegeben viele Beobachtungsfolgen, finde die Parameter des HMMs!“
  - Berechne $\arg\max_{\theta} P(\{(O_1,...,O_T),...\} \mid \theta)$

Problem 1: Likelihood einer Beobachtungsfolge

- $P(O_1,\ldots,O_T \mid \theta)$

$$= \sum_{q_1 \in \Omega} \cdots \sum_{q_T \in \Omega} P(O_1,\ldots,O_T \mid q_1,\ldots,q_T,\theta) P(q_1,\ldots,q_T \mid \theta)$$

Randverteilung & Produktregel

$$= \sum_{q_1 \in \Omega} \cdots \sum_{q_T \in \Omega} b_{q_1}(O_1) \cdot \ldots \cdot b_{q_T}(O_T) \cdot \pi_{q_1} \cdot a_{q_1 q_2} \cdot \ldots \cdot a_{q_{T-1} q_T}$$

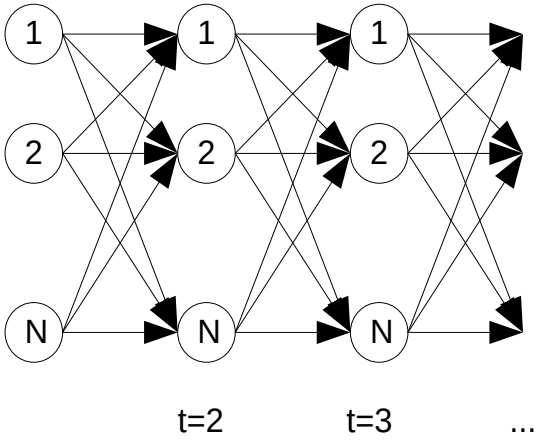
HMM-Definition

- Anzahl der Summanden: N^T
- Auswertung exponentiell in der Länge der Beobachtungsfolge.

- Gesucht: polynomieller Algorithmus.
- Dynamische Programmierung: Zwischenergebnisse speichern.

Trellis

- Trellis: Array über Zustände x Zeit.
- Hilft der Vorstellung, muss man aber nicht implementieren.



- Ziel: Rekursive Definitionen nutzen

Forward-Algorithmus

- Gesucht: $P(O_1,\ldots,O_T \mid \theta) = \sum_{q_T \in \Omega} P(O_1,\ldots,O_T, q_T \mid \theta)$

$$
\begin{aligned}
P(O_1,\ldots,O_T, q_T \mid \theta) & \\
&= \sum_{q_{T-1}\in\Omega} P(O_1,\ldots,O_T, q_T, q_{T-1} \mid \theta) && \text{Randverteilung} \\
&= \sum_{q_{T-1}\in\Omega} P(q_T, O_T \mid O_1,\ldots,O_{T-1}, q_{T-1}, \theta) P(O_1,\ldots,O_{T-1}, q_{T-1} \mid \theta) && \text{Produktregel} \\
&= \sum_{q_{T-1}\in\Omega} P(q_T, O_T \mid q_{T-1}, \theta) P(O_1,\ldots,O_{T-1}, q_{T-1} \mid \theta) && \text{Bedingte Unabhängigkeit} \\
&= \sum_{q_{T-1}\in\Omega} b_{q_T}(O_T)\, a_{q_{T-1}q_T} P(O_1,\ldots,O_{T-1}, q_{T-1} \mid \theta) && \text{HMM}
\end{aligned}
$$

- Anzahl der zu berechnenden Summanden:
  - T Rekursionstiefe mit jeweils N Summanden für N zu berechnende Wahrscheinlichkeiten $O(TN^2)$

Anwendungsbeispiel

- Worterkennung:
  - Ein HMM für jedes Wort.
  - „Welches Modell passt am besten?“
  - gegeben: Sprachsignal (Beobachtungssequenz)
    gesucht: Wahrscheinlichstes der gegebenen Modelle

$$\arg\max_k P(\theta_k \mid O_1,\ldots,O_T)$$

$$= \arg\max_k \sum_{q_T \in \Omega} P(O_1,\ldots,O_T, q_T \mid \theta_k) P(\theta_k)$$

Satz v. Bayes & Randverteilung

Likelihood durch Forward-Algorithmus in polynomieller Zeit berechenbar

A-Priori-Wahrscheinlichkeit durch Abzählen der Worthäufigkeit in der Trainingsmenge.

Drei Basisprobleme

- Problem 1: Likelihood einer Beobachtungsfolge:
  - $P(O_1,...,O_T \mid \theta)$ ist effizient durch Forward-Algorithmus berechenbar
- Problem 2: Optimale Zustandskette finden:
  - „Welche Zustandskette hat die Beobachtung am wahrscheinlichsten erzeugt?“
  - Berechne $\arg\max_{q_1,...,q_T} P(q_1,...,q_T \mid O_1,...,O_T,\theta)$
- Problem 3: Lernproblem
  - „Gegeben viele Beobachtungsfolgen, finde die Parameter des HMMs!“
  - Berechne $\arg\max_{\theta} P(\{(O_1,...,O_T),...\} \mid \theta)$

Problem 2: Optimale Zustandskette

- Beispiele:
  - Part-of-Speech-Tagging, ein Zustand pro Wortart
  - Gensequenzanalyse, Zustände entsprechen Tags, mit denen das Genom annotiert werden soll.
- Möglichkeit 1: Welcher einzelne Zustand zur Zeit t passt am besten zur Beobachtungsfolge?
  - Forward-Backward-Algorithmus für
    $\arg\max_{q_t} P(q_t \mid O_1, \ldots, O_T, \theta)$
- Möglichkeit 2: Welche komplette Zustandsfolge passt am besten zur Beobachtungsfolge?
  - Viterbi-Algorithmus für
    $\arg\max_{q_1,\ldots,q_T} P(q_1, \ldots, q_T \mid O_1, \ldots, O_T, \theta)$

Wahrscheinlichkeit eines Zustandes

$$P(q_t \mid O_1,\ldots,O_T,\theta)$$

$$= \frac{P(O_1,\ldots,O_T,q_t \mid \theta)}{P(O_1,\ldots,O_T \mid \theta)}$$

Produktregel

$$= \frac{P(O_{t+1},\ldots,O_T \mid O_1,\ldots,O_t,q_t,\theta)P(O_1,\ldots,O_t,q_t \mid \theta)}{\sum_{q_T \in \Omega} P(O_1,\ldots,O_t,q_t \mid \theta)}$$

Produktregel + Randverteilung

$$= \frac{P(O_{t+1},\ldots,O_T \mid q_t,\theta)P(O_1,\ldots,O_t,q_t \mid \theta)}{\sum_{q_T \in \Omega} P(O_1,\ldots,O_t,q_t \mid \theta)}$$

Bedingte Unabhängigkeit

??????????

Durch Forward-Algorithmus in polynomieller Zeit berechenbar

Durch Forward-Algorithmus in polynomieller Zeit berechenbar

Backward-Algorithmus

- Gesucht: $P\left(O_{t+1},\ldots,O_T \mid q_t,\theta\right)$

$$P\left(O_{t+1},\ldots,O_T \mid q_t,\theta\right)$$

$$= \sum_{q_{t+1}\in\Omega} P\left(O_{t+1},\ldots,O_T,q_{t+1} \mid q_t,\theta\right)$$ Randverteilung

$$= \sum_{q_{t+1}\in\Omega} P\left(O_{t+2},\ldots,O_T \mid O_{t+1},q_{t+1},q_t,\theta\right) P\left(O_{t+1},q_{t+1} \mid q_t,\theta\right)$$ Produktregel

$$= \sum_{q_{t+1}\in\Omega} P\left(O_{t+2},\ldots,O_T \mid q_{t+1},\theta\right) b_{q_{t+1}}\left(O_{t+1}\right) a_{q_t q_{t+1}}$$ Markov-Modell

- Aufwand um Terme $P\left(O_{t+1},\ldots,O_T \mid q_t,\theta\right)$ für alle Zeitpunkte $t=1,\ldots,T-1$ und Belegungen $q_t \in \Omega$ zu berechnen ist $O\left(TN^2\right)$

Wahrscheinlichkeit eines Zustandes

$$P(q_t \mid O_1,\ldots,O_T,\theta)$$

$$= \frac{P(O_1,\ldots,O_T,q_t \mid \theta)}{P(O_1,\ldots,O_T \mid \theta)}$$

Produktregel

$$= \frac{P(O_{t+1},\ldots,O_T \mid O_1,\ldots,O_t,q_t,\theta)P(O_1,\ldots,O_t,q_t \mid \theta)}{\sum_{q_T} P(O_1,\ldots,O_T,q_T \mid \theta)}$$

Produktregel + Randverteilung

$$= \frac{P(O_{t+1},\ldots,O_T \mid q_t,\theta)P(O_1,\ldots,O_t,q_t \mid \theta)}{\sum_{q_T} P(O_1,\ldots,O_T,q_T \mid \theta)}$$

Bedingte Unabhängigkeit

Durch Backward-Algorithmus in polynomieller Zeit berechenbar

Durch Forward-Algorithmus in polynomieller Zeit berechenbar

Durch Forward-Algorithmus in polynomieller Zeit berechenbar

- Term $P(q_t \mid O_1,\ldots,O_T,\theta)$ kann durch Forward-Backward-Algorithmus berechnet werden

Forward-Backward-Algorithmus

- Forward
  - Initialisiere $P(O_1, q_1 \mid \theta) = \pi_{q_1} b_{q_1}(O_1)$ (für alle Belegungen q_1)
  - Für t von 1 bis T
    - Berechne (für alle Belegungen q_t): $P(O_1, \ldots, O_t, q_t \mid \theta)$
- Berechne $P(O_1, \ldots, O_T \mid \theta) = \sum_{q_T} P(O_1, \ldots, O_T, q_T \mid \theta)$
- Backward
  - Initialisiere $P(O_T \mid q_{T-1}, \theta) = b_{q_T}(O_T) a_{q_{T-1} q_T}$
  - Für t von T-2 bis 1
    - Berechne (für alle Zustände q_t): $P(O_{t+1}, \ldots, O_T \mid q_t, \theta)$
    - Berechne (für alle Zustände q_t):

$$P(q_t \mid O_1, \ldots, O_T, \theta) = \frac{P(O_{t+1}, \ldots, O_T \mid q_t, \theta) P(O_1, \ldots, O_t, q_t \mid \theta)}{P(O_1, \ldots, O_T \mid \theta)}$$

Forward-Backward-Algorithmus

- Läuft mit quadratischem Aufwand: $O(TN^2)$
  - Rekursionstiefe für Forward/Backward: T
  - Für jede Rekursion t den Wert $P(q_t \mid O_1,...,O_T,\theta)$ aller N Belegungen von q_t berechnen
  - Für jeden Wert N Summen berechnen
- Berechnet $P(q_t \mid O_1,...,O_T,\theta)$ und $P(O_1,...,O_T \mid \theta)$
- Forward-Backward multipliziert viele Wahrscheinlichkeiten auf ⇨ numerisch oft 0
  - Mit Log-Wahrscheinlichkeiten arbeiten
    - Bsp: log(0,0001*0,001*0,00001) wird zu (-4)+(-3)+(-5)

Problem 2: Optimale Zustandskette

- Möglichkeit 1: Welcher einzelne Zustand zur Zeit t passt am besten zur Beobachtungsfolge?
  - $\arg\max_{q_t} P(q_t \mid O_1, \ldots, O_T, \theta)$
  - Bestimmung durch Forward-Backward-Algorithmus
- Möglichkeit 2: Welche komplette Zustandsfolge passt am besten zur Beobachtungsfolge?
  - $\arg\max_{(q_1, \ldots, q_T)} P(q_1, \ldots, q_T \mid O_1, \ldots, O_T, \theta)$
  - Bestimmen mit Viterbi-Algorithmus

Viterbi-Algorithmus

- Gesucht: $\arg\max_{q_1,\ldots,q_t} P(q_1,\ldots,q_t \mid O_1,\ldots,O_t,\theta)$

$$\arg\max_{q_1,\ldots,q_t} P(q_1,\ldots,q_t \mid O_1,\ldots,O_t,\theta)$$

$$= \arg\max_{q_1,\ldots,q_t} \frac{P(q_1,\ldots,q_t,O_1,\ldots,O_t \mid \theta)}{P(O_1,\ldots,O_t,\theta)}$$

Produktregel

$$= \arg\max_{q_1,\ldots,q_t} P(q_1,\ldots,q_t,O_1,\ldots,O_t \mid \theta)$$

Normalisierer hat keinen Einfluss auf argmax

$$= \arg\max_{q_t} \left(\arg\max_{q_1,\ldots,q_{t-1}} P(q_1,\ldots,q_t,O_1,\ldots,O_t \mid \theta) \right)$$

Operator argmax bzw. max

Mit dynamischer Programmierung in polynomieller Zeit berechenbar (nächste Folie)

Viterbi-Algorithmus

- Gesucht: $\max_{q_1,\ldots,q_{t-1}} P(q_1,\ldots,q_t,O_1,\ldots,O_t \mid \theta)$

$$\max_{q_1,\ldots,q_{t-1}} P\left(q_1,\ldots,q_t,O_1,\ldots,O_t \mid \theta\right)$$

$$= \max_{q_1,\ldots,q_{t-1}} P\left(q_1,\ldots,q_{t-1},O_1,\ldots,O_{t-1} \mid \theta\right) \cdot P\left(q_t,O_t \mid q_1,\ldots,q_{t-1},O_1,\ldots,O_{t-1},\theta\right)$$

Produktregel

$$= \max_{q_1,\ldots,q_{t-1}} P\left(q_1,\ldots,q_{t-1},O_1,\ldots,O_{t-1} \mid \theta\right) P\left(q_t,O_t \mid q_{t-1},\theta\right)$$

Bedingte Unabhängigkeit

$$= \max_{q_1,\ldots,q_{t-1}} P\left(q_1,\ldots,q_{t-1},O_1,\ldots,O_{t-1} \mid \theta\right) a_{q_{t-1}q_t} b_{q_t}\left(O_t\right)$$

HMM

$$= b_{q_t}\left(O_t\right) \max_{q_{t-1}} a_{q_{t-1}q_t} \left(\max_{q_1,\ldots,q_{t-2}} P\left(q_1,\ldots,q_{t-1},O_1,\ldots,O_{t-1} \mid \theta\right) \right)$$

Rekursion

Viterbi-Algorithmus

- Für t von 2 bis T-1
  - Berechne (für alle Belegungen q_t):
    - Maximale Wahrscheinlichkeit
      $$\max_{q_1,\ldots,q_{t-1}} P(q_1,\ldots,q_{t-1},q_t,O_1,\ldots,O_t \mid \theta)$$
    - sowie maximierenden Vorgängerzustand
      $$q_{t-1}(q_t) = \arg\max_{q_{t-1}} \left(\max_{q_1,\ldots,q_{t-2}} P(q_1,\ldots,q_{t-1},q_t,O_1,\ldots,O_t \mid \theta) \right)$$
- Termination
  $$q_T^* = \arg\max_{q_T} \left(\max_{q_1,\ldots,q_{T-1}} P(q_1,\ldots,q_{T-1},q_T,O_1,\ldots,O_T \mid \theta) \right)$$
- Für t von T-1 bis 1
  - $q_t^* = q_t\left(q_{t+1}^*\right)$
- Ausgabe der $q_1^*,\ldots,q_T^*$

Drei Basisprobleme

- Problem 1: Likelihood einer Beobachtungsfolge:
  - $P(O_1,...,O_T \mid \theta)$ ist effizient durch Forward-Algorithmus berechenbar
- Problem 2: Optimale Zustandskette finden:
  - Berechne $\arg\max_{q_1,...,q_T} P(q_1,...,q_T \mid O_1,...,O_T,\theta)$
  - Durch Viterbi- bzw. Forward-Backward-Algorithmus effizient lösbar.
- Problem 3: Lernproblem
  - „Gegeben viele Beobachtungsfolgen, finde die Parameter des HMMs!“
  - Berechne $\arg\max_{\theta} P(\{(O_1,...,O_T),...\} \mid \theta)$

Problem 3: Lernproblem

- Gegeben: Sammlung von Beobachtungsfolgen.
- Gesucht HMM-Parameter θ.
- Sichtbare Zustände
  - z.B. Part-of-Speech-Tagging: Jede Beobachtung ist mit dem zugehörigen Zustand markiert.
  - Schätzen der Parameter durch Zählen der Starthäufigkeiten, Transitionen, Beobachtungen.
- Unsichtbare Zustände
  - z.B. Worterkennung: Nur Sprachsignal gegeben, Zustandsfolgen sind unbekannt.
  - Lernen der Parameter durch Baum-Welch-Algorithmus.

Sichtbare Zustände

- Trainingsmenge

$$S = \left\langle (O^{(1)}, Q^{(1)}), \ldots, (O^{(m)}, Q^{(m)}) \right\rangle$$

$$O^{(k)} = O_1^{(k)}, \ldots, O_{T_k}^{(k)}$$

$$Q^{(k)} = q_1^{(k)}, \ldots, q_{T_k}^{(k)}$$

Notation

$$[\![q_1^{(k)} = i]\!] = \begin{cases} 0, & \text{wenn } q_1^{(k)} \neq i \\ 1, & \text{wenn } q_1^{(k)} = i \end{cases}$$

- Schätze $\pi_i = \frac{1}{m} \sum_{k=1}^{m} [\![q_1^{(k)} = i]\!]$
- Schätze $a_{ij} = \sum_{k=1}^{m} \sum_{t=1}^{T_k - 1} [\![q_t^{(k)} = i \wedge q_{t+1}^{(k)} = j]\!] \Big/ \sum_{k=1}^{m} \sum_{t=1}^{T_k - 1} [\![q_t^{(k)} = i]\!]$
- Schätze $b_i(O) = \sum_{k=1}^{m} \sum_{t=1}^{T_k} [\![q_t^{(k)} = i \wedge O_t^{(k)} = O]\!] \Big/ \sum_{k=1}^{m} \sum_{t=1}^{T_k} [\![q_t^{(k)} = i]\!]$

Unsichtbare Zustände

- Trainingsmenge

  $$S = \left\langle O^{(1)}, \ldots, O^{(m)} \right\rangle$$

  $$O^{(k)} = O_1^{(k)}, \ldots, O_{T_k}^{(k)}$$

  Zustände unbekannt.

- Forward-Backward-Algorithmus kann Wahrscheinlichkeit der Zustände berechnen, braucht dafür aber Modell.
- Letzte Folie: Können Modell schätzen, brauchen aber Zustandswahrscheinlichkeiten.

Baum-Welch-Algorithmus

- Lösung: Abwechselndes Schätzen von Modell und Zuständen

- Instanz des EM-Algorithmus.
- Beginne mit zufälligen Parametern und iteriere zwei Schritte bis zur Konvergenz
  - Schätze die Zustände durch Forward-Backward-Algorithmus auf Grundlage des aktuellen Modells
  - Optimiere die Parameter des Modells auf Grundlage berechneter Zustände.

Baum-Welch-Algorithmus

- Übergangswahrscheinlichkeiten:

$$P(q_t, q_{t+1} \mid O_1, \ldots, O_T, \theta) = \frac{P(q_t, q_{t+1}, O_1, \ldots, O_T \mid \theta)}{\sum_{q_t \in \Omega} P(O_1, \ldots, O_T, q_t \mid \theta)}$$

Durch Forward-Algorithmus in polynomieller Zeit berechenbar

$$\begin{aligned}
&P(q_t, q_{t+1}, O_1, \ldots, O_T \mid \theta) \\
&= P\left(O_{t+1}, \ldots, O_T \mid q_{t+1}, q_t, O_1, \ldots, O_t, \theta\right) P(q_{t+1}, q_t, O_1, \ldots, O_t \mid \theta) && \text{Produktregel} \\
&= P(O_{t+1} \mid q_{t+1}, q_t, O_1, \ldots, O_t, \theta) P(O_{t+2}, \ldots, O_T \mid q_{t+1}, q_t, O_1, \ldots, O_{t+1}, \theta) \\
&\quad \times P(O_1, \ldots, O_t, q_t \mid \theta) P(q_{t+1} \mid q_t, O_1, \ldots, O_t, \theta) && \text{Produktregel} \\
&= P(O_{t+1} \mid q_{t+1}, \theta) P(O_{t+2}, \ldots, O_T \mid q_{t+1}, \theta) \\
&\quad \times P(O_1, \ldots, O_t, q_t \mid \theta) P(q_{t+1} \mid q_t, \theta) && \text{Bedingte Unabhängigkeit} \\
&= b_{q_{t+1}}(O_{t+1}) P(O_{t+2}, \ldots, O_T \mid q_{t+1}, \theta) P(O_1, \ldots, O_t, q_t \mid \theta) a_{q_t q_{t+1}} && \text{HMM}
\end{aligned}$$

Durch Backward-Algorithmus in polynomieller Zeit berechenbar

Durch Forward-Algorithmus in polynomieller Zeit berechenbar

Baum-Welch-Algorithmus

- Trainingsmenge $S = \left\langle O^{(1)}, \ldots, O^{(m)} \right\rangle; O^{(k)} = O_1^{(k)}, \ldots, O_{T_k}^{(k)}$

1. Initialisiere θ zufällig.
2. Wiederhole bis Konvergenz: Für alle k von 1 bis m
   - Berechne durch Forward-Backward für alle t und q_t: $P(q_t \mid O_1, \ldots, O_T, \theta), P\left(O_1, \ldots, O_T, q_T \mid \theta\right), P\left(O_{t+1}, \ldots, O_T \mid q_t, \theta\right)$
   - Für alle t, q_t und q_{t+1} berechne $P(q_t, q_{t+1} \mid O_1, \ldots, O_T, \theta)$
   - Schätze $\pi_i^{(k)} = P(q_1 = i \mid O_1, \ldots, O_T, \theta)$
   - Schätze $a_{ij}^{(k)} = \sum_t P(q_t = i, q_{t+1} = j \mid O_1, \ldots, O_T, \theta) \Big/ \sum_t P(q_t = i \mid O_1, \ldots, O_T, \theta)$
   - Schätze $b_i^{(k)}(O) = \sum_{t: O_t = O} P(q_t = i \mid O_1, \ldots, O_T, \theta) \Big/ \sum_t P(q_t = i \mid O_1, \ldots, O_T, \theta)$
3. Mittle Schätzer für θ über m Beispiele und Wiederhole ab Schritt 2.

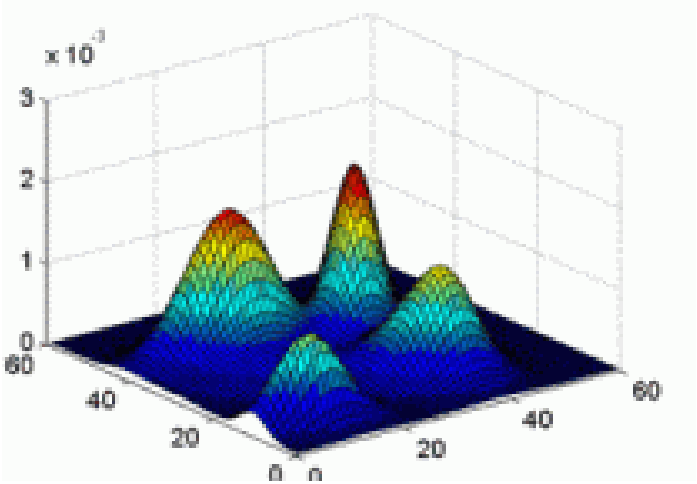
Drei Basisprobleme

- Problem 1: Likelihood einer Beobachtungsfolge:
  - $P(O_1,...,O_T \mid \theta)$ ist effizient durch Forward-Algorithmus berechenbar
- Problem 2: Optimale Zustandskette finden:
  - Berechne $\arg\max_{q_1,...,q_T} P(q_1,...,q_T \mid O_1,...,O_T,\theta)$
  - Durch Viterbi- bzw. Forward-Backward-Algorithmus effizient lösbar.
- Problem 3: Lernproblem
  - Berechne $\arg\max_{\theta} P(\{(O_1,...,O_T),...\} \mid \theta)$
  - Durch Baum-Welch-Algorithmus lösbar.

Kontinuierliche HMM

- Bisher: endlich viele Beobachtungen O, $b_i(O)$ diskrete Verteilung
- Sprachverarbeitung: Beobachtung kontinuierliches Signal, kontinuierliche Dichtefunktion $b_i(X)$.
- $b_i(X)$ Überlagerung von M multivariaten Gaußverteilungen.

$$b_i(x) = \sum_{k=1}^{M} c_{ik} N\left(x \mid \mu_{ik}, \Sigma_{ik}\right) = \sum_{k=1}^{M} c_{ik} b_{ik}(x)$$



Kontinuierliche HMM

- Bisher: $b_i^{(k)}(O) = \sum_{t:O_t=O} P(q_t = i \mid O_1,\ldots,O_T,\theta) \Big/ \sum_t P(q_t = i \mid O_1,\ldots,O_T,\theta)$
  - Stattdessen: ML-Schätzer der kontinuierlichen Verteilung.
  - Likelihood: $b_i(x) = \sum_{k=1}^{M} c_{ik} N\left(x \mid \mu_{ik}, \Sigma_{ik}\right) = \sum_{k=1}^{M} c_{ik} b_{ik}(x)$
  - Problem: sowohl die Zugehörigkeit c_{ik} der Beobachtungen als auch die M Verteilungen $N(\ldots)$ sind unbekannt.
- Schätzen einer multivariaten Gaußschen Mischverteilung wird durch EM-Algorithmus gelöst.
- In jeder Baum-Welch-Iteration und für jeden der N Zustände wird der EM-Algorithmus zur Schätzung der $b_i(X)$ gestartet.

Schätzen der Gaußschen Mischverteilung

- Hier: Vereinfachte Darstellung. Nur univariate ($x \in \mathbb{R}$) Mischverteilung. $b_{ik}(x) = N\left(x \mid \mu_{ik}, \sigma_{ik}\right)$
- Vereinfachte Darstellung: x_1 bis x_m seien die Beobachtungen, die im aktuellen Zustand i erzeugt wurden
  - tatsächlich wird jede Beobachtung O_t mit $P(q_t = i \mid O_1, \ldots, O_T, \theta)$ im Zustand i erzeugt).
- ML-Schätzer für univariate Normalverteilung:

$$\mu_{ML} = \frac{1}{m} \sum_{i=1}^{m} x_i \qquad \sigma_{ML} = \sqrt{\frac{\sum_{i=1}^{m} (x_i - \mu)^2}{m(m-1)}}$$

ML-Hypothese für mehrere normalverteilte Cluster

- Feste Clusterzuordnung: $z_{ij} = 1$, wenn x_i aus Cluster j stammt, sonst $z_{ij} = 0$.

$$\mu_j = \sum_{i=1}^{m} z_{ij} x_i \Big/ \sum_{i=1}^{m} z_{ij}$$

- Wenn x_i mit einer Wahrscheinlichkeit von $p(z_{ij}=1)$ aus Cluster j stammt:

$$\mu_j = \sum_{i=1}^{m} p(z_{ij} = 1) x_i \Big/ \sum_{i=1}^{m} p(z_{ij} = 1)$$

Probabilistische Clusterzuordnung

- Wahrscheinlichkeit dafür, dass x_i aus Cluster j stammt:
  - Cluster j hat a-priori Wahrscheinlichkeit $P(j)$

$$p(z_{ij} = 1) = P(j \mid x_i)$$

$$= \frac{N(x_i \mid \mu_j, \sigma_j) P(j)}{\sum_k N(x_i \mid \mu_j, \sigma_j) P(k)}$$

- Bei gleich wahrscheinlichen Clustern:

$$p(z_{ij} = 1) = P(j \mid x_i)$$

$$= \frac{N(x_i \mid \mu_j, \sigma_j)}{\sum_k N(x_i \mid \mu_j, \sigma_j)}$$

EM-Algorithmus (intuitiv)

- Beginne mit zufälligen Werten für alle
- Wiederhole
  - E-Schritt: Schätze für alle Datenpunkte i:

$$p(z_{ij}=1)=\frac{N(x_i \mid \mu_j,\sigma_j)P(j)}{\sum_k N(x_i \mid \mu_k,\sigma_k)P(k)}$$

  - M-Schritt: Optimiere alle μ_i,σ_i :

$$\mu_j=\frac{\sum_{i=1}^{m} p(z_{ij}=1)x_i}{\sum_{i=1}^{m} p(z_{ij}=1)};\sigma_j=\frac{\sqrt{\sum_{i=1}^{m} p(z_{ij}=1)(x_i-\mu)^2}}{\sum_{i=1}^{m} p(z_{ij}=1)}$$

Fragen?