

Blatt 12**Letzter Übungszettel**

Abgabe 8. Juli 2008 in der Vorlesung. Praktische Übungsaufgaben über PASS abgeben (<https://ml01.zrz.tu-berlin.de/~mikio/pass.pl?conf=blatt12.conf>). Verwende matlab (/home/ml/ml/bin), oder octave (<http://www.octave.org> frei verfügbar).

Aufgaben**1. Simulieren eines Hidden-Markov-Modells (20 Punkte)**

Schreibe eine Skript `hidden_coins.m`, welches das Werfen zweier Münzen simuliert, wobei nicht sichtbar ist, welche Münze aktuell verwendet wird.

Die Münzen sollen Kopf jeweils mit den Wahrscheinlichkeiten 0,9 und 0,5 erzeugen, und die Übergangswahrscheinlichkeiten zwischen der ersten zweiten Münze sind durch die Matrix

$$a = \begin{pmatrix} 0,1 & 0,9 \\ 0,5 & 0,5 \end{pmatrix}$$

(d.h. a_{ij} ist die Wahrscheinlichkeit von Münze i zu Münze j zu wechseln) gegeben.

Das Skript bekommt die Anzahl der zu erzeugenden Würfe übergeben und gibt zwei Vektoren zurück: einen, in dem die Münzwurfresultate enthalten sind (kodiert als 1 für Kopf und 0 für Zahl), sowie einen zweiten Vektor, der die gewählte Münze angibt (kodiert als 1 und 2).

2. Der Viterbi Algorithmus (20 Punkte) Implementiere den Viterbialgorithmus, der die wahrscheinlichste Abfolge von Zuständen für eine gegebene Sequenz und ein Modell berechnet. Schreibe eine Funktion `viterbi` mit der Signatur

$$q = \text{viterbi}(A, b, p, o),$$

die die Übergangswahrscheinlichkeiten A , Ausgabewahrscheinlichkeiten b , und Anfangsverteilung p , sowie die beobachtete Sequenz o übergeben bekommt und die Sequenz von Zuständen q zurückgibt.

Wende den Algorithmus auf Sequenzen der Längen 5, 10, und 20 an, die Du mit dem in Aufgabe 1 geschriebenen Skript erzeugst. Gib jeweils 3 Resultate für jede Länge in der Datei `states.txt` ab (jeweils zwei Zeilen pro Sequenz, erste Zeile “`true: 1 1 1 2 ...`”, zweite Zeile “`estimated: 1 1 2 2 ...`”).

(Für eine Zusammenfassung des Viterbi-algorithmus, siehe Rückseite.)

Viterbi-algorithmus (siehe auch Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, Vol. 77, No. 2, 1989, Seite 264.)

- Variablen: N Anzahl der Zustände
 M Größe des Ausgabealphabets
 T Länge der Sequenz
 a_{ij} Übergangswahrscheinlichkeiten
 $b_i(o)$ Wahrscheinlichkeit, in Zustand i Symbol o auszugeben
 π_i Wahrscheinlichkeit, in Zustand i zu beginnen.
 o_t Beobachtete Sequenz
 q_t^* wahrscheinlichste Zustandssequenz
 P^* Wahrscheinlichkeit für o_t gegeben q_t^*

- Initialisierung:

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N,$$

$$\psi_1(i) = 0.$$

- Rekursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \quad 2 \leq t \leq T, 1 \leq j \leq N$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, 1 \leq j \leq N.$$

- Terminierung:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)],$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)].$$

- Back-Tracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1.$$