



Übungsblatt 7

Ausgabe: 23.06.2016; Abgabe: bis 07.07.2016, 23:59 Uhr

Aufgabe 1 (Hidden Markov Model)

(20 Punkte)

Gegeben sei das folgende Hidden Markov Model, bei dem aufgrund der besseren Übersicht exemplarisch nur die Übergänge für die sichtbaren Zustände für den verborgenen Zustand G_4 eingezeichnet sind. Man muss sich solche Übergänge für alle Zustände G_1, \dots, G_n denken.

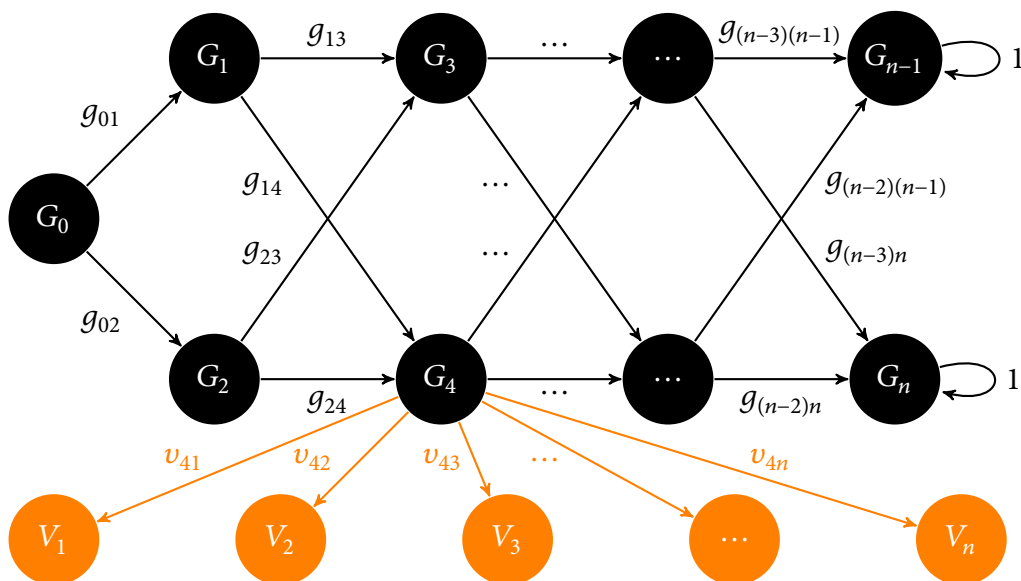


Abbildung 1 Schematische Darstellung des gegebenen HMM

Die Übergangsmatrix $G = (g_{ij})$ für die verborgenen Zustände mit den Übergangswahrscheinlichkeiten g_{ij} für Zustand G_i zu G_j der verborgenen Zustände und die für die sichtbaren Zustände $V = (v_{ij})$ mit den Übergangswahrscheinlichkeiten v_{ij} des Übergangs von G_i zu V_j stehen auf der nachfolgenden Seite.

$$G = \begin{bmatrix} 0 & 0.4 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 0 \end{bmatrix}.$$

Es sind dann folgende Anweisungen auszuführen:

- Man erstelle in Matlab eine geeignete Repräsentation des Hidden Markov Models. (Siehe dazu auch die unten stehenden Hinweise.)
- Man implementiere eine Funktion `forward(X, t, model)`, die zu einer gegebenen Sequenz $X = X_1, \dots, X_n$ an der t -ten Stelle den *Forward-Algorithmus* auswertet. (Wenn das Modell aus mehreren Variablen besteht, verändert sich die Signatur der Funktion entsprechend.) (4 Punkte)
- Man implementiere eine Funktion `backward(X, t, model)`, die zu einer gegebenen Sequenz $X = X_1, \dots, X_n$ an der t -ten Stelle den *Backward-Algorithmus* auswertet. (Für Modelle aus mehreren Variablen gilt derselbe Hinweis wie in der letzten Teilaufgabe.) (4 Punkte)
- Man implementiere eine Funktion `viterbi(X, model)`, die zu einer gegebenen Sequenz $X = X_1, \dots, X_n$ den *Viterbi-Algorithmus* auswertet. (Die Ausgabe ist hier keine Zahl, sondern ein Pfad.) (8 Punkte)

- e) Man bestimme $\mathbb{P}(G = G_9 | X = V_1 V_3 V_6 V_7 V_9)$. (1 Punkt)
- f) Man berechne $\mathbb{P}(G = G_9 | X = V_2 V_3 V_5 V_7 V_{10})$. (1 Punkt)
- g) Man bestimme die wahrscheinlichste Sequenz G_1, \dots, G_n , die die Sequenz $X = V_1 V_3 V_6 V_7 V_9$ hervorgerufen haben könnte. (2 Punkte)

Hinweise:

- Ist es für die Rekursion nötig, dann darf den Anfangs- und Endzuständen die Wahrscheinlichkeit 1 zuwiesen werden (wie in der Vorlesung erwähnt).
- Aufgabenteil d) ist nicht trivial, wenn Ihr nicht viel Erfahrung bei der Matlab-Programmierung habt. Überlegt Euch dazu, wie man sich während der Suche nach dem Maximum die Knoten merken kann.
- Um die Abtipparbeit der obigen Übergangswahrscheinlichkeiten zu reduzieren, wird das Matlab-Skript `createHmm.m` zur Verfügung gestellt.

Aufgabe 2 (Fragen zur Vorlesung)

(9 Punkte)

Man beantworte die folgenden Fragen:

- a) Aus der Vorlesung ist bekannt, dass die Wahrscheinlichkeit eines Punktes x durch den Ausdruck

$$p(x) \approx \frac{k/n}{V}$$

angenähert werden kann. Warum gilt für $n \rightarrow \infty$ mit $k_n \rightarrow \infty$ und $V_n \rightarrow 0$ für den Schätzer $p_n(x)$ die Identität

$$p_n(x) = \frac{k_n/n}{V_n} \rightarrow p(x)?$$

- b) Was ist die Strategie der Dichteschätzung bei dem *Parzenfenster-Verfahren* und bei dem *Nächster-Nachbar-Verfahren*?
- c) Inwiefern handelt es sich bei beiden in der letzten Aufgabenstellung genannten Verfahren um sogenannte *nicht-parametrische* Methoden? Sie werden gelegentlich auch *Prototypen-basierte Verfahren* genannt. Erkläre, weswegen dies sinnvoll ist.

Aufgabe 3 (Parzenfenster)

(10 Punkte)

Die Schätzung $p_n(x)$ einer Dichte für eine Normalverteilung bei einem Datensatz mit n Stichproben hat die Gestalt

$$p_n(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - x)^2}{2\sigma^2}\right).$$

- a) Plottet die Punkte aus $D = \{2, 4, 5, 8, 15\}$ und deren Beitrag zur Dichteschätzung. In dieselbe Grafik soll die über den Datensatz D geschätzte Dichtefunktion im Intervall $(0, 20)$ geplottet werden. Für die Fensterfunktion soll dabei $\sigma = 1$ verwendet werden.

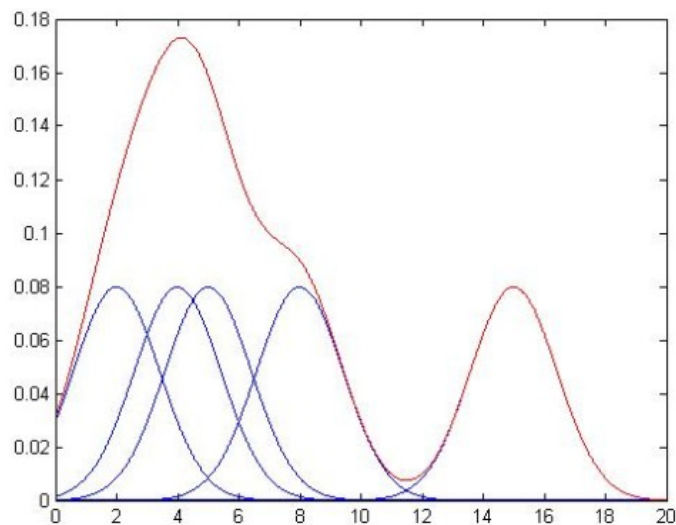


Abbildung 2 Dichteschätzung mit Parzenfenster und dem Beitrag der einzelnen Datenpunkte

- b) Man implementiere ein probabilistisches neuronales Netz (PNN) für die Datensätze D_1 und D_2 , die als .csv-Files vorliegen.
- c) Man klassifiziere die Punkte aus der Menge

$$M := \{(0, 0), (7, 7), (30, 20), (14, 40)\}$$

nach D_1 und D_2 .

