

Sprachverarbeitung: Übung 20

Erkennen diskreter Beobachtungssequenzen mit dem Viterbi-Algorithmus

In Übung 19 ging es darum, mit DDHMM (*discrete density hidden Markov models*) Beobachtungssequenzen zu generieren und mit dem Forward-Algorithmus zu erkennen. In dieser Übung geht es hauptsächlich um den Viterbi-Algorithmus, den man als Alternative zum Forward-Algorithmus zum Erkennen von Beobachtungssequenzen einsetzen kann. Zudem kann der Viterbi-Algorithmus auch gebraucht werden, um eine Beobachtungssequenz zu segmentieren.

Für diese Übung werden die folgenden DDHMMs gebraucht:

DDHMM(1): [ɸif] Zustand 1: Anfangszustand
Zustand 2: Laut [ɸ]
Zustand 3: Laut [i]
Zustand 4: Laut [f]
Zustand 5: Endzustand

$$A_1 = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.3 & 0.7 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

DDHMM(2): [ɸin] Zustand 1: Anfangszustand
Zustand 2: Laut [ɸ]
Zustand 3: Laut [i]
Zustand 4: Laut [n]
Zustand 5: Endzustand

$$A_2 = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.8 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

DDHMM(3): [ɸaf] Zustand 1: Anfangszustand
Zustand 2: Laut [ɸ]
Zustand 3: Laut [a]
Zustand 4: Laut [f]
Zustand 5: Endzustand

$$A_3 = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.9 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.9 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 & 0.9 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

DDHMM(4): [fiɸ] Zustand 1: Anfangszustand
Zustand 2: Laut [f]
Zustand 3: Laut [i]
Zustand 4: Laut [ɸ]
Zustand 5: Endzustand

$$A_4 = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.4 & 0.6 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

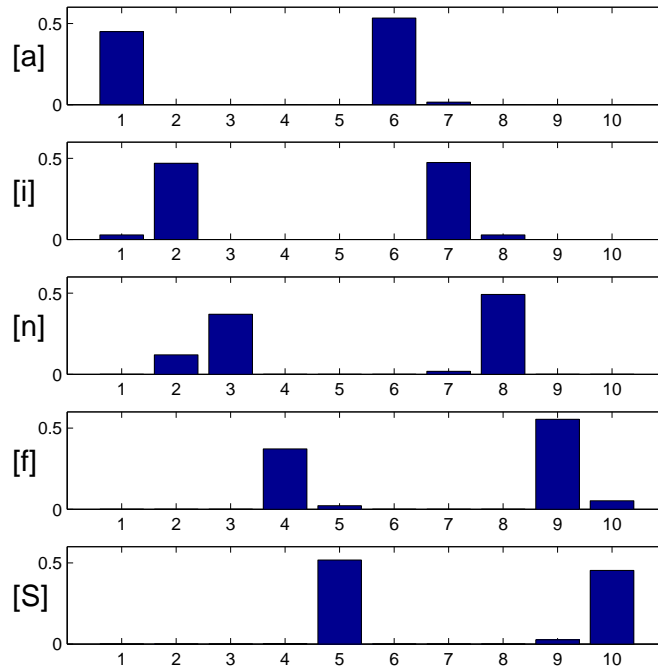
DDHMM(6): [ɸnif] Zustand 1: Anfangszustand
Zustand 2: Laut [ɸ]
Zustand 3: Laut [n]
Zustand 4: Laut [i]
Zustand 5: Laut [f]
Zustand 6: Endzustand

$$A_4 = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.7 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.7 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Diese DDHMM können Sie mit dem Befehl `load('ddhmm')` laden. Sie sind anschliessend im Array `ddhmm` verfügbar.¹ Jedes Element `j` des Arrays `ddhmm` hat die drei folgenden Felder:

<code>ddhmm(j).name</code>	Folge von Lauten in ETHPA-Notation (Matlab character array)
<code>ddhmm(j).a</code>	Zustandsübergangs-Wahrscheinlichkeitsmatrix ($N \times N$ -Array)
<code>ddhmm(j).b</code>	Beobachtungs-Wahrscheinlichkeitsdichten ($N \times M$ -Array)

Die diskreten Beobachtungs-Wahrscheinlichkeitsdichten der Laute sind in Figur 1 dargestellt.



Figur 1: Diskrete Beobachtungs-Wahrscheinlichkeitsdichten der Merkmale für die Laute [a], [i], [n], [f] und [S]

Aufgabe 1: Viterbi-Algorithmus

Schreiben Sie eine Matlab-Funktion `[p,optQ] = discr_viterbi_alg(a,b,X)`, welche für die Beobachtungssequenz `X` und ein DDHMM mittels des Viterbi-Algorithmus die Produktionswahrscheinlichkeit `p` entlang des optimalen Pfades berechnet. Diese Wahrscheinlichkeit wird im Folgenden kurz als Viterbi-Wahrscheinlichkeit² bezeichnet. Das DDHMM ist durch die Matrizen `a` und `b`, also die Zustandsübergangs- und die Beobachtungswahrscheinlichkeiten definiert. Die Viterbi-Wahrscheinlichkeit `p` und die Zustandssequenz des optimalen Pfades `optQ` sind die Ausgabeargumente der Funktion. Falls `p` gleich null ist, soll der optimale Pfad leer sein.

Um die realisierte Funktion zu überprüfen, können Sie das Matlab-Skript `ueb20_1` aufrufen, das Ihnen mitteilen wird, ob Sie die Aufgabe 1 als korrekt gelöst betrachten und zur nächsten Aufgabe gehen können.

¹In diesem Array sind noch weitere DDHMM vorhanden, die jedoch in dieser Übung nicht gebraucht werden.

²Im Kapitel 5.4.4 des Buches wird diese Wahrscheinlichkeit genauer als Verbundwahrscheinlichkeit des optimalen Pfades und der Beobachtungssequenz bezeichnet, also $P(\mathbf{X}, \hat{Q}|\lambda)$.

Aufgabe 2: Generierte vs. optimale Zustandssequenz

Bei dieser Aufgabe wird die Funktion `discr_viterbi_alg` von Aufgabe 1 gebraucht, um generierte Zustandssequenzen mit den optimalen aus dem Viterbi-Algorithmus zu vergleichen. Starten Sie dazu das Matlab-Skript `ueb20_2` und versuchen Sie durch Experimentieren bzw. Vergleichen der DDHMM-Spezifikationen die folgenden Fragen zu beantworten:

- a) Generieren Sie mit `ddhmm(2)` eine Beobachtungssequenz (die zugehörige Zustandssequenz wird auch angezeigt). Für die generierte Beobachtungssequenz und `ddhmm(2)` wird automatisch die optimale Zustandssequenz ermittelt und angezeigt. Wenn Sie dies wiederholen und jeweils die Zustandssequenzen vergleichen, dann werden Sie sehen, dass sie nicht immer übereinstimmen. Warum nicht?
- b) Wenn Sie denselben Versuch mit `ddhmm(1)` machen, dann werden die Zustandsfolgen immer übereinstimmen. Haben Sie auch dafür eine Erklärung?

Aufgabe 3: Viterbi- vs. Forward-Algorithmus

Mit dem Matlab-Skript `ueb20_3` können Sie den Viterbi- und den Forward-Algorithmus miteinander vergleichen. Starten Sie das Matlab-Skript `ueb20_3`. Wenn Sie auf eines der generierenden DDHMM klicken, dann wird mit diesem eine Beobachtungssequenz erzeugt und für alle erkennenden DDHMM die Forward- und die Viterbi-Wahrscheinlichkeit ermittelt und angezeigt. Die Schaltfläche des Modells mit der höchsten Forward- und/oder Viterbi-Wahrscheinlichkeit wird eingefärbt. Zudem wird die Erkennungsrate jedes Modells angegeben. Um eine grössere Anzahl von Generierungs- und Erkennungsversuchen zu machen, können Sie nach dem Wählen des Modells auf Start klicken.

Beantworten Sie die folgenden Fragen:

- a) Für welche der DDHMM liefern der Forward- und der Viterbi-Algorithmus immer die gleiche Wahrscheinlichkeit? Geben Sie eine Erklärung dafür.
- b) Warum können für die Wörter [Sif] und [Snif] generierte Beobachtungssequenzen nicht eindeutig dem richtigen Modell (nämlich `ddhmm(1)` bzw. `ddhmm(6)`) zugeordnet, also trotz unterschiedlicher Anzahl von Lauten miteinander verwechselt werden?
- c) Ist es möglich, dass mit `ddhmm(1)` generierte Beobachtungssequenzen mit dem Forward- und dem Viterbi-Algorithmus unterschiedlich klassiert werden?