Datenstruktur Trellis

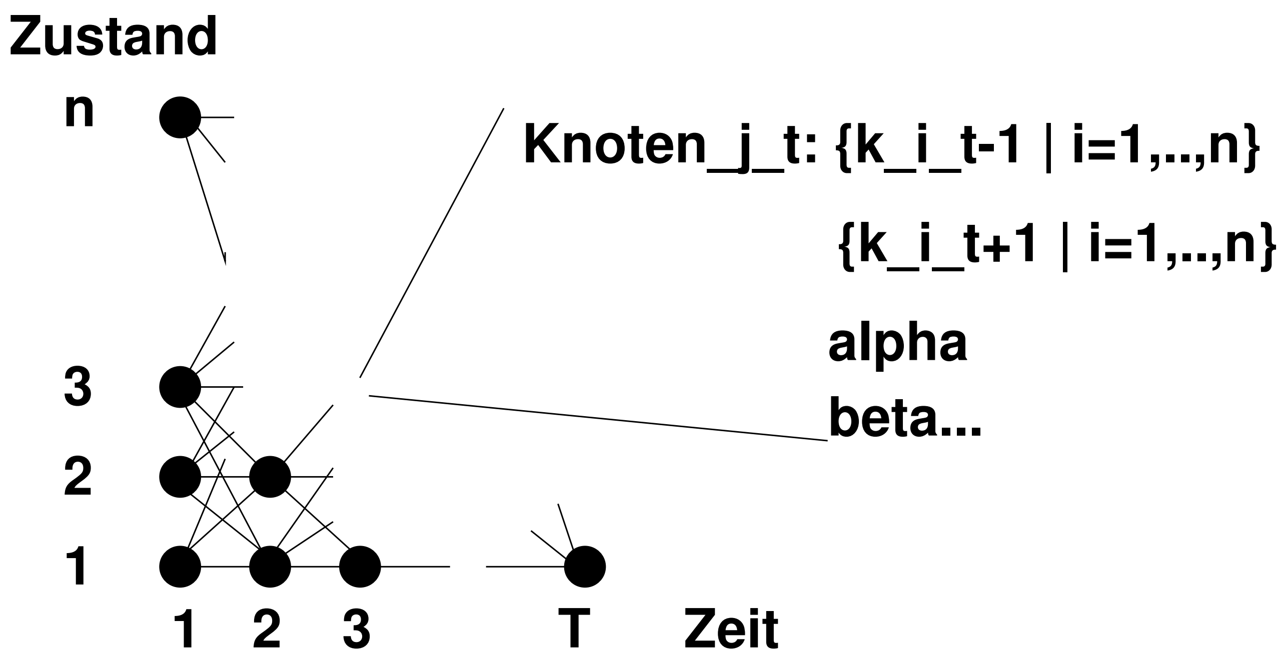
Ein Trellis ist ein Graph mit je einem Knoten für jeden Zustand an jedem Zeitpunkt.

Jeder Knoten zum Zeitpunkt t ist mit den Knoten zu den Zeitpunkten $t-1$ und $t+1$ verbunden.

Die Knoten werden mit Variablen attributiert, in denen Werte akkumuliert werden können (z.B. Vorwärtsvariablen).

Wenn eine passende Gleichung vorliegt, können aus den Werten für t die für $t=1$ berechnet werden.

Beispiel:



2. Bestimmung von $\mathrm{S}, s^*_{i_t}$

Def.: $\max_x f(x)$ ist der maximale Wert von $f(x)$

Def.: $\arg\max_x f(x)$ ist derjenige Wert von x, mit dem $f(x)$ maximal wird.

Viterbi-Algorithmus

Gesucht: Variablen δ mit:

$$\delta_t(i) = \max_{\mathbf{S}_{\leq t-1}} P(\mathbf{S}_{\leq t-1}, X_t = s_i; \mathbf{O}_{\leq t})$$

$$= \max_{s_{i1},..s_{i\,t-1}} P(X_1 = s_{i_1}, .., X_{t-1} = s_{i_{t-1}}, X_t = S_{i_t}; \mathbf{O}_{\leq t}),$$

der Wahrscheinlichkeit der wahrscheinlichsten Zustandsfolge von Zeit 1 bis t mit s_i am Zeitpunkt t und beobachteter Signalfolge $\mathbf{O}_{\leq t}$.

Der Vektor $\psi_t(i)$ bezeichnet den Vorgänger vom Zustand s_i im Pfad zu $\delta_t(i)$.

Unterschied zu Vorwärts-Algorithmus:

Maxima statt Summen.

Gleiche Komplexität $O(n^2T)$

Viterbi-Algorithmus

Es gilt:

$$\delta_1(i) = v_i \cdot a_{ik_1}, \qquad i = 1, .., n$$

$$\delta_t(j) = [\mathsf{max}_i \delta_{t-1}(i) \cdot p_{ij}] \cdot a_{jk_t}, \qquad t = 2, .., T, j = 1, .., n$$

$$\psi_t(j) = \mathsf{arg\ \ max}_{1 \le i \le n}(\delta_{t-1}(i) \cdot p_{ij}), \quad t = 2, .., T, j = 1, .., n$$

$$P^* = \mathsf{max}_i \delta_T(i)$$

$$s^*_{k_t} = \mathsf{arg\ \ max}_{1 \le i \le n} \delta_T(i)$$

$$s^*_{k_t} = \psi_{t+1}(s^*_{k_{t+1}}), \qquad t = 1, .., T-1$$

Viterbi-Algorithmus

Die Begründung für $\delta_t(j)$:

$$
\begin{aligned}
\delta_t(j) &= \text{max}_{\mathbf{S}_{\leq t-1}} P(\mathbf{S}_{\leq t-1}, X_t = s_i; \mathbf{O}_{\leq t}) \\
&= \text{max}_i \text{max}_{\mathbf{S}_{\leq t-2}} P(\mathbf{S}_{\leq t-2}, X_{t-1} = s_i, X_t = s_j; \mathbf{O}_{\leq t-1}, \eta_t = \sigma_{k_t}) \\
&= \text{max}_i \text{max}_{\mathbf{S}_{\leq t-2}} P(X_t = s_j; \eta_t = \sigma_{k_t} \mid \mathbf{S}_{\leq t-2}, X_{t-1} = s_i; \mathbf{O}_{\leq t-1}) \\
&\qquad \cdot P(\mathbf{S}_{\leq t-2}, X_{t-1} = s_i; \mathbf{O}_{\leq t-1}) \\
&= \text{max}_i \text{max}_{\mathbf{S}_{\leq t-2}} P(X_t = s_j \mid X_{t-1} = s_i) \cdot P(\eta_t = \sigma_{k_t} \mid X_t = s_j) \\
&\qquad \cdot P(\mathbf{S}_{\leq t-2}, X_{t-1} = s_i; \mathbf{O}_{\leq t-1}) \\
&= [\text{max}_i P(X_t = s_j \mid X_{t-1} = s_i) \cdot \text{max}_{\mathbf{S}_{\leq t-2}} P(\mathbf{S}_{\leq t-2}, X_{t-1} = s_i; \mathbf{O}_{\leq t-1})] \\
&\qquad \cdot P(\eta_t = \sigma_{k_t} \mid X_t = s_j) \\
&= [\text{max}_i p_{ij} \cdot \delta_{t-1}(i) \cdot a_{ik_t}
\end{aligned}
$$

Stochastisches Tagging

1. Einem Wort wird das wahrscheinlichste Tag, d.h. dasjenige, mit dem es im Trainingsset am häufigsten assoziert war, zugeordnet.

   Zu 90 % korrekt (Englisch, nach Allen 1995)

   Nachteil: kann unzulässige Folgen von Tags ergeben
2. Wahrscheinlichkeit einer Folge von Tags, Tag-N-Gramme, Viterbi-Algorithmus
3. HMMs, Kombination von Worthäufigkeit und Tag-Folgen-Wahrscheinlichkeit, Viterbi-Algorithmus; jedes Wort ist unabhängig von den anderen Wörtern, aber hängt von den n vorigen Tags ab.

Unbekannte Wörter:

- morphologische Information, Prefixe, Suffixe
- Menge von Default-Tags (offene Wortklassen), dann davon nach W'keit im N-Gramm davor auswählen
- W'keit aller Tags im Tagset am Ende des entsprechenden N-Gramms (ungeeignet für grosse Tagsets)

Tagging, Beispiel (deRose 1988)

2. Variante: (n-best) Tag-N-Gramme

"The man still saw her "

the	Art		
man	N	V	
still	Adv	N	V...
saw	N	V Pst	
her	PsPr	Pr Dat	

Häufigkeiten bei 4017 Bigrammen:

	N	Pr Dat	PsPr	Adv	V	V Pst	.
Art	186	0	0	8	1	8	9
N	40	1	3	40	9	66	186
Pr Dat	7	3	16	164	109	16	313
PsPr	176	0	0	5	1	1	2
Adv	5	3	16	164	109	16	313
V	22	694	146	98	9	1	59
V pst	11	584	143	160	2	1	91

1. Schritt, (Art) (eindeutig).

2. Schritt, möglich: Art - N und Art - V, beide expandieren (obwohl W'keit f. Art - V sehr gering).

3. Schritt, usw. besten Pfad zu jedem Tag aufheben, hier (Art - N - N), (Art - N - V) und (Art - N - Adv).

3. Parameterbestimmung für HMMs

Aufgabe: Bestimmung von Anfangsw'keiten $\mathbf{v}$, Übergangsw'keiten $\mathbf{P}$ und Signalw'keiten $\mathbf{A}$ für ein HMM.

1. Fall:

Annotierte Trainingsdaten liegen vor, d.h. Signal- und Zustandsfolge sind bekannt.

Dann können die relativen Häufigkeiten als Schätzwerte verwendet werden.

Problem: "sparse Data", spärliche Daten.

2. Fall:

Es gibt nur rohe Trainingsdaten, d.h. nur die Signalfolge ist bekannt.

Es werden Iterationsgleichungen definiert, die in jedem Schritt bessere Schätzwerte liefern.

Spärliche Daten

Problem: manche Phänomene sind so selten, dass sie im Trainingsset gar nicht auftreten.

Lösungen:

- *Smoothing* für Trigramme

  Bigramme und Unigramme hinzuziehen und gewichten:

  $$\begin{aligned} P(w_n \mid T_{n-2}, T_{n-1}) \approx & \quad \lambda_1 P_e(T_n) \\ & + \lambda_2 P_e(T_n \mid T_{n-1}) \\ & + \lambda_3 P_e(T_n \mid T_{n-2}, T_{n-1}) \end{aligned}$$

  mit $\lambda_1 + \lambda_2 + \lambda_3 = 1, \lambda_i > 0$, λ_3 “relativ“ hoch.

- ...

Parameterschätzungen durch Iteration

Baum-Welch-Verfahren oder Vorwärts-Rückwärts-Wiedereinschätzungsalgorithmus:

Definiere W'keiten:

$$\begin{aligned}\epsilon_t(i,j) &= P(X_t = s_i, X_{t+1} = s_j \mid \mathbf{O}) \\ &= \frac{P(\mathbf{O}; X_t = s_i, X_{t+1} = s_j)}{P(\mathbf{O})},\end{aligned}$$

d.h. gemeinsame W'keit, dass zum Zeitpunkt t Zustand s_i und zum Zeitpunk $t+1$ Zustand s_j vorlag unter der Bedingung, dass die Signalfolge $\mathbf{O}$ beobachtet wurde (von Zeit 1 bis T).

Es gilt:

$$\begin{aligned}\epsilon_t(i,j) &= \frac{\alpha_t(i) \cdot p_{ij} \cdot a_{jk_{t+1}} \cdot \beta_{t+1}(j)}{P(\mathbf{O})} \\ &= \frac{\alpha_t(i) \cdot p_{ij} \cdot a_{jk_{t+1}} \cdot \beta_{t+1}(j)}{\Sigma_{i,j} \alpha_t(i) \cdot p_{ij} \cdot a_{jk_{t+1}}}\end{aligned}$$

weiter: Baum-Welch

Weil:

$$
\begin{aligned}
& P(\mathbf{O}; X_t = s_i, X_{t+1} = s_j) \\
& = P(\mathbf{O}_{\leq t+1}, \mathbf{O}_{>t+1}; X_t = s_i, X_{t+1} = s_j) \\
& = P(\mathbf{O}_{\leq t+1}; X_t = s_i, X_{t+1} = s_j) \\
& \quad \cdot P(\mathbf{O}_{>t+1} \mid \mathbf{O}_{\leq t+1}; X_t = s_i, X_{t+1} = s_j) \\
& = P(\mathbf{O}_{\leq t+1}; X_t = s_i) \\
& \quad \cdot P(\eta_{t+1} = \sigma_{k_{t+1}}; X_{t+1} = s_j \mid \mathbf{O}_{\leq t+1}; X_t = s_i) \\
& \quad \cdot P(\mathbf{O}_{>t+1} \mid X_{t+1} = s_j) \\
& = P(\mathbf{O}_{\leq t+1}; X_t = s_i) \cdot P(X_{t+1} = s_j \mid X_t = s_i) \\
& \quad \cdot P(\eta_{t+1} = \sigma_{k_{t+1}} \mid X_{t+1} = s_j) \\
& \quad \cdot P(\mathbf{O}_{>t+1} \mid X_{t+1} = s_j) \\
& = \alpha_t(i) \cdot p_{ij} \cdot a_{j k_{t+1}} \cdot \beta_{t+1}(j)
\end{aligned}
$$

Diese ϵ heissen ebenfalls Vorwärts-Rückwärts-variablen, und hängen mit den γ wie folgt zusammen:

$$
\begin{aligned}
\gamma_t(i) & = P(X_t = S_i \mid \mathbf{O}) \\
& = \Sigma_{j=1}^{n} P(X_t = s_i, X_t + 1 = s_j \mid \mathbf{O}) \\
& = \Sigma_{j=1}^{n} \epsilon_t(i, j)
\end{aligned}
$$

weiter: Baum-Welch

Wir haben also:

- $\gamma_1(i)$ - W'keit, im Zustand s_i zu starten
- $\Sigma_{t=1}^{T-1}\epsilon_t(i,j)$ - erwartete Anzahl von Übergängen von s_i nach s_j
- $\Sigma_{t=1}^{T-1}\gamma_t(i)$ - erwartete Anzahl von Übergängen aus s_i
- $\Sigma_{t=1:\sigma_{k_t}=\sigma_j}^{T}\gamma_t(i)$ - erwartete Anzahl von Ausgaben des Signals σ_j im Zustand s_i

Die gesuchten Iterationsgleichungen:

$$\begin{aligned}
\overline{v}_i &= \gamma_1(i) \\
\overline{p}_{ij} &= \frac{\Sigma_{t=1}^{T-1}\epsilon_t(i,j)}{\Sigma_{t=1}^{T-1}\gamma_t(i)} \\
\overline{a}_{ij} &= \frac{\Sigma_{t=1:\sigma_{k_t}=\sigma_j}^{T}\gamma_t(i)}{\Sigma_{t=1}^{T}\gamma_t(i)},
\end{aligned}$$

wobei die Summen über i von 1 bis n der v_i, p_{ij} und a_{ij} immer 1 ergeben müssen.

Es kann gezeigt werden, dass $P(\mathbf{O} \mid (\mathbf{v}, \mathbf{P}, \mathbf{A}))$ mit jedem Schritt steigt oder die Parameter gleich bleiben.