



Vorlesung Maschinelles Lernen

Strukturelle Modelle – Conditional Random Fields

Katharina Morik

LS 8 Informatik
Technische Universität Dortmund

20.12. 2011



Gliederung

1 Einführung

2 HMM

3 CRF

- Strukturen der CRF
- Forward Backward Wahrscheinlichkeiten rechnen
- Viterbi Annotation einer Sequenz von Beobachtungen
- Bestimmen des Gewichtsvektors



Graphische Modelle

Lernen mit einer Struktur:

- Structural SVM
- Hidden Markov Models (HMM)
- Conditional Random Fields (CRF)
- Bayes Networks

Raum-zeitliche Verhältnisse

- Sprache
- Handlungen
- Bilder
- Geodaten

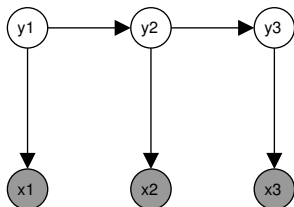


Hidden Markov Models

$$\lambda = (Y, A, B, \pi, X)$$

- Struktur gegeben durch einen Graph möglicher Zustände y_1, \dots, y_i, \dots (versteckte Knoten) Y
- Matrix A der Übergangswahrscheinlichkeit von einem Zustand zum nächsten (Transition Features): $a_{ij} = p(y_i|y_j)$
- Wahrscheinlichkeit B , im Zustand y_i die Beobachtung x_j zu machen (State Features) $b_{ij} = p(y_i|x_j)$
- Anfangswahrscheinlichkeit, dass y_i der Startzustand ist:
 $\pi(i) = p(y_1 = y_i)$
- Menge der möglichen Beobachtungen (beobachtete Knoten) X

HMM Modell



- Eine Beobachtung x_t hängt nur vom Zustand y_t ab: $p(x_t|y_t)$
- Ein Zustand y_t hängt nur vom Vorgängerzustand ab:
 $p(y_t|y_{t-1})$
- Die Anfangswahrscheinlichkeit schreiben wir als $p(y_1|y_0)$, so dass es von $p(y_t|y_{t-1})$ mit abgedeckt ist.
- Die Wahrscheinlichkeit für die Zustandssequenz \vec{y} und die Beobachtungssequenz \vec{x} ist

$$p(\vec{y}, \vec{x}) = \prod_{t=1}^T p(y_t|y_{t-1})p(x_t|y_t)$$



Aufgaben für Algorithmen

- Lernen der Wahrscheinlichkeiten $a_{ij}, b_{ij}, \pi(i)$ (Training Problem)
- Annotieren einer Beobachtungssequenz durch das (gelernte) Modell (Decoding problem)

In der Literatur gibt es noch die Evaluierungsaufgabe, die die Wahrscheinlichkeit dafür angibt, dass eine bestimmte Beobachtungssequenz durch eine gegebene Sequenz versteckter Zustände zustande kommt [2]. Wir besprechen hier zwei Algorithmen, die immer wieder vorkommen:

- Forward Backward
- Viterbi



Einführung in den Forward Backward Algorithmus

Es gibt eine Illustration des Algorithmus mit Excel-Berechnungen [1], die in den Algorithmus einführen soll.

Eiscreme und Klima (Eisner)

Im Jahr 2799 will eine Forschergruppe den Klimawandel untersuchen. Es werden keine Aufzeichnungen des Wetters in Baltimore gefunden, aber das Tagebuch von Jason Eisner über 33 aufeinander folgende Tage im Jahr 2001, in dem er notiert hat, wie viele Kugeln Eis er aß: $X = \{1, 2, 3\}$.

Gesucht ist für diese Sequenz von Tagen jeweils die verborgene Temperatur, hier nur kalt oder heiß: $Y = \{C, H\}$.



Wahrscheinlichkeiten

State

- Wahrscheinlichkeit an einem kalten Tag für eine Kugel
 $p(1|C) = 0,7$ und zwei $p(2|C) = 0,2$ und drei $p(3|C) = 0,1$.
- Wahrscheinlichkeit an einem heißen Tag
 $p(1|H) = 0,1$ und $p(2|H) = 0,2$ und $p(3|H) = 0,7$.

Transition

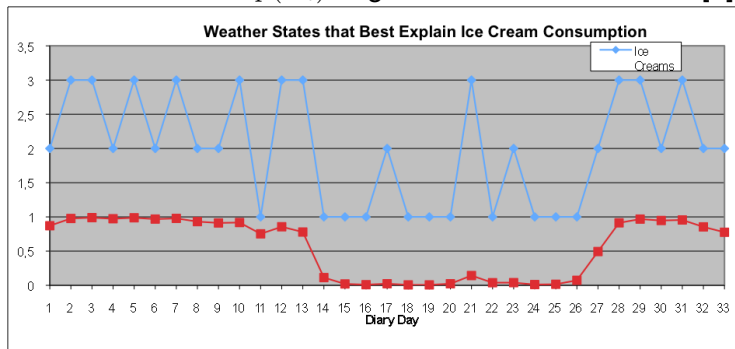
- Wahrscheinlichkeit von heiß nach kalt $p(C_{i+1}|H_i) = 0,1$
und von kalt nach kalt $p(C_{i+1}|C_i) = 0,8$
- Wahrscheinlichkeit von kalt nach heiß $p(H_{i+1}|C_i) = 0,1$
und von heiß nach heiß $p(H_{i+1}|H_i) = 0,8$

Boundary

- Wahrscheinlichkeit, das Tagebuch anzufangen/zu beenden
an einem kalten/heißen Tag $p(C|start) = p(H|start) = 0,5$
und $p(stop|C) = p(stop|H) = 0,5$

Wetterrekonstruktion

Wahrscheinlichkeit $p(H_i)$ angesichts des Eiskonsums [1]:





Rechnen der Wahrscheinlichkeit von Pfaden

- Wahrscheinlichkeit für $start, H, H, C$ und 2, 3, 3
Produkt von Übergangswahrscheinlichkeit und
Zustandswahrscheinlichkeit für jeden Übergang
 $(0,5 \cdot 0,2) \cdot (0,8 \cdot 0,7) \cdot (0,1 \cdot 0,1) = 0,1 \cdot 0,56 \cdot 0,01 = 0,00056$
- Welche der 8 Zustandsfolgen ist die wahrscheinlichste bei
der Beobachtung 2, 3, 3?
 $P(start, H, H, H) = 0,1 \cdot 0,56 \cdot 0,56 = 0,0314$ – diese!
 $P(start, H, C, H) = 0,1 \cdot 0,01 \cdot 0,07 = 0,00007$
 $P(start, C, H, H) = 0,1 \cdot 0,07 \cdot 0,56 = 0,0039$
 $P(start, C, C, H) = 0,1 \cdot 0,08 \cdot 0,007 = 0,000056$
- α : Summe der Wahrscheinlichkeiten aller Pfade von $start$
bis t , z.B. $y_t = H$



Forward Variable α

Pfad vom Anfang bis Position T , wo der Zustand irgendein festes y_i ist:

$$\alpha_t(i) = P(x_1, \dots, x_T, Y_t = y_i | \lambda)$$

- Anfang: π mal Wahrscheinlichkeit für erste Beobachtung

$$\alpha_1(i) = P(x_1, y_i | \lambda) = \pi_i b_i(x_1)$$

- Rekursion für $1 < t \leq T$:

$$\alpha_t(j) = P(x_1, \dots, x_t, y_j | \lambda) = \sum_{i=1}^{|Y|} \alpha_{t-1}(i) a_{ij} b_j(x_t)$$

- Ende:

$$P(x_1, \dots, x_T, y_T | \lambda) = \sum_{i=1}^{|Y|} \alpha_T(i)$$

Rechnen des Beispiels

Eisners Excel-Tabelle hat die Wahrscheinlichkeiten in den Matrizen A (Übergang) und B (Zustand) sowie Randverteilung bereits gegeben.

Wir schauen uns die Propagierung einmal an.

Danach betrachten wir CRFs und den Forward Backward Algorithmus für CRFs.

Was wissen Sie jetzt?

- Sie haben eine Formulierung für Sequenzen kennengelernt: HMM.
- Wir sind ein Beispiel durchgegangen, bei dem alle Wahrscheinlichkeiten schon gegeben waren.
- Sie wissen noch nicht, wie man diese Wahrscheinlichkeiten aus Daten heraus schätzt!

Conditional Random Fields

CRF=(X,Y,A,B)

- Die Struktur ist (wie bei HMMs) durch die multivariaten Zufallsvariablen X und Y gegeben, aber ungerichtet.
- Zustandsübergänge $a_{i,j,k}$ in A als *transition features* gewichtet: $\lambda_k f_k(y_{t-1}, y_t, x_t)$ wobei $\lambda \in [0, 1]$,
 $y_{t-1} = y_i, y_t = y_j, x_t = x_k$.
- Zustandsmerkmale $b_{i,k}$ in B als *state features* gewichtet:
 $\lambda f_k(y_t, x_t)$ wobei $y_t = y_i, x_t = x_k$.
- Gewichtung von Start- und Endzuständen für $y_0 = start$ bzw. $y_{T+1} = stop$ sind ebenfalls in A und B enthalten.
- Wir vereinigen die Merkmale und erhalten den Gewichtungsvektor Θ für die $k = 1, \dots, K$ Merkmale.

Die Beobachtungen müssen NICHT voneinander unabhängig sein!



Beispiel

Lexikon $X = \{Udo, goes, comes, to, from, Unna\}$

Zustände (Kategorien) $Y = \{Pos, Per, \emptyset\}$

Beobachtung: $x_1 = Udo, x_2 = goes, x_3 = to, x_4 = Unna$

$|X|$ Matrizen der Form $|Y| \times |Y|$

M_{Felix}	Pos	Per	\emptyset
Pos	1	2	3
Per	4	5	6
\emptyset	7	8	9

2:

$$\begin{aligned} &exp[\lambda_1 f_1(Pos, Per, Udo) + \\ &\lambda_2 f_2(Per, Udo) + \\ &\lambda_3 f_3(Per, capitalletter)] \end{aligned}$$

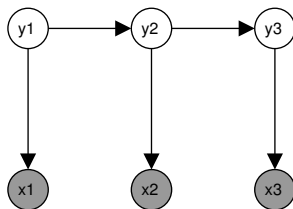
$$M_x = exp[\sum_k^K \lambda_k f_k(y_{t-1}, y_t, x_t) + \sum_k^K \lambda_k f_k(y_t, x_t)]$$

Potenzialfunktion rechnen: Für jedes x_t der beobachteten Sequenz die gesamte Matrix M_x ausrechnen!

CRF vs. HMM

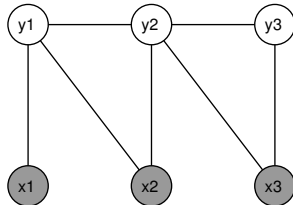
HMM:

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^3 p(y_t|y_{t-1})p(x_t|y_t)$$



CRF:

$$Z(\mathbf{x})^{-1} \prod_{t=1}^3 \exp [a_{y_{t-1}, y_t, x_t} + b_{y_t, x_t}]$$



Bemerkung: $y_0 = start, p(y|start) = \pi(y)$



CRF Modell

Normalisieren, wenn wir Wahrscheinlichkeit wollen, etwa:

$$Z(x) = M_1(x) \cdot \dots \cdot M_T(x)$$

$$Z(x) = \sum_y \exp\left[\sum_k \lambda_k f_k(y_{t-1}, y_t, x_t)\right]$$

$$\frac{1}{Z(x)} \prod_{t=1}^T \exp\left[\sum_k \lambda_k f_k(y_{t-1}, y_t, x_t) + \sum_k \lambda_k f_k(y_t, x_t)\right]$$

Warum Exponentialfunktion?



Exponentialfunktionen

Hammersley Clifford 1990

Eine lokale Verteilung mit Markov-Eigenschaft hat auch global die Markov Eigenschaft.

Eine Wahrscheinlichkeitsverteilung hat die Markov-Eigenschaft, gdw.

$$P(X) \propto \exp\left(\sum Q(C, X)\right)$$

wobei C die Menge aller Cliques im Graph ist und Q irgendeine reellwertige Funktion, die die likelihood für bestimmte Ausprägungen der Zufallsvariablen im Graph ausdrückt.

Die Exponentialfunktion liefert immer einen Wert > 0 .

Aufgaben für Algorithmen

- Training:
 - Wie oft kam t_0 in den Trainingsdaten mit $y = \emptyset$ vor?
 - Forward Backward Algorithmus* für *linear chain CRF* (i.e. y_t sind linear verbunden)
 - Gewichte lernen (parameter estimate) $\Theta = (\lambda_1, \dots, \lambda_K)$
 - Maximum Likelihood*
- Annotierung einer Beobachtungssequenz durch das gelernte Modell
 - Viterbi Algorithmus* für die wahrscheinlichste Annotation

Forward Backward Algorithmus

Sequenz $1, \dots, t, \dots, T$ und festes $y_t = y_i$

forward Wahrscheinlichkeit $\alpha_t(i)$ für Sequenz $1, \dots, t$

backward Wahrscheinlichkeit $\beta_t(i)$ für Sequenz t, \dots, T

- **Anfang:**

$$\alpha_1(i) = P(y_0 = \text{start}, y_1 = y_i, x_1)$$

$$\beta_T(i) = P(y_{T+1} = \text{stop}, y_T = y_i, x_T)$$

- **Rekursion für $1 < t < T$:**

$$\alpha_t(x) = \alpha_{t-1} \cdot M_t(x) =$$

$$\alpha_{t-1}(x) \cdot \exp\left[\sum_k^K \lambda_k f_k(y_{t-1}, y_t, x_t) + \sum_k^K \lambda_k f_k(y_t, x_t)\right]$$

$$\beta_t(x) = \beta_{t+1}(x) \cdot M_{t+1}(x) =$$

$$\beta_{t+1}(x) \cdot \exp\left[\sum_k^K \lambda_k f_k(y_t, y_{t+1}, x_t) + \sum_k^K \lambda_k f_k(y_{t+1}, x_t)\right]$$



Algorithmus

Für alle Beobachtungen $t=1$ bis T

für alle Label $i=1$ bis $|Y|$

berechne Forward $\alpha_t(y_i|x)$

berechne $Z(x)$

Für alle Beobachtungen $t=T$ bis 1

für alle Label $i=1$ bis $|Y|$

berechne Backward $\beta_t(y_i|x)$

$$O(T|Y|^2)$$

Für jeden Schritt (Beobachtung) $|Y|$ Nachrichten (α, β) rechnen
und jede Nachricht summiert über $|Y|$ Terme.



Annotation einer Sequenz

Wir wollen die Sequenz der Label als Vektor erhalten, \vec{y}_t .
Forward α wurde eben als Summe von Übergangs- und
Zustandswahrscheinlichkeiten gerechnet.

- Jetzt wird das maximale Label y vorwärts propagiert!

$$\alpha^*(y|x) =$$

$$\alpha_{t-1} \cdot \max_y \exp\left[\sum_k^K \lambda_k f_k(y_{t-1}, y_t, x_t) + \sum_k^K \lambda_k f_k(y_t, x_t)\right]$$

- dabei der wahrscheinlichste Vorgänger bestimmt:

$$\delta_t(y|x) =$$

$$\alpha_{t-1} \cdot \max_y \exp\left[\sum_k^K \lambda_k f_k(y_{t-1}, y_t, x_t) + \sum_k^K \lambda_k f_k(y_t, x_t)\right]$$

- Vom maximal wahrscheinlichen Label für den letzten Knoten $\delta_T(y|x)$ rechnet man rekursiv zurück und erhält die Sequenz.



Viterbi Algorithmus

Für alle Beobachtungen $t=1$ bis T

für alle Label $i=1$ bis $|Y|$

berechne MaxForward $\alpha_t^*(y_i|x)$

berechne wahrscheinlichsten Vorgänger: $\delta_t(y|x)$

$$\vec{y}_T = \max_y \alpha_t^*(y_i|x)$$

Für alle Beobachtungen $t=T-1$ bis 1

$$\vec{y}_t = \delta_t(y|x, \vec{y}_{t+1})$$

Bestimmen der Gewichte Θ

Gegeben: Trainingsmenge \mathcal{T}

$$\max_{A,B} \mathcal{L}(A, B | \mathcal{T}) \approx \max_{A,B} \prod_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} p(\mathbf{y}, \mathbf{x}), \text{ HMM}$$

$$\max_{A,B} \mathcal{L}(A, B | \mathcal{T}) \approx \max_{A,B} \prod_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} p(\mathbf{y} | \mathbf{x}), \text{ CRF}$$

Zielfunktion ist Konvex

⇒ Maximierung mit Gradientenabstieg oder Newton-Verfahren



Maximierung der Likelihood

Sei $\mathcal{T}(X_t = x)$ die Menge aller Trainingsbeispiele die an der t -ten Stelle der Sequenz die Beobachtung x enthalten.

Sei $1_{Y_{t-1}=y', Y_t=y}$ eine Indikatorfunktion die den Wert 1 annimmt gdw. die Sequenz an Position t im Zustand y und an Position $t - 1$ im Zustand y' ist.

$$a_{y',y,x}^{(t+1)} = a_{y',y,x}^{(t)} + \eta^{(t)} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}(X_t=x)} [1_{Y_{t-1}=y', Y_t=y} - p(y', y | x)]$$

$\eta^{(t)}$ ist eine Schrittweite.

Die Optimierung konvergiert für $\eta^{(t)} = \mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$

Update für $b_{i,k}$ analog.



Was wissen Sie jetzt?

- CRFs können beliebige Graphen sein. Die Beobachtungen werden unabhängig von der Reihenfolge genutzt. Gesucht ist $p(y|x)$.
- Wir haben die Übergangswahrscheinlichkeiten in der Matrix A und die Zustandswahrscheinlichkeiten in der Matrix B gesehen. Übergänge und Zustände werden als Merkmale aufgefasst.
- Die Merkmale sind jeweils gewichtet. Der Vektor aller Gewichte ist Θ und wird nach Maximum Likelihood bestimmt.
- Sie kennen die Matrizen M_x für jede Beobachtung x mit allen Zustandsübergängen.
- Der Forward Backward-Algorithmus berechnet die Wahrscheinlichkeiten. Sie wissen auch, wie!
- Der Viterbi-Algorithmus maximiert, um für eine Beobachtungssequenz die Label-Sequenz zu finden.



Jason Eisner.

An Interactive Spreadsheet for Teaching the Forward Backward Algorithm.

In: Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL

Dragomir Radev and Chris Brew (Eds.)

2002.

<http://cs.jhu.edu/~jason/papers/#tnlp02>



Lawrence R. Rabiner.

A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.

In: Proceedings of the IEEE

Vol. 77 No. 2

1989.



John Lafferty, Andrew McCallum, Fernando Pereira.

Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data.

In: Proceedings 18th Int. Conf. Machine Learning, 2001.



Charles Sutton, Andrew McCallum.

An Introduction to Conditional Random Fields for Relational Learning.

In: Introduction to Statistical Relational Learning.

Lise Getoor, Ben Taskar (eds). MIT Press, 2007.