Active Hidden Markov Models for Information Extraction

Seminar Informationsextraktion im WiSe 2002/2003

Madeleine Theile

Inhaltsübersicht

- Ziel
- formalere Aufgabenbeschreibung
- Theorie zu Hidden Markov Models
- Einsatz in der Informationsextraktion
  - Aufgabenbeschreibung
- aktives Lernen mit HMMs
  - Algorithmus für das Lernen von spärlich gelabelten Dokumenten
  - Erweiterung auf die Interaktion mit dem Benutzer

Ziel

- Informationsextraktion aus kaum vorverarbeiteten Dokumenten
- Lernen einer ’Funktion’, die ’Wörtern’ eines Dokuments semantische Labels zuweist
  - siehe Terrorismus-Beispiel aus einem vorherigen Seminarbeitrag
- Automatisierung durch Hidden Markov Models bei gering vorverarbeiteten Dokumenten
- aktives Lernen; Interaktion mit dem Benutzer
  $\rightarrow$ identifiziere die Tokens, deren Kennzeichnung der Automatisierung am dienlichsten sind

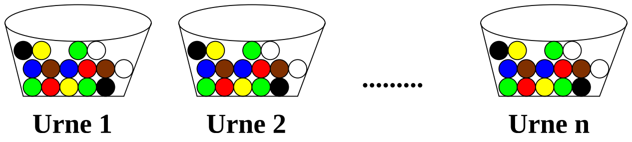
formalere Aufgabenbeschreibung

- Dokument besteht aus einer Folge von Beobachtungen (observations)
  - observation (IE: Token) = Vektor von Attributen
    z.B. Wortstamm, HTML-Kontext etc.
  - $O = (O_1, ..., O_T)$
    O_i entspricht einem Token im Dokument
- Aufgabe: Finde einen semantischen Tag $X_i \epsilon \{X_1, .., X_N, unknown, nolabel\}$ für die Beobachtung O_j bzw. für eine Folge von Beobachtungen finde eine Folge von Tags
- minimiere den Fehler, der bei der Zuweisung eines semantischen Labels zu einem Token auftritt

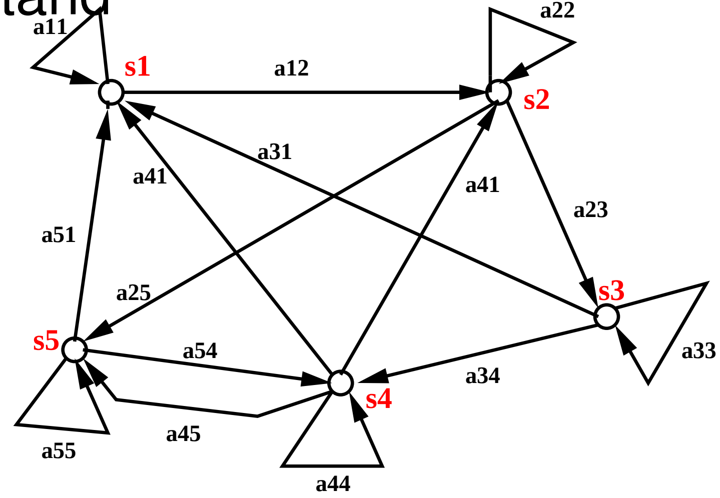
Und nun HMMs - Übersicht

- Motivation
- Definition
- verschiedene Aufgabenstellungen / Anwendungen
- Algorithmen
  - Viterbi
  - backward-forward
  - Baum-Welch

Motivation

- Urnen-Ball-Modell 

- HMMs modellieren stochastische Prozesse
  - Zustand bzw. Zustandsübergänge unbeobachtbar
  - jedoch Ausgabe beobachtbar
  - Ausgabe selbst ist auch ein stochastischer Prozess
  - Zustandskette ist gedächtnislos, d.h. Nachfolgezustand nur abhängig vom aktuellen Zustand
- Beispiele
  - Spracherkennung
  - Bioinformatik
    - DNA-Sequenzanalyse



Definition

- Anzahl N an (verborgenen) Zuständen (target states) $S_1, .., S_N$
  - es existiert auch noch eine Menge an Hintergrundzuständen, die wir speziell für den IE-Kontext benötigen
- Anzahl M an (physikalischen) Ausgaben, die je Zustand möglich sind
- Übergangswahrscheinlichkeit A zwischen zwei Zuständen i und $j : A = a_{ij}, \sum_{j=1}^{M} a_{ij} = 1$
- Ausgabewahrscheinlichkeit B eines Symbols k in einem Zustand $j : B = \{b_j(k)\}$
- Wahrscheinlichkeitsverteilung π für den Anfangszustand
- HMM beschreibbar durch $\lambda = (A, B, \pi)$
- Observationssequenz $O = O_1, .., O_T$ mit zugehöriger Zustandsequenz $q_1, ..., q_T \epsilon \{S_1, .., S_N\}$

Verschiedene Aufgaben für HMMs

- Typ 1: Wie kann man die Wahrscheinlichkeit ausrechnen, dass ein bestimmtes Modell eine gegebene Ausgabesequenz erzeugt hat?
- Typ 2: Gegeben ein Modell und eine Ausgabesequenz, wie kann man die dazugehörende Sequenz an versteckten Zuständen rekonstruieren?
  - Bewertung nach einem Optimalitätskriterium
- Typ 3: Wie kann man die Parameter eines HMM so trainieren, dass die Wahrscheinlichkeit, eine bestimmte Ausgabesequenz zu erzeugen, maximiert wird?
  - Trainingsbeispiele

Typ 1: Auswertung einer HMM

- Aufgabe
  - berechne die Wahrscheinlichkeit einer beobachteten Sequenz gegeben ein HMM $\lambda = (A, B, \pi)$
  - produziere Beispiel-Ausgabesequenzen $\rightarrow$ benötigt man, um z.B. im grossen Umfang Trainingsbeispiele herzustellen

1. wähle einen Anfangszustand $q_1 = S_i$, wobei $t = 1$
2. wähle der Ausgabewahrscheinlichkeit nach die wahrscheinlichste Ausgabe
3. $t = t + 1$
4. Übergang zu einem neuen Zustand $q_{t+1} = S_j$ nach der Übergangswahrscheinlichkeit
5. gehe zu 2. solange $t < T$

Beispiel zum Typ 1

- Wahrscheinlichkeit der Sequenz gegeben das Model berechnen
- Beispiel: Wetter
  - Zustand 1: Regen oder Schnee
  - Zustand 2: bewölkt
  - Zustand 3: sonnig
  - $A = a_{ij} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$
- das Wetter sei an Tag 1 sonnig
  Wie hoch ist dann die Wahrscheinlichkeit dafür, dass das Wetter in den nächsten 7 Tagen 'Sonne, Sonne, Regen, Regen, Sonne, bewölkt, Sonne' ist?
  - formaler: $O = \{S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$

Typ 2: Zustandssequenz einer HMM

- Aufgabe: finde eine Zustandssequenz, die die Beobachtungen erklärt
- Optimalitätskriterium: z.B. wähle den Zustand, der für sich genommen am wahrscheinlichsten ist $\rightarrow$ maximiert die erwartete Anzahl an korrekten Einzelzuständen
- dieser Typ hier kann nur bzgl. des Optimalitätskriteriums gelöst werden

Zustandssequenz: Beispiel

- gegeben: Beobachtungssequenz O = {wir, werden, geschickt}

HMM	Adj.	AuxVerb	Verb	Nomen	Partizip	geschickt	werden	wird	π
Adj.	0.2	0.1	0.1	0.4	0.2	0.2	0	0	0.3
AuxVerb	0.2	0.2	0.2	0.2	0.2	0	0.3	0	0.2
Verb	0.2	0.2	0.2	0.3	0.1	0	0.5	0	0.1
Nomen	0.1	0.4	0.3	0.1	0.1	0	0	0.2	0.3
Partizip	0.3	0.1	0.2	0.1	0.3	0.4	0	0	0.1

- gesucht: wahrscheinlichste Pfadsequenz
  $max \quad P(wir, werden, geschickt|\lambda)$
- Lösung: Viterbi-Algorithmus (dynamische Programmierung), um das single best state sequence-Problem zu lösen
  - Zeit $O(n|Q|^2)$, Platz $O(n|Q|)$
- alternativ: forward-backward Algorithmus

Optimierung

- Aufgabe: Maximiere die Wahrscheinlichkeit der Zustandssequenz, die die Ausgabe erzeugt hat, durch Anpassung der Parameter der HMM
  - nur lokales Optimum erreichbar
- dafür benötigen wir Trainingsbeispiele
- Algorithmen
  - backward-forward, Viterbi
    - nur anwendbar, wenn ein HMM gegeben
  - Baum-Welch Algorithmus
    - anwendbar bei nur teilweise gelabelten Sequenzen $\rightarrow$ werden wir zur Lösung des IE-Problems benötigen
    - benutzt auch backward-forward Algorithmus

forward-backward Methode (1)

- forward-Variable $\alpha_t(i)$: Wahrscheinlichkeit, dass zum Zeitpunkt t schon $O_1, .., O_t$ beobachtet wurde und sich das System im Zustand S_i befindet
- backward-Variable $\beta_t(i)$: Wahrscheinlichkeit, auch noch den Rest $O_{t+1}, .., O_T$ zu beobachten ausgehend vom Zustand S_i zum Zeitpunkt t
- Definition $\gamma_t(i)$: Wahrscheinlichkeit zum Zeitpunkt t im Zustand S_i zu sein gegeben die Observationssequenz O
ist von $\alpha_t(t)$ und $\beta_t(i)$ abhängig

forward-backward Methode (2)

1. Eingabe: beobachtete Sequenz O, ein HMM λ
2. Vorinitialisierung der forward/backward-Variable
3. für $t = 1..T$ und $i = 1..N$ sei $\alpha_{t+1}(j) = (\sum_{i=1}^{N} \alpha_t(i) a_{ij}) b_j(Ot+1)$
4. für $t = T..1$ und $i = 1..N$ sei $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
5. für $t = 1..T, i = 1...N$ sei $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$
   das Maximum von $\gamma_i(t)$ besagt dann, dass wir uns zum Zeitpunkt t im Zustand i befinden sollten, um den per-token Fehler zu minimieren

- so bauen wir eine Zustandsfolge zusammen

Aufgabenstellung: HMMs in der IE

- ein Hidden Markov Model soll alle Dokumente einer Domäne ’erklären’
  - finde semantische Bezeichner für Informationen im Text
  - z.B. Domäne Terrorismus: extrahiere aus dem Text die Informationen zum Tatort, Tatzeit etc.
- Eingabe: Observationssequenz
- Ausgabe: Hidden Markov Model, das die Informationen eines Dokuments in Template-Form giessen kann
- d.h. wir lernen einen Klassifizierer für semantische Labels

Rückblick-Folie

- Theorie von Hidden Markov Models
- verschiedene Aufgabenstellungen
  - Typ 1: Wahrscheinlichkeitsberechnung
  - Typ 2: Berechnung einer Zustandssequenz
- Informationsextraktion mit Hidden Markov Models
- was kommt heute noch?
  - Typ 3: Wie kann man die Parameter eines HMM so trainieren, dass die Wahrscheinlichkeit, eine bestimmte Ausgabesequenz zu erzeugen, maximiert wird?
  - aktives Lernen mit HMMs
  - Experimente

Informationsextraktion mit HMMs

1. Eingabe: Dokument, HMM λ, Menge an Tags
2. Tokenizer, part of speech tagger, parser etc.
3. Ausgabe: Observationssequenz $O = (O_1, ..., O_n)$
   - O_t besteht aus einem Vektor aus Worten und Attributen, der zu
     den Eigenschaften des Wortes korrespondiert
     Beispiel: O_1 = Verb, 1. Person Singular, Präsens, bedrohen
4. rufe die forward-backward-Prozedur auf, um den per-token Fehler zu minimieren
5. sei q_t^* das Maximum von $\gamma_t(i)$
   - *if* $q_t^* = S_i \epsilon \{S_1, .., S_N\}$ *(target states)* $\rightarrow$ *gib aus*
     $< X_i > O_t < /X_i >$
   - else q_t^* ist Element der Hintergrundzustände $\rightarrow$ gib einfach das Token O_t aus

Lernen eines Hidden Markov Models

- HMM $\lambda = (A, B, \pi)$ beschrieben durch ihre Parameter $\rightarrow$ Lernen der Parameter
- Baum-Welch Algorithmus
  - abgeleitet vom EM-Algorithmus
    E = Expectation, M = Maximization
- benötigen Trainingsdaten
- Grösse des Hypothesenraums ist fest $\rightarrow$ Modellparameter werden angepasst
- garantiert leider nur lokales Maximum

Baum-Welch Algorithmus

- Eingabe: Menge von Tokens $O^{(i)}$ aus Dokumenten der Trainingsmenge, Anzahl N der benötigten Ziel-Zustände

1. initialisiere die Parameter zufällig
2. benutze backward-forward, um Zustandswahrscheinlichkeit $\gamma_t(i)$ mit den aktuellen Parameter zu bestimmen $\rightarrow$ erhalten eine Zustandsfolge
3. zähle die Häufigkeit der Übergänge von S_i zu S_j $\rightarrow$ berechne daraus die neue Übergangswahrscheinlichkeit $a_{ij} = \frac{|Transisitonen_{S_i \rightarrow S_j}|}{|alleTransisitonen|}$
4. für jeden Zustand S_i zähle, wie oft eine Beobachtung O_t in diesem Zustand gemacht wird $\rightarrow$ berechne daraus die neue Ausgabewahrscheinlichkeit $b_i(O_t) = \frac{|Ausgabe_{O_t in S_i}|}{|alleAusgaben|}$
5. beginne wieder in Schritt 2 mit den neuen Parametern bis λ über eine Iteration (fast) konstant bleibt

- Ausgabe: Parameter $\lambda = (A, B, \pi)$

Neu: aktives Lernen mit HMMs (1)

- aktives Lernen als Ansatz aus dem Maschinellen Lernen bekannt; wird hier adaptiert und auf Hidden Markov Models angewandt
- Aufgabe: 'schwierige' Tokens identifizieren, die der Benutzer per Hand labeln soll $\Rightarrow$ grösstmöglicher Informationsgewinn
  - 'schwieriges Token' = Token, bei dem der Unterschied (Margin) zwischen dem wahrscheinlichsten semantischen Tag und dem zweitwahrscheinlichsten nicht allzu gross ist
- der Baum-Welch Algorithmus wird adaptiert, um diese Idee der schon vorgebenen semantischen Bezeichner zu implementieren
  - für alle schon vorgebenen semantischen Tags müssen die Ausgabe-Wahrscheinlichkeiten bzw. auch Übergangswahrscheinlichkeiten der einzelnen Ziel-Zustände im Algorithmus berücksichtigt werden

Neu: aktives Lernen mit HMMs (2)

- Eingabe: Menge von Tokens $O^{(i)}$ aus verschiedenen Dokumenten, Anzahl N der benötigten Ziel-Zustände

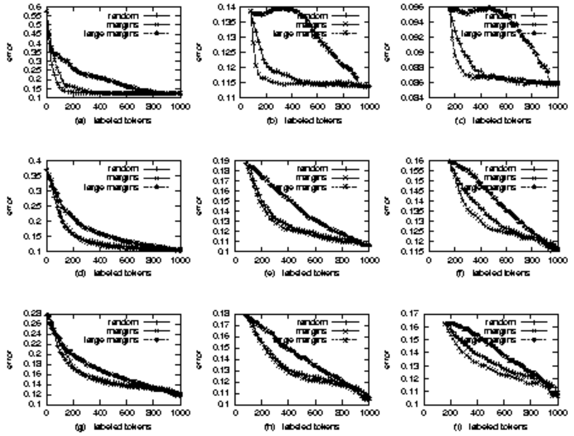
1. Baum-Welch Algorithmus, um initiale Parameter λ zu bestimmen
2. *for* $k = 2..\infty$ repeat
    - rufe für jedes Dokument die forward-backward Methode auf
    - für jedes Dokument und jedes Token bestimme den Margin zwischen den zwei wahrscheinlichsten Zuständen für ein Token
    - der Benutzer soll nun die ungelabelten Token mit dem kleinsten Margin mit einem semantischen Bezeichner versehen
    - berechne die neuen Parameter des HMM mit dem Baum-Welch Algorithmus
3. *until Benutzer hat keine Lust mehr, Beobachtungen zu labeln*

- Ausgabe: Parameter der HMM $\lambda_i = (\pi, a, b)$

Experimente (1)

- HMMs verschiedener Grösse, um automatisiert Beispiele zu erstellen
- auf diesen Trainingsbeispielen wird ein HMM trainiert, das auf Dokumenten einer Domäne die korrekten semantischen Tags labeln können soll
- Erfolg wurde empirisch gemessen
  $\rightarrow$ wie entwickelt sich der Fehler ?
  - HMMs verschiedener Grösse
    - Hintergrund-Zustände
    - Zielzustände
  - Labeling-Strategien
    - random
    - low margin $\rightarrow$ gelerntes Hidden Markov Model
    - high margin (Kontrollstrategie für den margin-Vergleich)

Experimente (2)



- HMMs
  - klein: 1 Hintergrundzustand, 2 Zielzustände
  - mittel: 10 Zustände (nicht näher spezifiziert)
  - gross: 15 Zustände
- 50 verschiedene Beobachtungs-Sequenzen mit 20 zu Beginn ungelabelten Beobachtungen

Experimente (3)

- wenn von Anfang an zu schwierige Tokens gewählt werden, so hat die random-Strategie mehr Erfolg
- erst in einer späteren Phase zahlt es sich aus, schwierigere Tokens mit kleinem Margin labeln zu lassen
- der Unterschied, wann der kleinste mögliche Fehler im Experiment erreicht wurde, ist für die random-Strategie und die margin-Strategie immens gross
  die margin-Strategie erreicht etwa doppelt so schnell (auf den Test-HMMs) die niedrigste Fehlerrate

Fragestellungen

- Wie gross sind precision und recall?
  - 60 -80 % auf Beispiele mit Flugreservierungen wurden erreicht
  - hier ?
- Wie gestalte ich den Wort-Vektor ?
  - klein / gross
  - Verben + Adverben
  - ist es sinnvoll, nur ein Wort zu betrachten?
  - ...

Vielen Dank für Eure Aufmerksamkeit

Q/A

Literatur

- ’Active Hidden Markov Models for Information Extraction’, Tobias Scheffer, Christian Decomain, Stefan Wrobel, Proceedings of the International Symposium on Intelligent Data Analysis, 2001
- ’A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition’, Lawrence L. Rabiner 1989
- ’Learning hidden Markov models for information extraction actively from partially labeled text’, Tobias Scheffer, Stefan Wrobel, Borislav Popov, Damyan Ognianov, Christian Decomain, and Susanne Hoche, In Künstliche Intelligenz 2/2002.