Sequence Labeling, Cont'd

Prof. Sameer Singh

CS 295: STATISTICAL NLP
WINTER 2017

February 2, 2017
Outline

- Marginal Inference in HMMs: F/B algorithm
- Maximum Entropy Markov Models
- Conditional Random Fields
- Neural Sequence Tagging
Outline

Marginal Inference in HMMs: F/B algorithm

Maximum Entropy Markov Models

Conditional Random Fields

Neural Sequence Tagging
**Expectation Maximization**

**Initialization**

- \( e(x_i | y_i) \sim \text{uniform} \)
- \( t(y_i | y_{i-1}) \sim \text{uniform} \)

**Label Data from the Model**

- Compute \( p(y | x) \)
- Fix \( e, t \)

**Update the Model from Data**

- Fix \( p(y | x) \)
- Update \( e, t \)

**K-Means**

- Pick K random centroids
- Cluster all the points
- Update centroids
Label Data from the Model

Hard-EM

$$y^* = \arg\max_y p(y|x) = \arg\max_y p(y, x)$$

$$\hat{e}(x; y_i) = \frac{\#x_i \land y_i \in y^*}{\#y_i \in y^*}$$

Soft-EM

$$p(y_i | x) = \sum_{y_j} p(y_j|x)$$

$$p(x_i, y_i) = \sum_{y_j} p(x_j=x_i, y_j=y_i)$$
Dynamic Programming

\[
\begin{align*}
    p(x_i, \ldots, x_n, y_i) &= \sum \sum p(x_i, \ldots, x_n, y_i, \ldots, y_n) \\
    &= p(x_i, \ldots, x_i, y_i) p(x_{i+1} \ldots x_n | y_i) \\
    \alpha(i, y_i) &= p(x_i, \ldots, x_i, y_i) = \sum p(x_i, \ldots, x_i, y_i, \ldots, y_i) \\
    &= \sum e(x_i | y_i) t(y_i | y_{i-1}) \alpha(i-1, y_{i-1}) \\
    \beta(i, y_i) &= p(x_{i+1} \ldots x_n | y_i) = \sum p(x_{i+1} \ldots x_n, y_{i+1} \ldots y_n) \\
    &= \sum e(x_{i+1}, y_{i+1}) t(y_{i+1} | y_i) \beta(y_{i+1})
\end{align*}
\]
Forward Backward Algorithm

Forward

\[ \alpha(0, s) = 1 \]
\[ \alpha(i, y_i) = \sum_{y_{i-1}} e(x_i | y_i) \cdot t(y_i | y_{i-1}) \cdot \alpha(i-1, y_{i-1}) \]

Backward

\[ \beta(n, e) = 1 \]
\[ \beta(i, y_i) = \sum_{y_{i+1}} e(x_{i+1} | y_{i+1}) \cdot t(y_{i+1} | y_i) \cdot \beta(i+1, y_{i+1}) \]
Updating the Model from Data

**Hard-EM**

\[
e(x_i | y_i) = \frac{\# x_i \text{ tagged as } y_i}{\# y_i}
\]

**Soft-EM**

\[
e(x_i | y_i) = \sum_j p(x_j = x_i, y_j = y_i | x_i, \ldots x_n)
\]

\[
p(x_i \ldots x_n, y_i) = \alpha(i, y_i) \beta(i, y_i)
\]

\[
p(x, \ldots x_n, y_i, y_{i+1}) = \alpha(i, y_i) t(y_{i+1} | y_i) e(x_{i+1} | y_{i+1}) \beta(i+1, y_{i+1})
\]
Outline

Marginal Inference in HMMs: F/B algorithm

Maximum Entropy Markov Models

Conditional Random Fields

Neural Sequence Tagging
Named Entity Recognition

George W. Bush spoke from the White House today.

PER PER PER O O O LOC LOC O O

B-PER I-PER I-PER O O O B-LOC I-LOC O O

B-PER I-PER E-PER O O O B-LOC E-LOC O O
Max. Entropy Markov Models

\[ p(y^* | x) = \prod_i p(y_i | x_i, y_{i-1}) \]

\[ p(y_i | x_i, y_{i-1}) = \frac{e^{\theta \cdot \phi(x_i, y_i, y_{i-1})}}{\sum_y e^{\theta \cdot \phi(x_i, y, y_{i-1})}} \]
Graphical Model Notation

HMMs

MEMMs
Adding Features (for POS)

- Word
- Lowercased word
- Prefixes
- Suffixes
- Capitalization
- Word shapes

**Current Word**
- the: the → DT
- Importantly: importantly → RB
- unfathomable: un- → JJ
- Surprisingly: -ly → RB
- Meridian: CAP → NNP
- 35-year: d-x → JJ

**Window Words**
- Add in previous / next word the ___
- Previous / next word shapes ___ X ___ X
- Occurrence pattern features [X: x X occurs]
- Crude entity detection ___ ..... (Inc. | Co.)
- Phrasal verb in sentence? put ..... ___
- Conjunctions of these things
Adding Features

\[
p(y_i|\mathbf{x}_i, y_{i-1}) = \frac{\Theta \cdot \phi(y_i, \mathbf{x}_i, y_{i-1})}{\sum_{y} e^{\Theta \cdot \phi(y, \mathbf{x}_i, y_{i-1})}}
\]
Predictions Using MEMMs

Greedy

\[ y_i = \arg \max_y p(y_i | x_i, s) \quad y_2 = \arg \max_y p(y_2 | x_2, y_1) \quad \ldots \]

Viterbi Decoding

\[ \overline{y}_i = \arg \max_y \prod_{i=1}^{n} p(y_i | x_i, y_{i-1}) \]

\[ \Pi(i, y_i) = \arg \max_{y_i, \ldots, y_{i-1}} \prod_{j=1}^{i-1} p(y_j | x_j, y_{j-1}) \]

\[ = \arg \max_{y_{i-1}} p(y_i | x_i, y_{i-1}) \Pi(i-1, y_{i-1}) \]
Training MEMMs

\[
\mathcal{L}(\theta, D) = \sum_{d \in D} \log P(y^d | x^d)
\]

\[
= \sum_{d} \sum_{i} \log P(y_i | x_i, y_{i-1})
\]

Independent

Train using off the shelf classifiers!
Outline

- Marginal Inference in HMMs: F/B algorithm
- Maximum Entropy Markov Models
- Conditional Random Fields
- Neural Sequence Tagging
Label Bias Problem

\[ P(B/B) = 0.5 \]
\[ P(B0O) = 0.5 \]

\[ P(B/B) = 1 \]
\[ P(1/"0", B) = 0.5 \]
\[ P(0/"0", B) = 0.5 \]

\[ P(B/"B", 1) = 0.5 \]
\[ P(0/"B", 0) = 0.5 \]

ignores "B"
Conditional Random Fields

\[ p(\hat{y} \mid \hat{x}) = \frac{e^{\Theta \cdot \Phi(\hat{x}, \hat{y})}}{\sum_{\hat{y}} e^{\Theta \cdot \Phi(\hat{x}, \hat{y})}} \]

\[ \Phi(\hat{x}, \hat{y}) = \sum_{i=1}^{n} \Phi_x(x_i, y_i) + \sum_{i=2}^{n} \Phi_t(y_i, y_{i-1}) \]

- \( \hat{x} \) \( \hat{y} \)
- \( \Phi \) features
- \( \Theta \) transition table

sometimes combined

\( \phi(\mathbf{x}_i, y_i, y_{i-1}) \)
Graphical Model Notation

HMMs/MEMMs

CRFs

locally normalized (score labels)

globally normalized (score seqs)
Predictions Using CRFs

\[ \hat{y} = \arg\max_y p(y|x) \]

\[ = \arg\max_y e^{\theta \cdot \phi(x,y)} = \arg\max_y \theta \cdot \phi(x,y) \]

\[ = \arg\max_y \theta \cdot \phi(x_i, y_i, y_{i-1}) = \arg\max_y \theta \cdot \phi(x_i, y_i, y_{i-1}) \]

\[ T_i(i, y_i) = \max_{y_i, \ldots, y_i} \sum_{j=1}^{i-1} \theta \cdot \phi(x_j, y_j, y_{j-1}) \]

\[ = \max_{y_i} \theta \cdot \phi(x_i, y_i, y_{i-1}) + T(i-1, y_{i-1}) \]
Likelihood Training of CRFs

$$L(\theta, D) = \log \prod_j P(\tilde{y}_j, \tilde{x}_j) = \sum_j \log \frac{e^{\theta \cdot \Phi(\tilde{x}_j, \tilde{y}_j)}}{\sum_{\tilde{y}} e^{\theta \cdot \Phi(\tilde{x}_j, \tilde{y})}}$$

$$= \sum_j \theta \cdot \Phi(\tilde{x}_j, \tilde{y}_j) - \log \sum_{\tilde{y}} e^{\theta \cdot \Phi(\tilde{x}_j, \tilde{y})}$$

$$\frac{\partial L(\theta, D)}{\partial \theta_k} = \sum_j \Phi_k(\tilde{x}_j, \tilde{y}_j) - \sum_{\tilde{y}} p(y|x_j) \Phi_k(\tilde{x}_j, y)$$
Likelihood Training of CRFs

\[
\sum_y \Phi_k (x, y) = \sum_y \sum_{i} \phi_k (x_i, y_i, y_{i-1}) = \sum_i \sum_{y_i, y_{i-1}} \phi_k (x_i, y_i, y_{i-1})
\]

\[
\sum_i \sum_{y_i, y_{i-1}} \phi_k (x_i, y_i, y_{i-1}) q_i (y_i, y_{i-1}) q_i (a, b) = \sum_y p(y|x) e^{\theta \Phi_k (x, y)}
\]

\[
\Psi (\bar{y}) = \sum_{y_i=a, y_{i+1}=b} \Phi (x, y)
\]

\[
q_i (a, b) = \frac{m_i (a, b)}{\sum_{a, b} m_i (a, b)}
\]
Forward-Backward Algorithm

\[ \alpha(i, y_i) = P(y_i \mid x_i \ldots x_i) = \sum_{y_i \ldots y_{i-1}} \psi(x_i, y_i, y_{i-1}) \prod_{j=1}^{i-1} \psi(x_j, y_j, y_{j-1}) \]

\[ = \sum_{y_{i-1}} \psi(x_i, y_i, y_{i-1}) \alpha(i-1, y_{i-1}) \]

\[ \beta(i, y_i) = P(y_i \mid x_{i+1} \ldots x_n) = \sum_{y_{i+1} \ldots y_n} \prod_{j=i+1}^{n} \psi(x_j, y_j, y_{j-1}) \]

\[ = \sum_{y_{i+1}} \psi(x_{i+1}, y_{i+1}, y_i) \beta(i+1, y_{i+1}) \]

\[ Z = \sum_{y} \alpha(h, y) \]

\[ \mu_j(a, b) = \alpha(j, a) \psi(x_j, b, a) \beta(j+1, b) \]
Outline

Marginal Inference in HMMs: F/B algorithm

Maximum Entropy Markov Models

Conditional Random Fields

Neural Sequence Tagging
Simple Neural Tagger

I love food.
MEMM-\textit{ish} Neural Tagger

\begin{itemize}
\item S
\item PRP
\item VB
\item NN
\item E
\end{itemize}

\begin{itemize}
\item Start
\item I
\item love
\item food
\item End
\end{itemize}
Recurrent Neural Tagger
Bidirectional RNN Tagger

I love food
Upcoming...

Homework

- Homework 2 is due (~10 days): **February 13, 2017**
- Write-up, data, and code for Homework 2 is up
- Ask questions early!

Proposal

- Proposal is due on Tuesday: **February 7, 2017**
- Only 2 pages

Summaries

- Paper summaries: **February 17, February 28, March 14**
- Only 1 page each