CSE574 - Administriva

• No class on Fri 01/25 (Ski Day)
Last Wednesday

• **HMMs**
  - Most likely individual state at time $t$: (forward)
  - Most likely sequence of states (Viterbi)
  - Learning using EM

• **Generative vs. Discriminative Learning**
  - Model $p(y,x)$ vs. $p(y|x)$
  - $p(y|x)$: don’t bother about $p(x)$ if we only want to do classification
Today

• **Markov Networks**
  - Most likely individual state at time t: (forward)
  - Most likely sequence of states (Viterbi)
  - Learning using EM

• **CRFs**
  - Model $p(y,x)$ vs. $p(y|x)$
  - $p(y|x)$: don’t bother about $p(x)$ if we only want to do classification
Finite State Models

Naïve Bayes

HMMs

Generative directed models

Logistic Regression

Linear-chain CRFs

General CRFs

Figure by Sutton & McCallum
Graphical Models

- Family of probability distributions that factorize in a certain way
- Directed (Bayes Nets)
- Undirected (Markov Random Field)
- Factor Graphs

Node is independent of its non-descendants given its parents

Node is independent all other nodes given its neighbors

\[ p(x) = \frac{1}{Z} \prod_{C} \Psi_C(x_C) \]

\( C \subset \{x_1, \ldots, x_K\} \) clique

\( \Psi_C \) potential function

\[ p(x) = \frac{1}{Z} \prod_{A} \Psi_A(x_A) \]

\( A \subset \{x_1, \ldots, x_K\} \)

\( \Psi_A \) factor function
Markov Networks

- Undirected graphical models

\[ P(X) = \frac{1}{Z} \prod_c \Phi_c(X) \]

\[ Z = \sum_X \prod_c \Phi_c(X) \]

- Potential functions defined over cliques

\[ \Phi(A, B) = \begin{cases} 
3.7 & \text{if } A \text{ and } B \\
2.1 & \text{if } A \text{ and } \overline{B} \\
0.7 & \text{otherwise} 
\end{cases} \]

\[ \Phi(B, C, D) = \begin{cases} 
2.3 & \text{if } B \text{ and } \overline{C} \text{ and } D \\
5.1 & \text{otherwise} 
\end{cases} \]
Markov Networks

• Undirected graphical models

A

B

C

D

• Potential functions defined over cliques

\[ P(X) = \frac{1}{Z} \exp \left( \sum_i w_i f_i(X) \right) \]

\[ Z = \sum_X \exp \left( \sum_i w_i f_i(X) \right) \]

\[ f(A, B) = \begin{cases} 1 & \text{if A and B} \\ 0 & \text{otherwise} \end{cases} \]

\[ f(B, C, D) = \begin{cases} 1 & \text{if B and } \overline{C} \text{ and D} \\ 0 & \text{otherwise} \end{cases} \]
Hammersley-Clifford Theorem

If Distribution is strictly positive \((P(x) > 0)\)
And Graph encodes conditional independences
Then Distribution is product of potentials over cliques of graph

Inverse is also true.
### Markov Nets vs. Bayes Nets

<table>
<thead>
<tr>
<th>Property</th>
<th>Markov Nets</th>
<th>Bayes Nets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>Prod. potentials</td>
<td>Prod. potentials</td>
</tr>
<tr>
<td>Potentials</td>
<td>Arbitrary</td>
<td>Cond. probabilities</td>
</tr>
<tr>
<td>Cycles</td>
<td>Allowed</td>
<td>Forbidden</td>
</tr>
<tr>
<td>Partition func.</td>
<td>$Z = ?$</td>
<td>$Z = 1$</td>
</tr>
<tr>
<td>Indep. check</td>
<td>Graph separation</td>
<td>D-separation</td>
</tr>
<tr>
<td>Indep. props.</td>
<td>Some</td>
<td>Some</td>
</tr>
<tr>
<td>Inference</td>
<td>MCMC, BP, etc.</td>
<td>Convert to Markov</td>
</tr>
</tbody>
</table>

*Slide by Domingos*
Inference in Markov Networks

- **Goal:** compute marginals & conditionals of

\[ P(X) = \frac{1}{Z} \exp \left( \sum_i w_i f_i(X) \right) \]

- **Example:**

What is \( P(x_i) \)?
What is \( P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N) \)?

- **Conditioning on Markov blanket is easy:**

\[
P(x \mid MB(x)) = \frac{\exp \left( \sum_i w_i f_i(x) \right)}{\exp \left( \sum_i w_i f_i(x = 0) \right) + \exp \left( \sum_i w_i f_i(x = 1) \right)}
\]

- **Gibbs sampling exploits this**
Markov Chain Monte Carlo

• **Idea:**
  - create chain of samples $x^{(1)}$, $x^{(2)}$, ...
    where $x(i+1)$ depends on $x(i)$
  - set of samples $x^{(1)}$, $x^{(2)}$, ... used to approximate $p(x)$

\[
\begin{align*}
  x^{(1)} &= (X_1 = x_1^{(1)}, X_2 = x_2^{(1)}, \ldots, X_5 = x_5^{(1)}) \\
  x^{(2)} &= (X_1 = x_1^{(2)}, X_2 = x_2^{(2)}, \ldots, X_5 = x_5^{(2)}) \\
  x^{(3)} &= (X_1 = x_1^{(3)}, X_2 = x_2^{(3)}, \ldots, X_5 = x_5^{(3)})
\end{align*}
\]
Markov Chain Monte Carlo

- **Gibbs Sampler**
  1. Start with an initial assignment to nodes
  2. One node at a time, sample node given others
  3. Repeat
  4. Use samples to compute $P(X)$

- **Convergence: Burn-in + Mixing time**

- **Many modes $\Rightarrow$ Multiple chains**

Slide by Domingos
Other Inference Methods

- Belief propagation (sum-product)
- Mean field / Variational approximations
Learning

• **Learning Weights**
  - Maximize likelihood
  - Convex optimization: gradient ascent, quasi-Newton methods, etc.
  - Requires inference at each step (slow!)

• **Learning Structure**
  - Feature Search
  - Evaluation using Likelihood, ...
Back to CRFs

- CRFs are conditionally trained Markov Networks
Linear-Chain
Conditional Random Fields

• From HMMs to CRFs

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

can also be written as

\[
p(y, x) = \frac{1}{Z} \exp \left( \sum_{t} \sum_{i,j \in S} \lambda_{ij} 1\{y_t = i\} 1\{y_{t-1} = j\} + \sum_{t} \sum_{i \in S} \sum_{o \in O} \mu_{oi} 1\{y_t = i\} 1\{x_t = o\} \right)
\]

(set \( \lambda_{ij} := \log p(y' = i | y = j) \), \( \ldots \))

We let new parameters vary freely, so we need normalization constant \( Z \).
Linear-Chain Conditional Random Fields

\[ p(y, x) = \frac{1}{Z} \exp \left( \sum_{t} \sum_{i,j} \lambda_{ij} 1_{y_t = i} 1_{y_{t-1} = j} + \sum_{t} \sum_{i \in S} \sum_{o \in O} \mu_{i,o} 1_{y_t = i} 1_{x_t = o} \right) \]

- **Introduce feature functions** \( f_k(y_t, y_{t-1}, x_t) \)
  - One feature per transition
  - One feature per state-observation pair

  \[ f_{ij}(y, y', x_t) := 1_{y = i} 1_{y' = j}, \quad f_{io}(y, y', x_t) := 1_{y = i} 1_{x = o} \]

  \[ p(y, x) = \frac{1}{Z} \exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right) \]

- Then the conditional distribution is

  \[ p(y|x) = \frac{p(y, x)}{\sum_{y'} p(y', x)} = \frac{\exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right)}{\sum_{y'} \exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_{t-1}, x_t) \right)} \]

This is a linear-chain CRF, but includes only current word’s identity as a feature.
Linear-Chain
Conditional Random Fields

- Conditional $p(y|x)$ that follows from joint $p(y,x)$ of HMM is a linear CRF with certain feature functions!
Linear-Chain
Conditional Random Fields

• Definition:

A **linear-chain CRF** is a distribution that takes the form

\[
p(y|x) = \frac{1}{Z(x)} \exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right)
\]

where \( Z(x) \) is a normalization function

\[
Z(x) = \sum_{y} \exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right)
\]
Linear-Chain Conditional Random Fields

• HMM-like linear-chain CRF

• Linear-chain CRF, in which transition score depends on the current observation
Questions

• #1 – Inference
  
  Given observations $x_1 \ldots x_N$ and CRF $\Theta$, what is $P(y_t, y_{t-1}|x)$ and what is $Z(x)$? (needed for learning)

• #2 – Inference
  
  Given observations $x_1 \ldots x_N$ and CRF $\Theta$, what is the most likely (Viterbi) labeling $y^* = \arg \max_y p(y|x)$?

• #3 – Learning
  
  Given iid training data $D=\{x^{(i)}, y^{(i)}\}$, $i=1\ldots N$, how do we estimate the parameters $\Theta=\{\lambda_k\}$ of a linear-chain CRF?
Solutions to #1 and #2

• Forward/Backward and Viterbi algorithms similar to versions for HMMs

• HMM as factor graph

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

\[
p(y, x) = \prod_{t=1}^{T} \Psi_t p(y_t, y_{t-1}, x_t)
\]

\[
\Psi_t(j, i, x) := p(y_t = j | y_{t-1} = i) p(x_t = x | y_t = j)
\]

• Then

\[
\alpha_t(i) = \sum_{i \in S} \Psi_t(j, i, x_t) \alpha_{t-1}(i)
\]

forward recursion

\[
\beta_t(i) = \sum_{j \in S} \Psi_{t+1}(j, i, x_{t+1}) \beta_{t+1}(j)
\]

backward recursion

\[
\delta_t(j) = \max_{i \in S} \Psi_t(j, i, x_t) \delta_{t-1}(i)
\]

Viterbi recursion
Forward/Backward for linear-chain CRFs ...

- ... identical to HMM version except for factor functions $\Psi_t(j, i, x_t)$
- CRF can be written as

$$p(y|x) = \frac{1}{Z} \exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right)$$

CRF Definition

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

$$\Psi_t(y_t, y_{t-1}, x_t) := \exp \left( \sum_{k} \lambda_k f_k(y_t, y_{t-1}, x_t) \right)$$

- Same:

$$\alpha_t(i) = \sum_{i \in S} \Psi_t(j, i, x_t) \alpha_{t-1}(i) \quad \text{forward recursion}$$

$$\beta_t(i) = \sum_{j \in S} \Psi_{t+1}(j, i, x_{t+1}) \beta_{t+1}(j) \quad \text{backward recursion}$$

$$\delta_t(j) = \max_{i \in S} \Psi_t(j, i, x_t) \delta_{t-1}(i) \quad \text{Viterbi recursion}$$
Forward/Backward for linear-chain CRFs

- **Complexity same as for HMMs**

\[
\text{Time: } O(K^2N) \quad \text{Space: } O(KN)
\]

Linear in length of sequence!

\[K = |S| \quad \text{#states}\]
\[N \quad \text{length of sequence}\]
Solution to #3 - Learning

- Want to maximize Conditional log likelihood

\[ l(\theta) = \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}) \]

- Substitute in CRF model into likelihood

CRFs typically learned using numerical optimization

(Also possible for HMMs, but we only discussed EM)

- Add Regularizer

Often large number of parameters, so need to avoid overfitting
Regularization

• **Commonly used $l_2$-norm (Euclidean)**
  - Corresponds to Gaussian prior over parameters

\[- \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}\]

• **Alternative is $l_1$-norm**
  - Corresponds to exponential prior over parameters
  - Encourages sparsity

\[- \sum_{k=1}^{K} \frac{|\lambda_k|}{\sigma}\]

• **Accuracy of final model not sensitive to $\sigma$**
Optimizing the Likelihood

• There exists no closed-form solution, so must use numerical optimization.

\[
l(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) - \sum_{i=1}^{N} \log Z(x^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}
\]

\[
\frac{\partial l}{\partial \lambda_k} = \sum_{i=1}^{N} \sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) - \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{y, y'} f_k(y, y', x_t^{(i)}) p(y, y'|x^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k}{\sigma^2}
\]

• \(l(\theta)\) is concave and with regularizer strictly concave

→ only one global optimum

Figure by Cohen & McCallum
Optimizing the Likelihood

• **Steepest Ascent**
  very slow!

• **Newton’s method**
  fewer iterations, but requires Hessian$^{-1}$

• **Quasi-Newton methods**
  approximate Hessian by analyzing successive gradients
  - BFGS
    fast, but approximate Hessian requires quadratic space
  - L-BFGS (limited-memory)
    fast even with limited memory!
  - Conjugate Gradient
Computational Cost

\[
l(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) - \sum_{i=1}^{N} \log Z(x^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}
\]

\[
\frac{\partial l}{\partial \lambda_k} = \sum_{i=1}^{N} \sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) - \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{y,y'} f_k(y,y', x_t^{(i)}) p(y', y'|x^{(i)}) - \sum_{k=1}^{K} \frac{\lambda_k}{\sigma^2}
\]

- For each training instance: \(O(K^2T)\) (using forward-backward)
- For \(N\) training instances, \(G\) iterations: \(O(K^2TNG)\)

Examples:
- Named-entity recognition 11 labels; 200,000 words < 2 hours
- Part-of-speech tagging 45 labels, 1 million words > 1 week
GEORGE E. BARRETT, CPA, AWARDED CERTIFICATE OF EDUCATIONAL ACHIEVEMENT IN EMPLOYEE BENEFIT ADMINISTRATION

Alloy, Silverstein, Shapiro, Adams, Mulford & Co., Cherry Hill, NJ, the 17th largest accounting firm with offices in the Philadelphia area, is pleased to announce that Associate Partner George E. Barrett, CPA, a Cherry Hill, NJ resident and 1983 graduate of Rutgers University, has been awarded a certificate of educational achievement in employee benefit administration from the Pennsylvania Institute of Certified Public Accountants. The certificate was awarded in recognition of Mr. Barrett's completion of a program which includes a series of seminars and comprehensive examinations.

Alloy, Silverstein, Shapiro, Adams, Mulford, & Co., which celebrates its 40th anniversary in 1999, provides a wide range of services including accounting, auditing, tax, management consulting, financial and estate planning, business valuations, litigation support and information technology.

For more information contact:

Reynold P. Cicalese, CPA
Alloy, Silverstein, Shapiro, Adams, Mulford & Co.
900 Kings Highway North
Cherry Hill, NJ 08034-1561
609.667.4100 extension 133
Person name Extraction

After record success last year (more than $119,000 was raised for the animals) all four co-persons decided to continue in their positions. The chairmen are Katie Cunningham, Marti Huizenga - HSBC Board Member, Ursula Kekich and Barbara Weintraub. This year’s tournament promises to be even better with a new two-day format brought about by popular demand. Even though it is hoped the event will be dominated by eagles and birds, it will literally be raining cats and dogs when arriving golfers are greeted by lots of furry friends, many of whom will melt the hearts of potential adopters.

In addition to the hard working Chainwomen of this event, the Committee Members are dedicated to making it a success and they are: Joy Abbott, Meredith Bruder, Dianne Davant, Liz Ferayoni, Ann Gremillion, Madelaine Halmos, Elaine Heinrich, Celia Hogan, Paige Hyatt, Joanne Johnsen, Patty Kearns, Karin Kirschbaum, Carol McCavill, Kay McFall, Annette Penrod, Tricia Rutsis, Caryl Sorensen, Katie Stephensen and Marlin Stull.

For the second year, the tournament is presented by M.A.B Paints and sponsored by Cundy Insurance, AutoNation Inc, the Miami Dolphins, American Airlines, Barbara & Michael Weintraub, E-Z-Go South Florida, Merrill Lynch, Dianne Davant Interiors, Katz, Barron, Squitero and Faust, P.A.

The $650 per-player entry fee will support the Humane Society of Broward County’s many programs and services including: providing services for more than 20,000 animals each year, educating the community about respect for animals through partnerships with the Boys and Girls Clubs, the Girl Scouts of Broward County and...
<table>
<thead>
<tr>
<th>Feature</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capitalized</td>
<td>XXXxxx</td>
<td>Character n-gram classifier says string is a person name (80% accurate)</td>
</tr>
<tr>
<td>Mixed Caps</td>
<td>XXXxxxx</td>
<td></td>
</tr>
<tr>
<td>All Caps</td>
<td>XXXXXX</td>
<td></td>
</tr>
<tr>
<td>Initial Cap</td>
<td>X....</td>
<td></td>
</tr>
<tr>
<td>Contains Digit</td>
<td>xxx5</td>
<td></td>
</tr>
<tr>
<td>All lowercase</td>
<td>xxxxx</td>
<td></td>
</tr>
<tr>
<td>Initial</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Punctuation</td>
<td>.,:;!(),</td>
<td></td>
</tr>
<tr>
<td>Period</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>Comma</td>
<td>,</td>
<td></td>
</tr>
<tr>
<td>Apostrophe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dash</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Preceded by HTML tag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hand-built FSM person-name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extractor says yes,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(prec/recall ~ 30/95)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conjunctions of all previous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feature pairs, evaluated at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the current time step.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conjunctions of all previous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feature pairs, evaluated at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>current step and one step</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ahead.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All previous features,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>evaluated two steps ahead.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All previous features,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>evaluated one step behind.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total number of features = ~500k**
Training and Testing

- Trained on 65k words from 85 pages, 30 different companies' web sites.
- Training takes 4 hours on a 1 GHz Pentium.
- Training precision/recall is 96% / 96%.
- Tested on different set of web pages with similar size characteristics.
- Testing precision is 92 – 95%, recall is 89 – 91%.
Part-of-speech Tagging

45 tags, 1M words training data, Penn Treebank

The asbestos fiber, crocidolite, is unusually resilient once it enters the lungs, with even brief exposures to it causing symptoms that show up decades later, researchers said.

Using spelling features*

<table>
<thead>
<tr>
<th></th>
<th>Error</th>
<th>ooV error</th>
<th>error</th>
<th>Δ err</th>
<th>ooV error</th>
<th>Δ err</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMM</strong></td>
<td>5.69%</td>
<td>45.99%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CRF</strong></td>
<td>5.55%</td>
<td>48.05%</td>
<td>4.27%</td>
<td>-24%</td>
<td>23.76%</td>
<td>-50%</td>
</tr>
</tbody>
</table>

* use words, plus overlapping features: capitalized, begins with #, contains hyphen, ends in -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies.
Cash receipts from marketings of milk during 1995 at $19.9 billion dollars, was slightly below 1994. Producer returns averaged $12.93 per hundredweight, $0.19 per hundredweight below 1994. Marketings totaled 154 billion pounds, 1 percent above 1994. Marketings include whole milk sold to plants and dealers as well as milk sold directly to consumers.

An estimated 1.56 billion pounds of milk were used on farms where produced, 8 percent less than 1994. Calves were fed 78 percent of this milk with the remainder consumed in producer households.

<table>
<thead>
<tr>
<th>Year</th>
<th>Number of Milk Cows 1/</th>
<th>Production of Milk and Milkfat 2/</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Milk of Fat in All</td>
<td>Total Milk Produced: Milkfat</td>
</tr>
<tr>
<td>1993</td>
<td>9,589</td>
<td>150,582 5,514.4</td>
</tr>
<tr>
<td>1994</td>
<td>9,500</td>
<td>153,664 5,623.7</td>
</tr>
<tr>
<td>1995</td>
<td>9,461</td>
<td>155,644 5,694.3</td>
</tr>
</tbody>
</table>

1/ Average number during year, excluding heifers not yet fresh.
2/ Excludes milk sucked by calves.
of milk during 1995 at $19.9 billion dollars, was
turns averaged $12.93 per hundredweight,
1994. Marketings totaled 154 billion pounds,
ings include whole milk sold to plants and dealers
consumers.

Is of milk were used on farms where produced,
s were fed 78 percent of this milk with the
er households.

Production of Milk and Milkfat: 1993-95
----------------------------------------

<table>
<thead>
<tr>
<th align="right">Milk Produced : Milk : Milkfat</th>
</tr>
</thead>
<tbody>
<tr>
<td align="right">---------------------------:</td>
</tr>
<tr>
<td align="right">--: of Fat in All :----------:</td>
</tr>
</tbody>
</table>

Production of Milk and Milkfat 2/  
----------------------------------------

<table>
<thead>
<tr>
<th align="right">Ratio : Percentage : Total</th>
</tr>
</thead>
<tbody>
<tr>
<td align="right">----: of Fat in All :----------:</td>
</tr>
</tbody>
</table>

Labels:
- Non-Table
- Table Title
- Table Header
- Table Data Row
- Table Section Data Row
- Table Footnote
- … (12 in all)

Features:
- Percentage of digit chars
- Percentage of alpha chars
- Indented
- Contains 5+ consecutive spaces
- Whitespace in this line aligns with prev.
- …
- Conjunctions of all previous features,
time offset: {0,0}, {-1,0}, {0,1}, {1,2}.
# Table Extraction

## Experimental Results

[Refs: Pinto, McCallum, Wei, Croft, 2003]

<table>
<thead>
<tr>
<th>Method</th>
<th>Line labels, percent correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>65 %</td>
</tr>
<tr>
<td>Stateless MaxEnt</td>
<td>85 %</td>
</tr>
<tr>
<td>CRF w/ out conjunctions</td>
<td>52 %</td>
</tr>
<tr>
<td>CRF</td>
<td>95 %</td>
</tr>
</tbody>
</table>

$\Delta$ error = 85%
South African provincial side Boland said on Thursday they had signed Leicestershire fast bowler David Millns on a one year contract. Millns, who toured Australia with England A in 1992, replaces former England all-rounder Phillip DeFreitas as Boland's overseas professional.
## Automatically Induced Features

[McCallum 2003]

<table>
<thead>
<tr>
<th>Index</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>inside-noun-phrase ($o_{t-1}$)</td>
</tr>
<tr>
<td>5</td>
<td>stopword ($o_t$)</td>
</tr>
<tr>
<td>20</td>
<td>capitalized ($o_{t+1}$)</td>
</tr>
<tr>
<td>75</td>
<td>word=the ($o_t$)</td>
</tr>
<tr>
<td>100</td>
<td>in-person-lexicon ($o_{t-1}$)</td>
</tr>
<tr>
<td>200</td>
<td>word=in ($o_{t+2}$)</td>
</tr>
<tr>
<td>500</td>
<td>word=Republic ($o_{t+1}$)</td>
</tr>
<tr>
<td>711</td>
<td>word=RBI ($o_t$) &amp; header=BASEBALL</td>
</tr>
<tr>
<td>1027</td>
<td>header=CRICKET ($o_t$) &amp; in-English-county-lexicon ($o_t$)</td>
</tr>
<tr>
<td>1298</td>
<td>company-suffix-word (firstmention$_{t+2}$)</td>
</tr>
<tr>
<td>4040</td>
<td>location ($o_t$) &amp; POS=NNP ($o_t$) &amp; capitalized ($o_t$) &amp; stopword ($o_{t-1}$)</td>
</tr>
<tr>
<td>4945</td>
<td>moderately-rare-first-name ($o_{t-1}$) &amp; very-common-last-name ($o_t$)</td>
</tr>
<tr>
<td>4474</td>
<td>word=the ($o_{t-2}$) &amp; word=of ($o_t$)</td>
</tr>
</tbody>
</table>
## Named Entity Extraction Results

[McCallum & Li, 2003]

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th># parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBN's Identifinder, word features</td>
<td>79%</td>
<td>~500k</td>
</tr>
<tr>
<td>CRFs word features, w/out Feature Induction</td>
<td>80%</td>
<td>~500k</td>
</tr>
<tr>
<td>CRFs many features, w/out Feature Induction</td>
<td>75%</td>
<td>~3 million</td>
</tr>
<tr>
<td>CRFs many candidate features with Feature Induction</td>
<td>90%</td>
<td>~60k</td>
</tr>
</tbody>
</table>
So far ...

• ... only looked at linear-chain CRFs

\[
p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right)
\]
General CRFs vs. HMMs

• More general and expressive modeling technique

• Comparable computational efficiency

• Features may be arbitrary functions of any or all observations

• Parameters need not fully specify generation of observations; require less training data

• Easy to incorporate domain knowledge

• State means only “state of process”, vs “state of process” and “observational history I’m keeping”
General CRFs

• Definition
  - Let \( G \) be a factor graph. Then \( p(y|x) \) is a CRF if for any \( x \), \( p(y|x) \) factorizes according to \( G \).

\[
p(y|x) = \frac{1}{Z(x)} \prod_{\Psi_A \in G} \exp \left( \sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(y_A, x_A) \right)
\]

But often some parameters tied: *Clique Templates*

\[
p(y|x) = \frac{1}{Z(x)} \exp \left( \sum_{k=1} \lambda_k f_k(y_t, y_{t-1}, x_t) \right)
\]
Questions

• **#1 – Inference**
  
  Again, learning requires computing $P(y_c|x)$ for given observations $x_1 \ldots x_N$ and CRF $\theta$.

• **#2 – Inference**

  Given observations $x_1 \ldots x_N$ and CRF $\theta$, what is the most likely labeling $y^* = \arg \max_y p(y|x)$?

• **#3 – Learning**

  Given iid training data $D=\{x^{(i)}, y^{(i)}\}, i=1\ldots N$, how do we estimate the parameters $\theta=\{ \lambda_k \}$ of a CRF?
Inference

• For graphs with small treewidth
  - Junction Tree Algorithm

• Otherwise approximate inference
  - Sampling-based approaches:
    - MCMC, ...
    • Not useful for training (too slow for every iteration)
  - Variational approaches:
    - Belief Propagation, ...
    • Popular
Learning

- Similar to linear-chain case
- Substitute model into likelihood ...

\[ l(\theta) = \sum_{C_p \in \mathcal{C}} \sum_{\Psi_c \in C_p} \sum_{k=1}^{K(p)} \lambda_{pk} f_{pk}(x, y_c) - \log Z(x) \]

... and compute partial derivatives, ...

\[ \frac{\partial l}{\partial \lambda_{pk}} = \sum_{\Psi_c \in C_p} f_{pk}(x, y_c) - \sum_{\Psi_c \in C_p} \sum_{y_c'} f_{pk}(x, y_c') p(y_c' | x) \]

and run nonlinear optimization (L-BFGS)
Markov Logic

- A general language capturing logic and uncertainty
- A Markov Logic Network (MLN) is a set of pairs $(F, w)$ where
  - $F$ is a formula in first-order logic
  - $w$ is a real number
- Together with constants, it defines a Markov network with
  - One node for each ground predicate
  - One feature for each ground formula $F$, with the corresponding weight $w$

$$P(x) = \frac{1}{Z} \exp \left( \sum_i w_i f_i(x) \right)$$
Example of an MLN

1.5 $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$
1.1 $\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

Suppose we have two constants: **Anna** (A) and **Bob** (B)
Example of an MLN

Suppose we have two constants: Anna (A) and Bob (B)

Friends(A,B)

Friends(A,A)  Smokes(A)  Smokes(B)  Friends(B,B)

Cancer(A)  Friends(B,A)  Cancer(B)

∀x Smokes(x) ⇒ Cancer(x)
∀x, y Friends(x, y) ⇒ (Smokes(x) ⇔ Smokes(y))
Example of an MLN

1.5  $\forall x \ Smokes(x) \Rightarrow Cancer(x)$
1.1  $\forall x, y \ Friends(x, y) \Rightarrow (\text{Smokes}(x) \leftrightarrow \text{Smokes}(y))$

Suppose we have two constants: Anna (A) and Bob (B)
Example of an MLN

\[ \forall x \text{ Smokes}(x) \implies \text{Cancer}(x) \]

\[ \forall x, y \text{ Friends}(x, y) \implies (\text{Smokes}(x) \iff \text{Smokes}(y)) \]

Suppose we have two constants: Anna (A) and Bob (B)
Joint Inference in Information Extraction

Hoifung Poon
University of Washington

(Joint work with Pedro Domingos)
Problems of Pipeline Inference

- **AI systems typically use pipeline architecture**
  - Inference is carried out in stages
  - E.g., information extraction, natural language processing, speech recognition, vision, robotics

- **Easy to assemble & low computational cost, but ...**
  - Errors accumulate along the pipeline
  - No feedback from later stages to earlier ones

- **Worse: Often process one object at a time**
We Need Joint Inference
