Lecture 10:
Conditional Random Fields Revisited,
Overview of General Structured Prediction
Today

• Naïve Bayes vs Logistic Regression
  – Detailed Comparison
  – Generalizes Conceptually to HMMs vs CRFs

• Conditional Random Fields Revisited
  – Using Logistic Regression Notation

• Overview of General Structured Prediction
Recall: Naïve Bayes

• Posits a generating model:
  – Single y
  – Multiple x features
  – Only keep track of:
    • $P(y), P(x^d | y)$

\[
P(x, y) = P(x | y)P(y) = P(y) \prod_{d} P(x^d | y)
\]

Each $x^d$ is conditionally independent given y. “Naïve” independence assumption!
Recall: Logistic Regression

\[ P(y \mid x) = \frac{\exp\left\{ w_y^T x - b_y \right\}}{\sum_k \exp\left\{ w_k^T x - b_k \right\}} = \frac{\exp\left\{ F(x, y) \right\}}{\sum_k \exp\left\{ F(x, k) \right\}} \]

\[ x \in \mathbb{R}^D \quad \text{and} \quad y \in \{1, 2, \ldots, L\} \]

- “Log-Linear” assumption
  - Linear scoring function (in exponent)
  - Most common discriminative probabilistic model
Naïve Bayes vs Logistic Regression

- NB has L parameters for $P(y)$ (i.e., $A$)
- LR has L parameters for bias $b$
- NB has $L \times D$ parameters for $P(x|y)$ (i.e., $O$)
- LR has $L \times D$ parameters for $w$
- **Same number of parameters!**

\[
P(x, y) = A_y \prod_{d=1}^{D} O_{x_d, y}^d
\]

**Naïve Bayes**

- $y \in \{0, 1\}^D$
- $x \in \{1, 2, ..., L\}$

\[
P(y | x) = \frac{e^{w_y^T x - b_y}}{\sum_k e^{w_k^T x - b_k}}
\]

**Logistic Regression**
Interpreting Parameters of LR

### Logistic Regression

\[ P(y \mid x) = \frac{e^{w_y^T x - b_y}}{\sum_k e^{w_k^T x - b_k}} \]

\[ \propto \exp\left\{ w_y^T x - b_y \right\} \]

\[ = \exp\left\{ -b_y \right\} \prod_d \exp\left\{ w_y^d x^d \right\} \]

\[ = \exp\left\{ A_y \right\} \prod_d \exp\left\{ O_{x^d, y}^d \right\} \]

### Naïve Bayes

\[ P(x, y) = A_y \prod_{d=1}^{D} O_{x^d, y}^d \]

- \( P(y) \)
- \( P(x^d \mid y) \)

**Exponent of LR looks similar to NB!**

**Cannot ignore denominator!!!**

**Rename Parameters**
Modeling $P(y | x)$

**Logistic Regression**

$$P(y | x) = \frac{\exp \left\{ w_y^T x - b_y \right\}}{\sum_k \exp \left\{ w_k^T x - b_k \right\}} = \frac{\exp \left\{ \sum_d O_{x^d,y} + A_y \right\}}{\sum_k \exp \left\{ \sum_d O_{x^d,k} + A_k \right\}}$$

**Naïve Bayes**

$$P(y | x) = \frac{P(x, y)}{P(x)} = \frac{P(x, y)}{\sum_k P(x, k)} = \frac{A_y \prod_{d=1}^{D} O_{x^d,y}}{\sum_k A_k \prod_{d=1}^{D} O_{x^d,k}}$$

There’s no need for each $A,O \leq 1$
Recall: Training Naïve Bayes

• Maximum Likelihood of Training Set:

\[
\arg\max P(S) = \arg\max \prod_i P(x_i, y_i) \quad \text{where} \quad S = \{(x_i, y_i)\}_{i=1}^N
\]

\[
= \arg\min \sum_i -\log P(x_i, y_i)
\]

– Subject to Naïve Bayes assumption on structure of \( P(x, y) \)

Only need to estimate \( P(y) \) and each \( P(x^d | y) \)!

\[
P(x, y) = P(x | y)P(y) = P(y)\prod_d P(x^d | y)
\]
Optimality Condition for Naïve Bayes

- **Define:**
  \[ P(x \mid y) = O_{x,y} = \frac{w_{x,y}}{\sum_{x'} w_{x',y}} \]

- **Supervised Training:**
  \[ \arg\min \sum_i \left[ -\log P(x_i \mid y_i) - \log P(y_i) \right] = \sum_i \left[ -\log w_{x_i,y_i} + \log \sum_{x'} w_{x',y_i} \right] \]

\[ \partial_{w_{x,y}} = -\frac{N_{x,y}}{w_{x,y}} + \frac{N_y}{\sum_{x'} w_{x',y}} \quad \Rightarrow \quad \frac{N_{x,y}}{N_y} = \frac{w_{x,y}}{\sum_{x'} w_{x',y}} \quad \Rightarrow \quad P(x \mid y) = \frac{N_{x,y}}{N_y} \]

Frequency counts in training set!

Just a re-parameterization

# training examples (x,y)
Recall: Training Logistic Regression

\[
\arg\min \sum_i -\log P(y_i \mid x_i) \equiv \sum_i \left[ -F(x_i, y_i) + \log \sum_{y'} \exp\{F(x_i, y')\} \right]
\]

\[
F(x, y) = w^T_y x - b_y = A_y + \sum_d O_{x,y}^d
\]

\[
P(y \mid x) = \frac{\exp\{F(x, y)\}}{\sum_{y'} \exp\{F(x, y')\}}
\]

Gradient (skipping derivation)

\[
\partial_{w_y} = \sum_i \left( -1_{[y_i=y]} + P(y \mid x_i) \right) \frac{\partial F(x_i, y)}{\partial_{w_y}} = - \sum_i \left( 1_{[y_i=y]} - P(y \mid x_i) \right) \frac{\partial F(x_i, y)}{\partial_{w_y}}
\]
Optimality Condition for Logistic Regression

Gradient (skipping derivation)

\[ \partial_{w_y} = \sum_i \left( -1_{[y_i=y]} + P(y \mid x_i) \right) \frac{\partial F(x_i, y)}{\partial w_y} = -\sum_i \left( 1_{[y_i=y]} - P(y \mid x_i) \right) \frac{\partial F(x_i, y)}{\partial w_y} \]

Setting gradient to 0:

\[ 0 = -\sum_i \left( 1_{[y_i=y]} - P(y \mid x_i) \right) \frac{\partial F(x_i, y)}{\partial w_y} \]

\[ \sum_i 1_{[y_i=y]} \frac{\partial F(x_i, y)}{\partial w_y} = \sum_i P(y \mid x_i) \frac{\partial F(x_i, y)}{\partial w_y} \]

Empirical frequency of y should match predicted frequency!
Comparison of Optimality Conditions

• Naïve Bayes:
  \[ P(x \mid y) = \frac{N_{x,y}}{N_y} \]
  \[ P(y) = \frac{N_y}{N} \]
  Correspond to exactly one model parameter!

• Logistic Regression:
  \[ \sum_i 1_{[y_i = y]} \frac{\partial F(x_i, y)}{\partial w_y} = \sum_i P(y \mid x_i) \frac{\partial F(x_i, y)}{\partial w_y} \]
  Does not correspond to exactly one model parameter!
Comparison of Optimality Conditions

• HMM:

\[ P(x | y) = \frac{N_{x,y}}{N_y} \]

\[ P(y' | y) = \frac{N_{y',y}}{N_y} \]

Correspond to exactly one model parameter!

• CRF:

\[ N_{y',y} \frac{\partial F(x_i, y)}{\partial w_{y,y'}} = \sum_i P(y', y | x_i) \frac{\partial F(x_i, y)}{\partial w_{y,y'}} \]

Does not correspond to exactly one model parameter!
<table>
<thead>
<tr>
<th>Generative</th>
<th>Discriminative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P(x,y)</strong></td>
<td>**P(y</td>
</tr>
<tr>
<td>• Joint model over x and y</td>
<td>• Conditional model</td>
</tr>
<tr>
<td>• Cares about everything</td>
<td>• Only cares about predicting well</td>
</tr>
<tr>
<td>Naïve Bayes, HMMs</td>
<td>Logistic Regression, CRFs</td>
</tr>
<tr>
<td>• Also Topic Models</td>
<td>• also SVM, Least Squares, etc.</td>
</tr>
<tr>
<td>Max Likelihood</td>
<td>Max (Conditional) Likelihood</td>
</tr>
<tr>
<td></td>
<td>• (=minimize log loss)</td>
</tr>
<tr>
<td></td>
<td>• Can pick any loss based on y</td>
</tr>
<tr>
<td></td>
<td>• Hinge Loss, Squared Loss, etc.</td>
</tr>
<tr>
<td>Always Probabilistic</td>
<td>Not Necessarily Probabilistic</td>
</tr>
<tr>
<td></td>
<td>• Certainly never joint over P(x,y)</td>
</tr>
<tr>
<td>Often strong assumptions</td>
<td>More flexible assumptions</td>
</tr>
<tr>
<td>• Keeps training tractable</td>
<td>• Focuses entire model on P(y</td>
</tr>
<tr>
<td>Mismatch between train &amp; predict</td>
<td>Train to optimize predict goal</td>
</tr>
<tr>
<td>• Requires Bayes’s rule</td>
<td></td>
</tr>
<tr>
<td>Can sample anything</td>
<td>Can only sample y given x</td>
</tr>
<tr>
<td>Can handle missing values in x</td>
<td>Cannot handle missing values in x</td>
</tr>
</tbody>
</table>
Recap: Sequence Prediction

- **Input:** \( x = (x^1, \ldots, x^M) \)
- **Predict:** \( y = (y^1, \ldots, y^M) \)
  - Each \( y^i \) one of \( L \) labels.

- \( x = \text{“Fish Sleep”} \)
- \( y = (\text{N}, \text{V}) \)

- \( x = \text{“The Dog Ate My Homework”} \)
- \( y = (\text{D}, \text{N}, \text{V}, \text{D}, \text{N}) \)

- \( x = \text{“The Fox Jumped Over The Fence”} \)
- \( y = (\text{D}, \text{N}, \text{V}, \text{P}, \text{D}, \text{N}) \)

**POS Tags:**
- \( \text{Det, Noun, Verb, Adj, Adv, Prep} \)
- \( L = 6 \)
"Log-Linear" 1st Order Sequential Model

\[ P(y \mid x) = \frac{1}{Z(x)} \exp \left\{ \sum_{j=1}^{M} \left( A_{y_j, y_{j-1}} + O_{y_j, x_j} \right) \right\} \]

\[ Z(x) = \sum_{y'} \exp \{ F(y', x) \} \quad \text{aka "Partition Function"} \]

\[ F(y, x) \equiv \sum_{j=1}^{M} \left( A_{y_j, y_{j-1}} + O_{y_j, x_j} \right) \]

\[ P(y \mid x) = \frac{\exp \{ F(y, x) \} }{Z(x)} \quad \text{log} P(y \mid x) = F(y, x) - \log(Z(x)) \]

\[ y^0 = \text{special start state, excluding end state} \]
\begin{itemize}
  \item $x = \text{“Fish Sleep”}$
  \item $y = (N,V)$
\end{itemize}

$$P(y \mid x) = \frac{1}{Z(x)} \exp \left\{ \sum_{j=1}^{M} (A_{y_j,y_{j-1}} + O_{y_j,x_j}) \right\}$$

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
 & $A_{N,*}$ & $A_{V,*}$ \\
\hline
$A_{*,N}$ & -2 & 1 \\
$A_{*,V}$ & 2 & -2 \\
$A_{*,\text{Start}}$ & 1 & -1 \\
\hline
\end{tabular}
\end{center}

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
 & $O_{N,*}$ & $O_{V,*}$ \\
\hline
$O_{*,\text{Fish}}$ & 2 & 1 \\
$O_{*,\text{Sleep}}$ & 1 & 0 \\
\hline
\end{tabular}
\end{center}

\[
P(N,V \mid \text{“Fish Sleep”}) = \frac{1}{Z(x)} \exp \left\{ A_{N,\text{Start}} + O_{N,\text{Fish}} + A_{V,N} + O_{V,\text{Sleep}} \right\} = \frac{1}{Z(x)} \exp \{ 4 \}
\]

\[
Z(x) = \text{Sum} \left( \begin{array}{c}
\exp(F(y,x)) \\
(N,N) \quad \exp(1+2-2+1) = \exp(2) \\
(N,V) \quad \exp(1+2+2+0) = \exp(4) \\
(V,N) \quad \exp(-1+1+2+1) = \exp(3) \\
(V,V) \quad \exp(-1+1-2+0) = \exp(-2)
\end{array} \right)
\]
- $x = \text{“Fish Sleep”}$
- $y = (N,V)$

$$P(N,V \mid \text{"Fish Sleep"}) = \frac{1}{Z(x)} \exp \left\{ A_{N,\text{Start}} + O_{N,\text{Fish}} + A_{V,N} + O_{V,\text{Sleep}} \right\}$$

$P(N,V \mid \text{"Fish Sleep"})$

*hold other parameters fixed

$$A_{N,\text{Start}} + O_{N,\text{Fish}} + A_{V,N} + O_{V,\text{Sleep}}$$
New Notation

Duplicate word features for each label.

\[
\phi_1^1 (\text{Noun} \mid \text{"Fish Sleep"}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = [1 = \text{Noun}] \land \left( x^j = \text{"Fish"} \right)
\]

\[
\phi_2^1 (\text{Noun} \mid \text{"Fish Sleep"}) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = [1 = \text{Noun}] \land \left( x^j = \text{"Sleep"} \right)
\]

\[
\phi_1^2 (\text{Verb} \mid \text{"Fish Sleep"}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = [1 = \text{Verb}] \land \left( x^j = \text{"Fish"} \right)
\]

\[
\phi_2^2 (\text{Verb} \mid \text{"Fish Sleep"}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = [1 = \text{Verb}] \land \left( x^j = \text{"Sleep"} \right)
\]
New Notation

One feature for every transition.

\[ \varphi_2(Noun, \text{Start}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \]

\[ \varphi_2(Verb, \text{Start}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \]

\[ \varphi_2(Verb, Noun) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \]

\[ \varphi_1(a \mid x) = \begin{bmatrix} 1 \left[ (a=Noun) \land x = 'Fish' \right] \\ 1 \left[ (a=Noun) \land x = 'Sleep' \right] \\ 1 \left[ (a=Verb) \land x = 'Fish' \right] \\ 1 \left[ (a=Verb) \land x = 'Sleep' \right] \end{bmatrix} \]

\[ \varphi_2(a, b) = \begin{bmatrix} 1 \left[ (a=Noun) \land (b=\text{Start}) \right] \\ 1 \left[ (a=Noun) \land (b=Noun) \right] \\ 1 \left[ (a=Verb) \land (b=\text{Start}) \right] \\ 1 \left[ (a=Verb) \land (b=Noun) \right] \\ 1 \left[ (a=Verb) \land (b=\text{Verb}) \right] \end{bmatrix} \]
New Notation

\[ F(y, x) \equiv \sum_{j=1}^{M} \left( A_{y^j, y^{j-1}} + O_{y^j, x^j} \right) \]

**Old Scoring Function**
- Scoring transitions
- Scoring input features

\[ F(y, x) \equiv \sum_{j=1}^{M} \left[ w^T \varphi^j(y^j, y^{j-1} | x) \right] \]

**New Scoring Function**
- Stacked Weight Vector
- Stacked Feature Vector

\[ w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \]

\[ \varphi^j(a, a' | x) = \begin{bmatrix} \varphi_1^j(a | x) \\ \varphi_2(a, a') \end{bmatrix} \]

\[ \varphi_1^j(a | x) = \begin{bmatrix} 1 \left[a=Noun\land x^j='Fish'\right] \\ 1 \left[a=Noun\land x^j='Sleep'\right] \\ 1 \left[a=Verb\land x^j='Fish'\right] \\ 1 \left[a=Verb\land x^j='Sleep'\right] \end{bmatrix} \]

\[ \varphi_2(a, a') = \begin{bmatrix} 1 \left[a=Noun\land a'='Start'\right] \\ 1 \left[a=Noun\land a'='Noun'\right] \\ 1 \left[a=Verb\land a'='Start'\right] \\ 1 \left[a=Verb\land a'='Noun'\right] \\ 1 \left[a=Verb\land a'='Verb'\right] \end{bmatrix} \]
\[ F(y, x) \equiv \sum_{j=1}^{M} \left[ w^T \varphi^j(y^j, y^{j-1} | x) \right] \]

\[ w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \]

\[ \varphi^j(a, a' | x) = \begin{bmatrix} \varphi^j_1(a | x) \\ \varphi^j_2(a, a') \end{bmatrix} \]

**Old Notation:**

<table>
<thead>
<tr>
<th>[ O_{N,*} ]</th>
<th>[ O_{V,*} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ O_{*,Fish} ]</td>
<td>2</td>
</tr>
<tr>
<td>[ O_{*,Sleep} ]</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ w_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \varphi^j_1(a | x) = \begin{bmatrix} 1 \left[ (a=Noun) \land (x^j='Fish') \right] \\ 1 \left[ (a=Noun) \land (x^j='Sleep') \right] \\ 1 \left[ (a=Verb) \land (x^j='Fish') \right] \\ 1 \left[ (a=Verb) \land (x^j='Sleep') \right] \end{bmatrix} \]

\[ w_2 = \begin{bmatrix} 1 \\ -2 \\ 2 \\ -1 \end{bmatrix} \quad \varphi^j_2(a, a') = \begin{bmatrix} 1 \left[ (a=Noun) \land (a'='Start') \right] \\ 1 \left[ (a=Noun) \land (a'='Noun') \right] \\ 1 \left[ (a=Verb) \land (a'='Verb') \right] \\ 1 \left[ (a=Verb) \land (a'='Start') \right] \\ 1 \left[ (a=Verb) \land (a'='Noun') \right] \\ 1 \left[ (a=Verb) \land (a'='Verb') \right] \end{bmatrix} \]
Why New Notation?

- Easier to reason about:
  - Computing Predictions
  - Computing Gradients
  - Extensions (just generalize $\phi$)

$$F(y, x) \equiv \sum_{j=1}^{M} w^T \phi^j (y^j, y^{j-1} \mid x)$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad \phi^j (a, b \mid x) = \begin{bmatrix} \phi_1^j (a \mid x) \\ \phi_2 (a, b) \end{bmatrix}$$

$$\phi_1^j (a \mid x) = \begin{bmatrix} 1 \left[(a=Noun) \land (x^j = 'Fish')\right] \\ 1 \left[(a=Noun) \land (x^j = 'Sleep')\right] \\ 1 \left[(a=Verb) \land (x^j = 'Fish')\right] \\ 1 \left[(a=Verb) \land (x^j = 'Sleep')\right] \end{bmatrix}$$

$$\phi_2 (a, b) = \begin{bmatrix} 1 \left[(a=Noun) \land (b=Start)\right] \\ 1 \left[(a=Noun) \land (b=Noun)\right] \\ 1 \left[(a=Noun) \land (b=Verb)\right] \\ 1 \left[(a=Verb) \land (b=Start)\right] \\ 1 \left[(a=Verb) \land (b=Noun)\right] \\ 1 \left[(a=Verb) \land (b=Verb)\right] \end{bmatrix}$$
Conditional Random Fields

\[ P(y \mid x) = \frac{1}{Z(x)} \exp \{ F(y, x) \} \]

\[ Z(x) = \sum_{y'} \exp \{ F(y', x) \} \]

\[ F(y, x) \equiv \sum_{j=1}^{M} \left[ w^T \varphi^j (y^j, y^{j-1} \mid x) \right] \]

\[ w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad \varphi^j (a, b \mid x) = \begin{bmatrix} \varphi_1^j (a \mid x) \\ \varphi_2 (a, b) \end{bmatrix} \]

\[ \varphi_1^j (a \mid x) = \begin{bmatrix} 1 \left[ (a=Noun) \land (x^j='Fish') \right] \\ 1 \left[ (a=Noun) \land (x^j='Sleep') \right] \\ 1 \left[ (a=Verb) \land (x^j='Fish') \right] \\ 1 \left[ (a=Verb) \land (x^j='Sleep') \right] \end{bmatrix} \]

\[ \varphi_2 (a, b) = \begin{bmatrix} 1 \left[ (a=Noun) \land (b=Start) \right] \\ 1 \left[ (a=Noun) \land (b=Noun) \right] \\ 1 \left[ (a=Verb) \land (b=Verb) \right] \\ 1 \left[ (a=Verb) \land (b=Start) \right] \\ 1 \left[ (a=Verb) \land (b=Noun) \right] \\ 1 \left[ (a=Verb) \land (b=Verb) \right] \end{bmatrix} \]
\[ P(N,V \mid x = "Fish Sleep") = \frac{1}{Z(x)} \exp \left\{ w_1^T \varphi_1(N,x) + w_2^T \varphi_2(N,Start) + w_1^T \varphi_1(V,x) + w_2^T \varphi_2(V,N) \right\} \]

\[ = \frac{1}{Z(x)} \exp \left\{ w_{1,1} + w_{2,1} + w_{1,4} + w_{2,5} \right\} = \frac{1}{Z(x)} \exp \{ 2 + 1 + 0 + 1 \} = \frac{1}{Z(x)} \exp \{ 4 \} \]

\[ Z(x) = \text{Sum}_{y} \exp(F(y,x)) \]

<table>
<thead>
<tr>
<th>y</th>
<th>exp(F(y,x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N,N)</td>
<td>exp(2+1+1-2) = exp(2)</td>
</tr>
<tr>
<td>(N,V)</td>
<td>exp(2+1+0+1) = exp(4)</td>
</tr>
<tr>
<td>(V,N)</td>
<td>exp(1-1+1+2) = exp(3)</td>
</tr>
<tr>
<td>(V,V)</td>
<td>exp(1-1+0-2) = exp(-2)</td>
</tr>
</tbody>
</table>
Summary of New Notation

• Generic Logistic Model Notation:

\[ P(y \mid x) = \frac{1}{Z(x)} \exp\{F(y, x)\} \]

\[ Z(x) = \sum_{y'} \exp\{F(y', x)\} \quad F(y, x) \equiv \sum_{j=1}^{M} [w^T \varphi^j(y^j, y^{j-1} \mid x)] \]

• Define feature function:
  – Linear model in feature representation
  – Applies to both CRFs and basic LR
Computing Predictions (Viterbi)

\[
\arg\max_y P(y \mid x) = \arg\max_y F(y, x) \quad F(y^1, x) = \sum_{j=1}^{k} [w^T \phi^j (y^j, y^{j-1} \mid x)]
\]

<table>
<thead>
<tr>
<th>Maintain length-k prefix solutions</th>
<th>( \hat{Y}^k(T) = \left( \arg\max_{y^{1:k-1}} F(y^{1:k-1} \oplus T, x) \right) \oplus T )</th>
</tr>
</thead>
</table>
| Recursively solve for length-(k+1) solutions | \( \hat{Y}^{k+1}(T) = \left( \arg\max_{y^{1:k} \in \{\hat{Y}^k(T)\}_T} F(y^{1:k} \oplus T, x) \right) \oplus T \\
\quad = \left( \arg\max_{y^{1:k} \in \{\hat{Y}^k(T)\}_T} F(y^{1:k}, x) + w^T \phi^{k+1}(T, y^k, x) \right) \oplus T \) |
| Predict via best length-M solution | \[ \arg\max_y F(y, x) = \arg\max_{y \in \{\hat{Y}^M(T)\}_T} F(y, x) \] |
Solve: \( \hat{Y}^2(V) = \left( \arg\max_{y^1} F(y^1, x) + w^T \varphi^2(V, y^1 | x) \right) \oplus V \)

Store each \( \hat{Y}^1(T) \) & \( F(\hat{Y}^1(T), x) \)

\( \hat{Y}^1(V) \) to \( \hat{Y}^2(V) \)

\( y^1 = V \)

\( \hat{Y}^1(D) \) to \( \hat{Y}^2(D) \)

\( y^1 = D \)

\( \hat{Y}^1(N) \) to \( \hat{Y}^2(N) \)

\( y^1 = N \)

\( \hat{Y}^1(T) \) is just T
Solve: \[
\hat{Y}^2(V) = \left( \arg\max_{y^1 \in \{\hat{Y}^1(T)\}_T} F(y^1, x) + w^T \varphi^2(V, y^1 | x) \right) \oplus V
\]

Store each \(\hat{Y}^1(T)\) & \(F(\hat{Y}^1(T), x)\)

\(\hat{Y}^1(V)\) \(\rightarrow\) \(\hat{Y}^2(V)\)

\(\hat{Y}^1(D)\) \(\rightarrow\) \(\hat{Y}^2(D)\)

\(\hat{Y}^1(N)\) \(\rightarrow\) \(\hat{Y}^2(N)\)

\(y^1 = N\)

\(\hat{Y}^1(T)\) is just T

Ex: \(\hat{Y}^2(V) = (N, V)\)
Solve: \[
\hat{Y}^3(V) = \left( \arg\max_{y^{1:2} \in \bar{\hat{Y}}^2(T)} F(y^{1:2}, x) + w^T \varphi_j(V, y^2 | x) \right) \oplus V
\]

Store each \(\hat{Y}^1(T) \& F(\hat{Y}^1(T), x)\)

Store each \(\hat{Y}^2(Z) \& F(\hat{Y}^2(Z), x)\)

\(\hat{Y}^1(V)\)

\(\hat{Y}^2(V)\)

\(\hat{Y}^3(V)\)

\(\hat{Y}^1(D)\)

\(\hat{Y}^2(D)\)

\(\hat{Y}^3(D)\)

\(\hat{Y}^1(N)\)

\(\hat{Y}^2(N)\)

\(\hat{Y}^3(N)\)

\(\hat{Y}^1(Z)\) is just Z

Ex: \(\hat{Y}^2(V) = (N, V)\)
Solve: $\hat{Y}^M(V) = \left\{ \begin{array}{ll} \arg\max_{y^{1:M-1} \in \{\hat{Y}^{M-1}(T)\}} & F(y^{1:M-1}, x) + w^T \varphi^M(V, y^{M-1} | x) \\ \oplus V \end{array} \right.$

Store each $\hat{Y}^1(Z)$ & $F(\hat{Y}^1(Z), x)$

$\hat{Y}^1(V)$

Store each $\hat{Y}^2(T)$ & $F(\hat{Y}^2(T), x)$

$\hat{Y}^2(V)$

Store each $\hat{Y}^3(T)$ & $F(\hat{Y}^3(T), x)$

$\hat{Y}^3(V)$

$\hat{Y}^1(D)$

$\hat{Y}^2(D)$

$\hat{Y}^3(D)$

$\hat{Y}^1(N)$

$\hat{Y}^2(N)$

$\hat{Y}^3(N)$

\(\hat{Y}^1(T)\) is just T

Ex: $\hat{Y}^2(V) = (N, V)$

Ex: $\hat{Y}^3(V) = (D, N, V)$
Solve: \[
\hat{Y}^M(V) = \arg\max_{y^{1:M-1} \in \{\hat{Y}^{M-1}(T)\}_T} F(y^{1:M-1}, x) + w^T \phi^M(V, y^{M-1} | x) \] \oplus V

Store each
\(\hat{Y}^1(Z) \& F(\hat{Y}^1(Z), x)\)

Store each
\(\hat{Y}^2(T) \& F(\hat{Y}^2(T), x)\)

Store each
\(\hat{Y}^3(T) \& F(\hat{Y}^3(T), x)\)

Decomposes additively by pairwise feature vector:
\(\phi^j(a,b | x)\)

Easier to keep track of!

\(\hat{Y}^1(T)\) is just \(T\)

Ex: \(\hat{Y}^2(V) = (N, V)\)

Ex: \(\hat{Y}^3(V) = (D,N,V)\)
Computing Conditional Probabilities

\[
P(y \mid x) = \frac{1}{Z(x)} \exp \{F(y, x)\} = \frac{1}{Z(x)} \exp \left\{ \sum_{j=1}^{M} w^T \varphi^j(y^j, y^{j-1} \mid x) \right\}
\]

\[
Z(x) = \sum_{y'} \exp \{F(y', x)\}
\]

Matrix Notation:

\[
G^j(b, a) = \exp \{w^T \varphi^j(b, a \mid x)\}
\]

Challenges:
- Compute \(Z(x)\) efficiently
- Numerical instability

\[
P(y \mid x) = \frac{1}{Z(x)} \prod_{j=1}^{M} G^j(y^j, y^{j-1})
\]

\[
Z(x) = \sum_{y'} \prod_{j=1}^{M} G^j(y'^j, y'^{j-1})
\]

See course notes.
Matrix Semiring

$$Z(x) = \sum_{y'} \prod_{j=1}^{M} G^j (y'^j, y'^{j-1})$$

$$G^j (b, a) = \exp \{ w^T \varphi^j (b, a \mid x) \}$$

$$G^{1:2} (b, a) = \sum_c G^2 (b, c) G^1 (c, a)$$

$$G^{i:j} (b, a) = G^i \equiv G^j G^{j-1} \ldots G^i$$

See course notes.
## Computing Partition Function

<table>
<thead>
<tr>
<th>General M</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Do M matrix computations to compute $G^{1:M}$</td>
</tr>
<tr>
<td>- $Z(x) = \text{sum column 'Start' of } G^{1:M}$</td>
</tr>
</tbody>
</table>

### Consider Length-1 ($M=1$)

$$Z(x) = \sum_a G^1(a, \text{Start})$$

Sum column ‘Start’ of $G^1$!

### M=2

$$Z(x) = \sum_{a,b} G^2(b, a) G^1(a, \text{Start}) = \sum_b G^{1:2}(b, \text{Start})$$

Sum column ‘Start’ of $G^{1:2}$!

### General M

- Do M matrix computations to compute $G^{1:M}$
- $Z(x) = \text{sum column 'Start' of } G^{1:M}$

See course notes for more efficient approach.
Dealing w/ Numerical Instability

• Previous slide suffers from numerical instability
  – \( G^{1:k} \) can easily overflow and/or underflow

Numerical Stability via Scaling:

\[
\hat{G}^{1:j} = \frac{1}{C^j} \left( G^j \hat{G}^{1:j-1} \right) \\
G^{1:M} = \hat{G}^{1:M} \prod_{j=1}^{M} C^j
\]

Example Scaling Factor:

\[
C^j = \sum_{a,b} \left[ G^j \hat{G}^{1:j-1} \right]_{ba}
\]

\[
\log(Z(x)) = \log \left( \sum_a G^{1:M}_{a,\text{Start}} \right) = \log \left( \sum_a \hat{G}^{1:M}_{a,\text{Start}} \right) + \sum_{j=1}^{M} \log(C^j)
\]

\[
\log(P(y \mid x)) = F(y, x) - \log(Z(x))
\]

Use log probs instead!

See course notes.
Training
(Stochastic) Gradient Descent

• Minimize log loss of training data:

\[
\arg\min_w \sum_{i=1}^{N} -\log P(y_i \mid x_i) = \arg\min_w \sum_{i=1}^{N} -F(y_i, x_i) + \log (Z(x_i))
\]

\[
\partial_w - F(y, x) = -\sum_{j=1}^{M} \varphi_j^j (y^j, y^{j-1} \mid x)
\]

\[
\partial_w \log(Z(x)) = \sum_{j=1}^{M} \sum_{a,b} P(y^j = b, y^{j-1} = a \mid x) \varphi_j^j (b, a \mid x)
\]

See course notes.
Optimality Condition

\[
\arg\min_{\Theta} \sum_{i=1}^{N} -\log P(y_i | x_i) = \arg\min_{\Theta} \sum_{i=1}^{N} -F(y_i, x_i) + \log (Z(x_i))
\]

- Optimality condition:

\[
\sum_{i=1}^{N} \sum_{j=1}^{M_i} \varphi^j(y_i^j, y_i^{j-1} | x_i) = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \sum_{a,b} P(y^j = b, y^{j-1} = a | x_i) \varphi^j(b, a | x_i)
\]

- Frequency counts = Cond. expectation on training data!
  - If each feature is disjoint, then above equality holds for each (a,b):

\[
\sum_{i=1}^{N} \sum_{j=1}^{M_i} 1_{[(y_i^j = b) \land (y_i^{j-1} = a)]} \varphi^j(b, a | x_i) = \sum_{i=1}^{N} \sum_{j=1}^{M_i} P(y^j = b, y^{j-1} = a | x_i) \varphi^j(b, a | x_i)
\]

\[
S = \{(x_i, y_i)\}_{i=1}^{N}
\]

See course notes.
Computing $P(y^j=b, y^{j-1}=a \mid x)$

(Forward-Backward)

$$\partial_w \log(Z(x)) = \sum_{j=1}^{M} \sum_{a,b} P(y^j = b, y^{j-1} = a \mid x) \varphi^j(b, a \mid x)$$

$$P(y^j = b, y^{j-1} = a \mid x) = \sum_{y^{1:j-2}} \sum_{y^{j+1:M}} P(y^{1:j-2} \oplus (a, b) \oplus y^{j+1:M} \mid x)$$

Forward Computes 1st Sum Efficiently

Backward Computes 2nd Sum Efficiently

See course notes.
Forward-Backward for CRFs

<table>
<thead>
<tr>
<th>$\alpha^1(a) = G^1(a, \text{Start})$</th>
<th>$\beta^M(b) = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha^j(a) = \sum_{a'} \alpha^{j-1}(a')G^j(a, a')$</td>
<td>$\beta^j(b) = \sum_{b'} \beta^{j+1}(b')G^{j+1}(b', b)$</td>
</tr>
</tbody>
</table>

$$P(y^j = b, y^{j-1} = a \mid x) = \frac{\alpha^{j-1}(a)G^j(b, a)\beta^j(b)}{Z(x)}$$

$$Z(x) = \sum_{y'} \exp\{F(y', x)\} \quad G^j(b, a) = \exp\{w^T \varphi^j(b, a \mid x)\}$$

See course notes.
Dealing w/ Numerical Instability

• Numerical instability: $\alpha^j$ & $\beta^j$ vectors can blow up

• Observation:

$$P(y^j = b, y^{j-1} = a \mid x) = \frac{\alpha^{j-1}(a)G^j(b,a)\beta^j(b)}{Z(x)} = \frac{\alpha^{j-1}(a)G^j(b,a)\beta^j(b)}{\sum_{a',b'}\alpha^{j-1}(a')G^j(b',a')\beta^j(b')}$$

• New $\alpha^j$ & $\beta^j$ vectors:

$$\hat{\alpha}^j(a) = \frac{1}{C^j_\alpha} \sum_b \hat{\alpha}^{j-1}(b)G^j(a,b) \quad \hat{\beta}^j(b) = \frac{1}{C^j_\beta} \sum_a \hat{\beta}^{j+1}(a)G^j(a,b)$$

$$C^j_\alpha = \sum_{a,b} \hat{\alpha}^{j-1}(b)G^j(a,b) \quad C^j_\beta = \sum_{a,b} \hat{\beta}^{j+1}(a)G^j(a,b)$$

See course notes.
Recap: Conditional Random Fields

• “Log-Linear” 1\textsuperscript{st} order sequence models
  – Can compute conditional probabilities $P(y|x)$

• Pairwise feature maps $\phi_j(b,a|x)$
  – Arbitrary features that depend on pairs of labels.

• Train via minimizing neg log likelihood

• Dynamic programming to train and predict
General Structured Prediction
More Elaborate Scoring Functions

- Structure encoded by linear scoring function:
  \[ F(y, x) \]

- 2\textsuperscript{nd} Order Sequential Model:
  \[ F(y, x) \equiv \sum_{j=1}^{M} [w^T \varphi^j(y^j, y^{j-1}, y^{j-2} | x)] \]

- Classification Model:
  \[ F(y, x) \equiv w^T \varphi(y | x) \]

- Efficient Prediction:
  \[ \arg\max_y F(y, x) \]
More Elaborate Scoring Functions

• Structure encoded by linear scoring function:

\[ F(y, x) \]

Remainder of Lecture:
Tour of Structured Prediction Models
Some Might be Interesting to You...

• Efficient Prediction:

\[ \arg\max_y F(y, x) \]
Graphical Models

\[ \varphi^j(a, b \mid x) = \begin{bmatrix} \varphi_1(a \mid x^j) \\ \varphi_2(a, b) \end{bmatrix} \]

Graph structure encodes structural dependencies between \( y^i \)!

https://piazza.com/cornell/fall2013/btry6790cs6782/resources
http://www.cs.cmu.edu/~guestrin/Class/10708/
https://www.coursera.org/course/pgm
Graphical Models

\[ \varphi^j(a, b \mid x) = \begin{bmatrix} \varphi_1^j(a \mid x) \\ \varphi_2(a, b) \end{bmatrix} \]

Graph structure encodes structural dependencies between \( y^j \)!

https://piazza.com/cornell/fall2013/btry6790cs6782/resources
http://www.cs.cmu.edu/~guestrin/Class/10708/
https://www.coursera.org/course/pgm
Graphical Models

\[ \varphi^i(a, b, c \mid x) = \begin{bmatrix} \varphi_1(a \mid x^i) \\ \varphi_3(a, b, c) \end{bmatrix} \]

Graph structure encodes structural dependencies between \( y^i \)!

Features depend on cliques in graphical model representation.

https://piazza.com/cornell/fall2013/btry6790cs6782/resources
http://www.cs.cmu.edu/~guestrin/Class/10708/
https://www.coursera.org/course/pgm
Tree Structured Models

\[
F(y, x) \equiv \sum_{j=1}^{M} \left[ w_1^T \varphi_1(y^j \mid x^j) + \sum_{i \in C_j} w_2^T \varphi_2(y^j, y^i) \right]
\]
Prediction via Dynamic Programming

\[ F(y, x) = \sum_{j=1}^{M} \left[ w_1^T \varphi_1(y^j \mid x^j) + \sum_{i \in C_j} w_2^T \varphi_2(y^j, y^i) \right] \]

1. Solve partial solutions of Leaves
2. Solve partial sol. of next level up
3. Repeat Step 2 until Root
4. Pick best partial solution of Root

*Max-Product Algorithm for Tree Graphical Models
*Viterbi = Max-Product for Linear Chain Graphical Models
Loopy Graphical Models

Stereo (binocular) Depth Detection

- Each $y^{ij}$ is depth of pixel
- Neighbor pixels are similar
- Features over pairs of pixels
- "Loopy" Graphical Model
- Prediction is NP-Hard!

$$\text{argmax } F(y, x)$$

http://vision.middlebury.edu/MRF/
http://www.seas.upenn.edu/~taskar/pubs/mmamn.pdf
String Alignment

**Predict Folding Structure & Function of Protein**

**Database D of Known Proteins**
(very well studied)

**Larger Database G of Homologies**
(proteins w/ known similarities to D)

Train on G: learn how to align any amino acid seq to proteins in D

\[
F(y, x) \text{ encodes score of different types of substitutions, insertions & deletions}
\]

\[
x = \text{pair of strings (one from D)}
\]
\[
y = \text{alignment}
\]

x = pair of strings (one from D)
y = alignment

\[
F(y, x) \text{ encodes score of different types of substitutions, insertions & deletions}
\]

See Also: http://journals.plos.org/ploscompbio/article?id=10.1371/journal.pcbi.1000173
Ranking

Find \( w \) that predicts best ranking of search results.

Every relevant result should be above every non-relevant result.

\[
y_{ij} \in \{-1, +1\}
\]

\[
F(y, x) = \sum_{i,j} y_{ij} \left[ w^T \varphi(x^i) - w^T \varphi(x^j) \right]
\]

\[
\text{argmax } F(y, x) = \text{sort} \left\{ w^T \varphi(x^j) \right\}_j
\]

\( x = \text{query \& set of results} \)

\( y = \text{ranking} \)

http://www.cs.cornell.edu/People/tj/publications/joachims_05a.pdf


Summary: Structured Prediction

• Very general setting
  – Applicable to prediction made jointly over multiple \( y \)'s
  – Prediction in Graphical Models

• Many learning algorithms for structured prediction
  – CRFs, SSVMs, Structured Perceptron, Learning Reductions

• Topic for Entire Class!

  http://www.nowozin.net/sebastian/cvpr2011tutorial/
  http://www.cs.cmu.edu/~nasmith/sp4nlp/
  http://www.cs.cornell.edu/Courses/cs778/2006fa/
  https://www.sites.google.com/site/spflodd/
  http://www.cs.cornell.edu/People/tj/publications/joachims_06b.pdf
Next Week

• Lecture Tuesday:
  – Learning Reductions
  – Recent Applications

• NO Lecture Thursday:
  – Student-Faculty Conference

• Recitation Thursday:
  – Review of Conditional Random Fields