Caltech

Machine Learning & Data Mining

CS/CNS/EE 155

Lecture 10:

Conditional Random Fields Revisited,

Overview of General Structured Prediction

Today

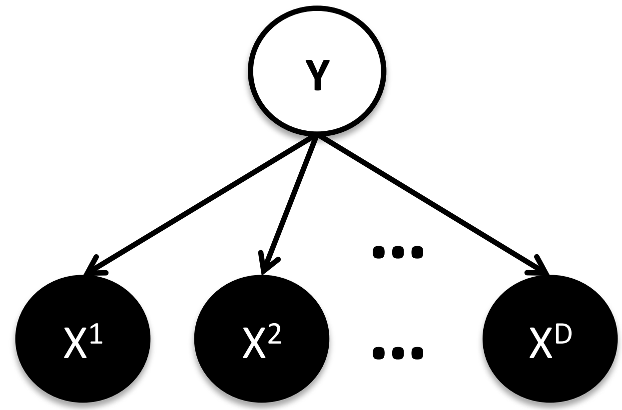
- Naïve Bayes vs Logistic Regression
  - Detailed Comparison
  - Generalizes Conceptually to HMMs vs CRFs

- 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), $P(x^d|y)$**



Graphical Model Diagram

$$P(x,y) = P(x \mid y)P(y) = P(y)\prod_d P(x^d \mid y)$$

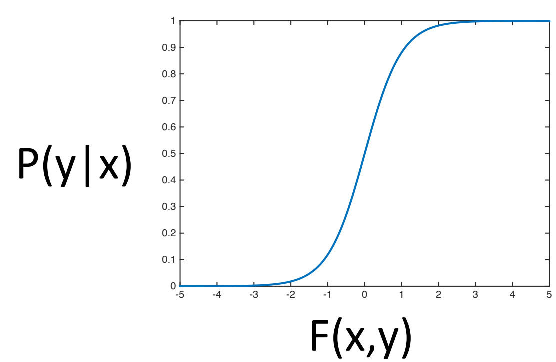
Each x^d is conditionally independent given y.
"Naïve" independence assumption!

Recall: Logistic Regression

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

$$x \in R^D$$
$$y \in \{1,2,\ldots,L\}$$

- "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*D parameters for P(x|y) (i.e, O)
- LR has L*D parameters for w

- **Same number of parameters!**

Naïve Bayes

$$P(x,y) = A_y \prod_{d=1}^{D} O^{d}_{x^d,y}$$

P(y) — A_y; P(x|y) — $O^{d}_{x^d,y}$

Logistic Regression

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

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

Interpreting Parameters of LR

Logistic Regression

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

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

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

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

Rename Parameters

Naïve Bayes

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

P(y) $\quad$ P(x^d|y)

Exponent of LR looks similar to NB!

Cannot ignore denominator!!!

Modeling P(y|x)

Logistic Regression

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

Naïve Bayes

P(y) → A_y; P(x^d|y) → $O^d_{x^d,y}$

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

There's no need for each A,O ≤ 1

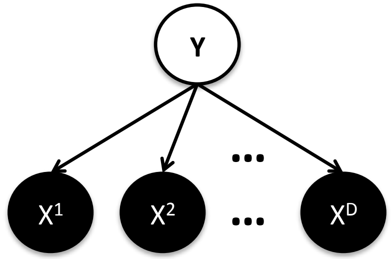
Recall: Training Naïve Bayes

- Maximum Likelihood of Training Set:

$$\operatorname{argmax} P(S) = \operatorname{argmax} \prod_i P(x_i, y_i) \qquad S = \{(x_i, y_i)\}_{i=1}^{N}$$

$$= \operatorname{argmin} \sum_i -\log P(x_i, y_i)$$

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



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

$$P(x,y) = P(x \mid y)P(y) = P(y)\prod_d P(x^d \mid y)$$

Optimality Condition for Naïve Bayes

- **Define:** $P(x \mid y) = O_{x,y} = \dfrac{w_{x,y}}{\sum_{x'} w_{x'}}$

Just a re-parameterization

- **Supervised Training:**

$$\operatorname{argmin} \sum_i \left[-\log P(x_i \mid y_i) - \log P(y_i) \right]$$

$$= \sum_i \left[-\log w_{x_i, y_i} + \log \sum_{x'} w_{x', y_i} \right]$$

training examples (x,y)

$$\partial_{w_{x,y}} = -\frac{N_{x,y}}{w_{x,y}} + \frac{N_y}{\sum_{x'} w_{x',y}} \rightarrow \frac{N_{x,y}}{N_y} = \frac{w_{x,y}}{\sum_{x'} w_{x',y}} \rightarrow P(x \mid y) = \frac{N_{x,y}}{N_y}$$

Frequency counts in training set!

Recall: Training Logistic Regression

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

$$F(x, y) = w_y^T x - b_y = A_y + \sum_d O_{x,y}^d$$

$$P(y \mid x) = \frac{\exp\left\{ F(x, y) \right\}}{\sum_{y'} \exp\left\{ F(x, y') \right\}}$$

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: $$P(x \mid y) = \frac{N_{x,y}}{N_y} \qquad P(y) = \frac{N_y}{N}$$

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: $$P(x \mid y) = \frac{N_{x,y}}{N_y} \qquad P(y \mid y') = \frac{N_{y',y}}{N_y}$$

Correspond to exactly one model parameter!

- CRF: $$N_{y',y} \frac{\partial F(x_i, y)}{\partial_{w_{y,y'}}} = \sum_i P(y', y \mid x_i) \frac{\partial F(x_i, y)}{\partial_{w_{y,y'}}}$$

Does **not** correspond to exactly one model parameter!

Generative	Discriminative
P(x,y) • Joint model over x and y • Cares about everything	P(y|x) (when probabilistic) • Conditional model • Only cares about predicting well
Naïve Bayes, HMMs • Also Topic Models	Logistic Regression, CRFs • also SVM, Least Squares, etc.
Max Likelihood	Max (Conditional) Likelihood • (=minimize log loss) • Can pick any loss based on y • Hinge Loss, Squared Loss, etc.
Always Probabilistic	Not Necessarily Probabilistic • Certainly never joint over P(x,y)
Often strong assumptions • Keeps training tractable	More flexible assumptions • Focuses entire model on P(y|x)
Mismatch between train & predict • Requires Bayes's rule	Train to optimize predict goal
Can sample anything	Can only sample y given x
Can handle missing values in x	Cannot handle missing values in x

Recap: Sequence Prediction

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

POS Tags:
Det, Noun, Verb, Adj, Adv, Prep

L=6

- x = "Fish Sleep"
- y = (N, V)

- x = "The Dog Ate My Homework"
- y = (D, N, V, D, N)

- x = "The Fox Jumped Over The Fence"
- y = (D, N, V, P, D, N)

"Log-Linear" 1st Order Sequential Model

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

$$Z(x) = \sum_{y'} \exp\left\{ F(y', x) \right\}$$

aka "Partition Function"

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

Scoring Function

Scoring transitions — Scoring input features

$$P(y \mid x) = \frac{\exp\left\{ F(y,x) \right\}}{Z(x)}$$

$$\log P(y \mid x) = F(y,x) - \log\left(Z(x) \right)$$

y^0 = special start state, excluding end state

- x = "Fish Sleep"
- y = (N,V)

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

$A_{N,V}$

	$A_{N,*}$	$A_{V,*}$
$A_{*,N}$	-2	1
$A_{*,V}$	2	-2
$A_{*,Start}$	1	-1

	$O_{N,*}$	$O_{V,*}$
$O_{*,Fish}$	2	1
$O_{*,Sleep}$	1	0

$w_{V,Fish}$

$$P(N,V\,|\,\text{"}Fish\ Sleep\text{"}) = \frac{1}{Z(x)} \exp\left\{ A_{N,Start} + O_{N,Fish} + A_{V,N} + O_{V,Sleep} \right\} = \frac{1}{Z(x)} \exp\{4\}$$

Z(x) = Sum (

y	exp(F(y,x))
(N,N)	exp(1+2-2+1) = exp(2)
(N,V)	exp(1+2+2+0) = exp(4)
(V,N)	exp(-1+1+2+1) = exp(3)
(V,V)	exp(-1+1-2+0) = exp(-2)

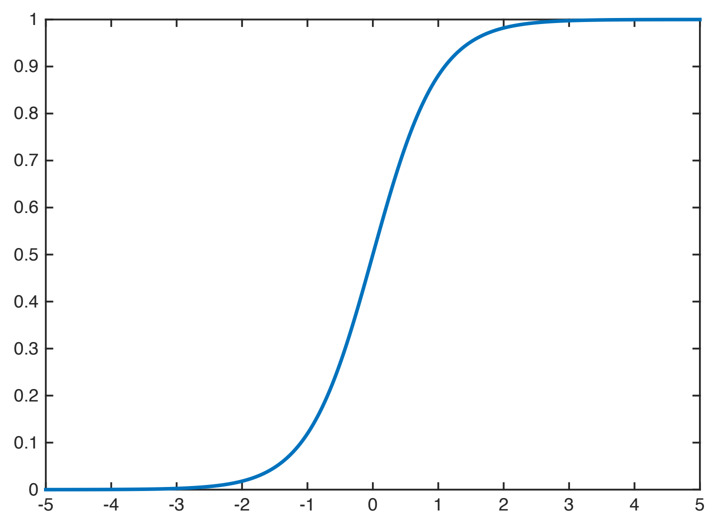
)

- x = “Fish Sleep”
- y = (N,V)

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

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

*hold other parameters fixed



$$A_{N,Start} + O_{N,Fish} + A_{V,N} + O_{V,Sleep}$$

New Notation

Duplicate word features for each label.

$$\varphi_1^j(a \mid x) = \begin{bmatrix} 1_{[(a=Noun)\wedge(x^j='Fish')]} \\ 1_{[(a=Noun)\wedge(x^j='Sleep')]} \\ 1_{[(a=Verb)\wedge(x^j='Fish')]} \\ 1_{[(a=Verb)\wedge(x^j='Sleep')]} \end{bmatrix}$$

Noun Class Features

$$\varphi_1^1(Noun \mid \text{"Fish Sleep"}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\varphi_1^2(Noun \mid \text{"Fish Sleep"}) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Verb Class Features

$$\varphi_1^1(Verb \mid \text{"Fish Sleep"}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\varphi_1^2(Verb \mid \text{"Fish Sleep"}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\varphi_1^j(a \mid x) = \begin{bmatrix} 1_{[a=1]}\phi_1(x^j) \\ \vdots \\ 1_{[a=L]}\phi_1(x^j) \end{bmatrix}$$

New Notation

One feature for every transition.

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

$$\varphi_2(Noun, Start) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\varphi_2(Verb, Start) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

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

$$\varphi_2(a,b) = \begin{bmatrix} 1_{\left[(a=Noun)\wedge(b=Start)\right]} \\ 1_{\left[(a=Noun)\wedge(b=Noun)\right]} \\ 1_{\left[(a=Noun)\wedge(b=Verb)\right]} \\ 1_{\left[(a=Verb)\wedge(b=Start)\right]} \\ 1_{\left[(a=Verb)\wedge(b=Noun)\right]} \\ 1_{\left[(a=Verb)\wedge(b=Verb)\right]} \end{bmatrix}$$

New Notation

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

Scoring transitions

Scoring input features

Old Scoring Function

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

Stacked Weight Vector

Stacked Feature Vector

New Scoring Function

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \qquad \varphi^j(a, a' \mid x) = \begin{bmatrix} \varphi_1^j(a \mid x) \\ \varphi_2(a, a') \end{bmatrix}$$

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

$$\varphi_2(a, a') = \begin{bmatrix} 1_{\left[(a=Noun)\wedge(a'=Start)\right]} \\ 1_{\left[(a=Noun)\wedge(a'=Noun)\right]} \\ 1_{\left[(a=Noun)\wedge(a'=Verb)\right]} \\ 1_{\left[(a=Verb)\wedge(a'=Start)\right]} \\ 1_{\left[(a=Verb)\wedge(a'=Noun)\right]} \\ 1_{\left[(a=Verb)\wedge(a'=Verb)\right]} \end{bmatrix}$$

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

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\varphi^j(a, a' \mid x) = \begin{bmatrix} \varphi_1^j(a \mid x) \\ \varphi_2^j(a, a') \end{bmatrix}$$

$$w_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \varphi_1^j(a \mid x) = \begin{bmatrix} 1_{[(a=Noun)\wedge(x^j='Fish')]} \\ 1_{[(a=Noun)\wedge(x^j='Sleep')]} \\ 1_{[(a=Verb)\wedge(x^j='Fish')]} \\ 1_{[(a=Verb)\wedge(x^j='Sleep')]} \end{bmatrix}$$

Old Notation:

	$O_{N,*}$	$O_{V,*}$
$O_{*,Fish}$	2	1
$O_{*,Sleep}$	1	0

Old Notation:

	$A_{N,*}$	$A_{V,*}$
$A_{*,N}$	-2	1
$A_{*,V}$	2	-2
$A_{*,Start}$	1	-1

$$w_2 = \begin{bmatrix} 1 \\ -2 \\ 2 \\ -1 \\ 1 \\ -2 \end{bmatrix} \quad \varphi_2(a, a') = \begin{bmatrix} 1_{[(a=Noun)\wedge(a'=Start)]} \\ 1_{[(a=Noun)\wedge(a'=Noun)]} \\ 1_{[(a=Noun)\wedge(a'=Verb)]} \\ 1_{[(a=Verb)\wedge(a'=Start)]} \\ 1_{[(a=Verb)\wedge(a'=Noun)]} \\ 1_{[(a=Verb)\wedge(a'=Verb)]} \end{bmatrix}$$

Why New Notation?

- Easier to reason about:
  - Computing Predictions
  - Computing Gradients
  - Extensions (just generalize ϕ)

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

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

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

$$\varphi_2(a,b) = \begin{bmatrix} 1_{\left[(a=Noun)\wedge(b=Start)\right]} \\ 1_{\left[(a=Noun)\wedge(b=Noun)\right]} \\ 1_{\left[(a=Noun)\wedge(b=Verb)\right]} \\ 1_{\left[(a=Verb)\wedge(b=Start)\right]} \\ 1_{\left[(a=Verb)\wedge(b=Noun)\right]} \\ 1_{\left[(a=Verb)\wedge(b=Verb)\right]} \end{bmatrix}$$

Conditional Random Fields

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

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

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

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

$$\varphi_1^j(a \mid x) = \begin{bmatrix} 1_{[(a=Noun)\wedge(x^j='Fish')]} \\ 1_{[(a=Noun)\wedge(x^j='Sleep')]} \\ 1_{[(a=Verb)\wedge(x^j='Fish')]} \\ 1_{[(a=Verb)\wedge(x^j='Sleep')]} \end{bmatrix}$$

$$\varphi_2(a,b) = \begin{bmatrix} 1_{[(a=Noun)\wedge(b=Start)]} \\ 1_{[(a=Noun)\wedge(b=Noun)]} \\ 1_{[(a=Noun)\wedge(b=Verb)]} \\ 1_{[(a=Verb)\wedge(b=Start)]} \\ 1_{[(a=Verb)\wedge(b=Noun)]} \\ 1_{[(a=Verb)\wedge(b=Verb)]} \end{bmatrix}$$

x = “Fish Sleep” y = (N,V)

$$w_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \varphi_1^j(a \mid x) = \begin{bmatrix} 1_{[(a=Noun)\wedge(x^j='Fish')]} \\ 1_{[(a=Noun)\wedge(x^j='Sleep')]} \\ 1_{[(a=Verb)\wedge(x^j='Fish')]} \\ 1_{[(a=Verb)\wedge(x^j='Sleep')]} \end{bmatrix}$$

$$w_2 = \begin{bmatrix} 1 \\ -2 \\ 2 \\ -1 \\ 1 \\ -2 \end{bmatrix} \qquad \varphi_2^j(a, a') = \begin{bmatrix} 1_{[(a=Noun)\wedge(a'=Start)]} \\ 1_{[(a=Noun)\wedge(a'=Noun)]} \\ 1_{[(a=Noun)\wedge(a'=Verb)]} \\ 1_{[(a=Verb)\wedge(a'=Start)]} \\ 1_{[(a=Verb)\wedge(a'=Noun)]} \\ 1_{[(a=Verb)\wedge(a'=Verb)]} \end{bmatrix}$$

$$P(N,V \mid x = \text{"Fish Sleep"}) = \frac{1}{Z(x)} \exp\left\{ w_1^T \varphi_1^1(N,x) + w_2^T \varphi_2(N,Start) + w_1^T \varphi_1^2(V,x) + w_2^T \varphi_2(V,N) \right\}$$

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

Z(x) = Sum (

y	exp(F(y,x))
(N,N)	exp(2+1+1-2) = exp(2)
(N,V)	exp(2+1+0+1) = exp(4)
(V,N)	exp(1-1+1+2) = exp(3)
(V,V)	exp(1-1+0-2) = exp(-2)

)

Summary of New Notation

- Generic Logistic Model Notation:

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

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

- Define feature function:
  - Linear model in feature representation
  - Applies to both CRFs and basic LR

Computing Predictions (Viterbi)

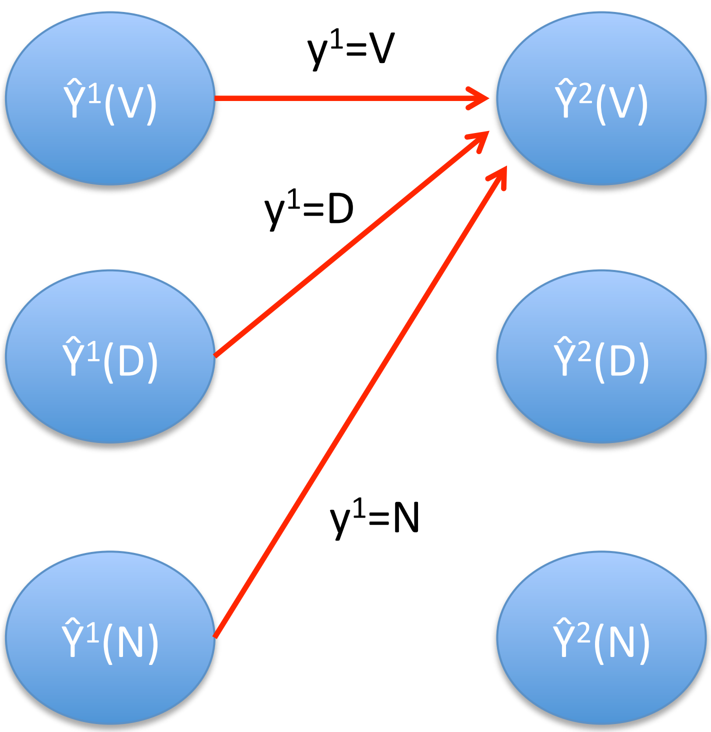
$$\underset{y}{\operatorname{argmax}} P(y \mid x) = \underset{y}{\operatorname{argmax}} F(y, x)$$

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

Maintain length-k prefix solutions	$\hat{Y}^k(T) = \left(\underset{y^{1:k-1}}{\operatorname{argmax}} F(y^{1:k-1} \oplus T, x) \right) \oplus T$
Recursively solve for length-(k+1) solutions	$\hat{Y}^{k+1}(T) = \left(\underset{y^{1:k} \in \left\{ \hat{Y}^k(T) \right\}_T}{\operatorname{argmax}} F(y^{1:k} \oplus T, x) \right) \oplus T$ $= \left(\underset{y^{1:k} \in \left\{ \hat{Y}^k(T) \right\}_T}{\operatorname{argmax}} F(y^{1:k}, x) + w^T \varphi^{k+1}(T, y^k, x) \right) \oplus T$
Predict via best length-M solution	$\underset{y}{\operatorname{argmax}} F(y, x) = \underset{y \in \left\{ \hat{Y}^M(T) \right\}_T}{\operatorname{argmax}} F(y, x)$

Solve: $$\hat{Y}^2(V) = \left(\operatorname*{argmax}_{y^1 \in \{\hat{Y}^1(T)\}_T} F(y^1, x) + w^T \varphi^2(V, y^1 \mid x) \right) \oplus V$$

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

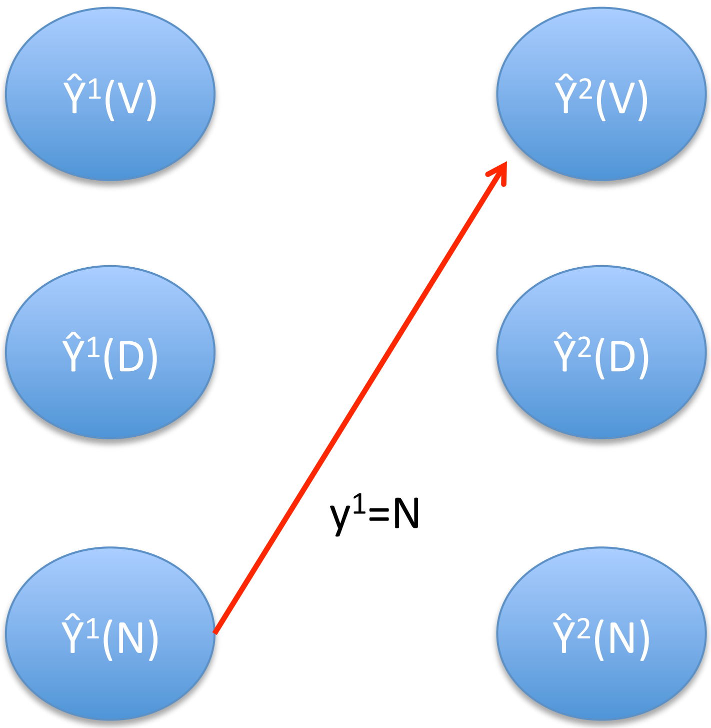


$\hat{Y}^1(T)$ is just T

Solve:

$$\hat{Y}^2(V) = \left(\underset{y^1 \in \{\hat{Y}^1(T)\}_T}{\operatorname{argmax}} F(y^1, x) + w^T \varphi^2(V, y^1 \mid x) \right) \oplus V$$

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



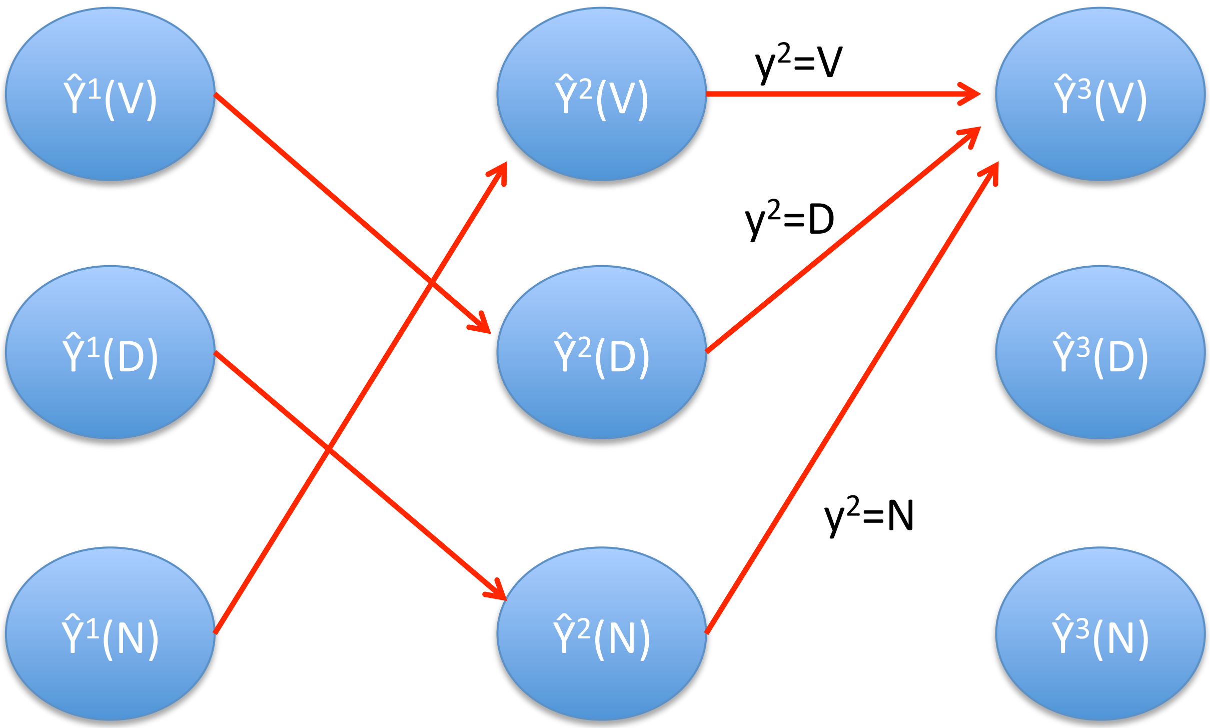
$\hat{Y}^1(T)$ is just T

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

Solve: $$\hat{Y}^3(V) = \left(\underset{y^{1:2} \in \left\{ \hat{Y}^2(T) \right\}_T}{\operatorname{argmax}} F(y^{1:2}, x) + w^T \varphi^j(V, y^2 \mid x) \right) \oplus V$$

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

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



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

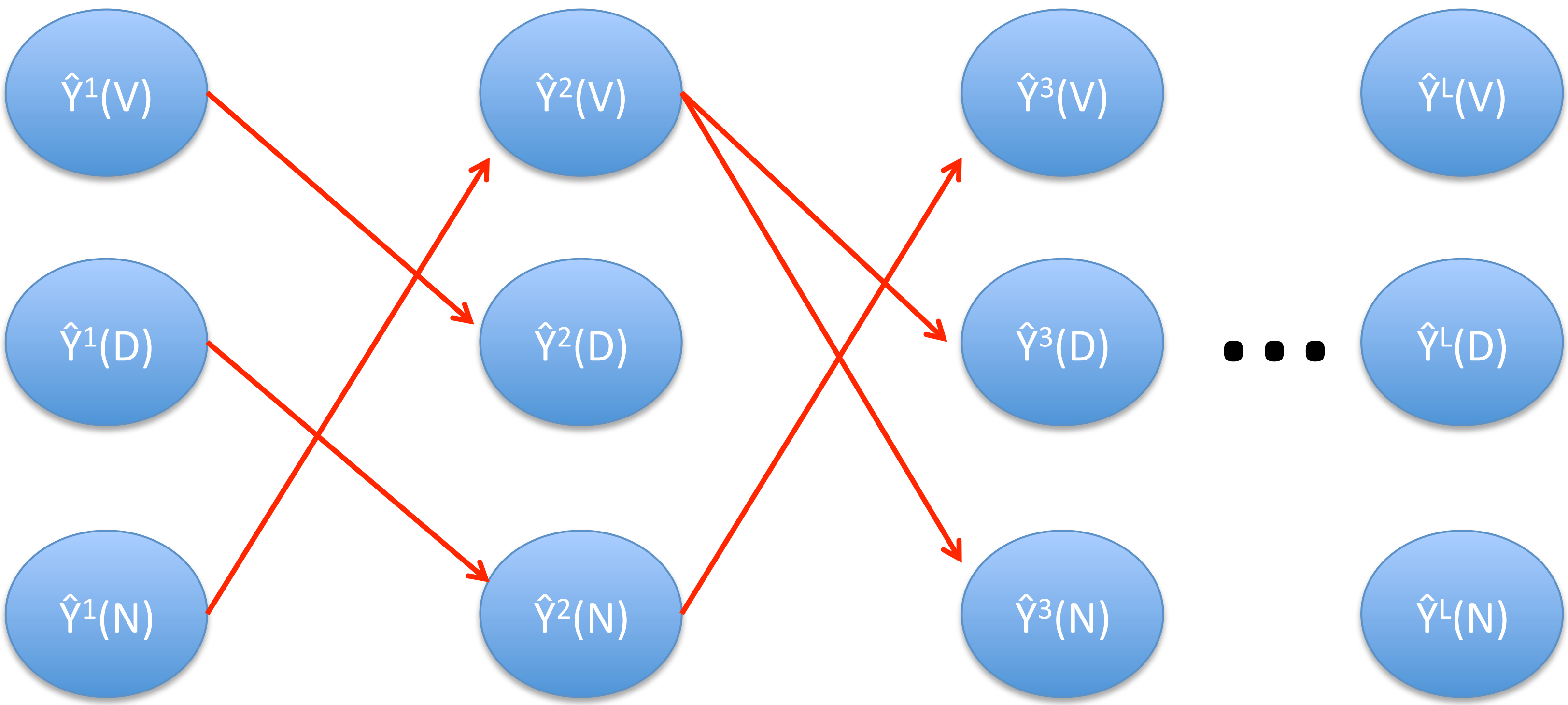
Ex: $\hat{Y}^2(V) = (N, V)$

Solve: $$\hat{Y}^M(V) = \left(\operatorname*{argmax}_{y^{1:M-1} \in \left\{\hat{Y}^{M-1}(T)\right\}_T} F(y^{1:M-1}, x) + w^T \varphi^M(V, y^{M-1} \mid x) \right) \oplus V$$

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



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

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

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

Solve: $$\hat{Y}^M(V) = \left(\underset{y^{1:M-1} \in \left\{ \hat{Y}^{M-1}(T) \right\}_T}{\operatorname{argmax}} F(y^{1:M-1}, x) + w^T \varphi^M(V, y^{M-1} \mid x) \right) \oplus V$$

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:

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

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

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

Challenges:

- Compute Z(x) efficiently
- Numerical instability

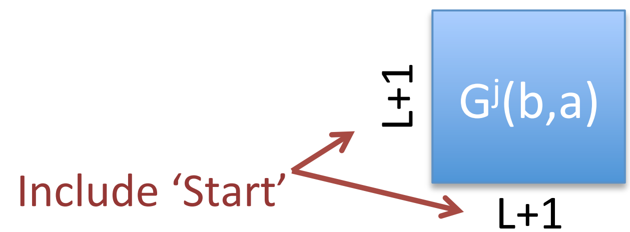
See course notes.

Matrix Semiring

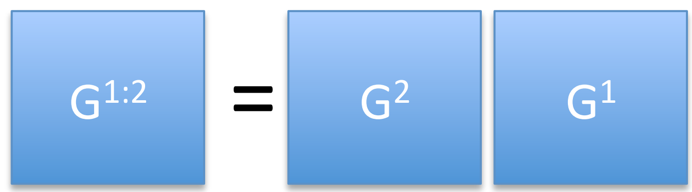
$$Z(x) = \sum_{y'} \prod_{j=1}^{M} G^j(y'^j, y'^{j-1})$$

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

Matrix Version of G^j



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



$$G^{i:j}(b,a) \equiv$$

See course notes.

Computing Partition Function

- Consider Length-1 (M=1)

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

Sum column 'Start' of G^1!

- M=2

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

Sum column 'Start' of $G^{1:2}$!

- General M

Sum column 'Start' of $G^{1:M}$!

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

$$G^{1:M} = G^{M} \; G^{M-1} \; \cdots \; G^{2} \; G^{1}$$

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 va Scaling:

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

Example Scaling Factor:

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

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

$$\log(P(y|x)) = F(y,x) - \log(Z(x))$$

Use log probs instead!

See course notes.

Training

(Stochastic) Gradient Descent

- Minimize log loss of training data:

$$\operatorname*{argmin}_w \sum_{i=1}^{N} -\log P(y_i \mid x_i) = \operatorname*{argmin}_w \sum_{i=1}^{N} -F(y_i, x_i) + \log\big(Z(x_i)\big)$$

$$\partial_w - F(y,x) = -\sum_{j=1}^{M} \varphi^j(y^j, y^{j-1} \mid x)$$

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

$$S = \big\{(x_i, y_i)\big\}_{i=1}^{N}$$

See course notes.

Optimality Condition

$$\operatorname*{argmin}_{\Theta} \sum_{i=1}^{N} -\log P(y_i \mid x_i) = \operatorname*{argmin}_{\Theta} \sum_{i=1}^{N} -F(y_i, x_i) + \log\left(Z(x_i)\right)$$

- Optimality condition:

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

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

$$\sum_{i=1}^{N}\sum_{j=1}^{M_i} 1_{\left[\left(y_i^j=b\right)\wedge\left(y_i^{j-1}=a\right)\right]}\varphi^j(b,a \mid x_i) = \sum_{i=1}^{N}\sum_{j=1}^{M_i} P(y^j = b, y^{j-1} = a \mid x_i)\varphi^j(b,a \mid x_i)$$

$$S = \left\{(x_i, y_i)\right\}_{i=1}^{N}$$

See course notes.

Computing P(y^j=b,y^j-1=a|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

$$\alpha^1(a) = G^1(a, Start)$$

$$\alpha^j(a) = \sum_{a'} \alpha^{j-1}(a') G^j(a, a')$$

$$\beta^M(b) = 1$$

$$\beta^j(b) = \sum_{b'} \beta^{j+1}(b') G^{j+1}(b', b)$$

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

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

See course notes.

Dealing w/ Numerical Instability

- Numerical instability: α^j & β^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 α^j & β^j vectors:

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

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

See course notes.

Recap: Conditional Random Fields

- "Log-Linear" 1^{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:

$$F(y,x)$$

- 2nd Order Sequential Model:

$$F(y,x) \equiv \sum_{j=1}^{M} \left[w^T \varphi^j(y^j, y^{j-1}, y^{j-2} \mid x) \right]$$

- Classification Model:

$$F(y,x) \equiv w^T \varphi(y \mid x)$$

- Efficient Prediction:

$$\operatorname*{argmax}_y F(y,x)$$

More Elaborate Scoring Functions

- Structure encoded by linear scoring function:

$$F(y,x)$$

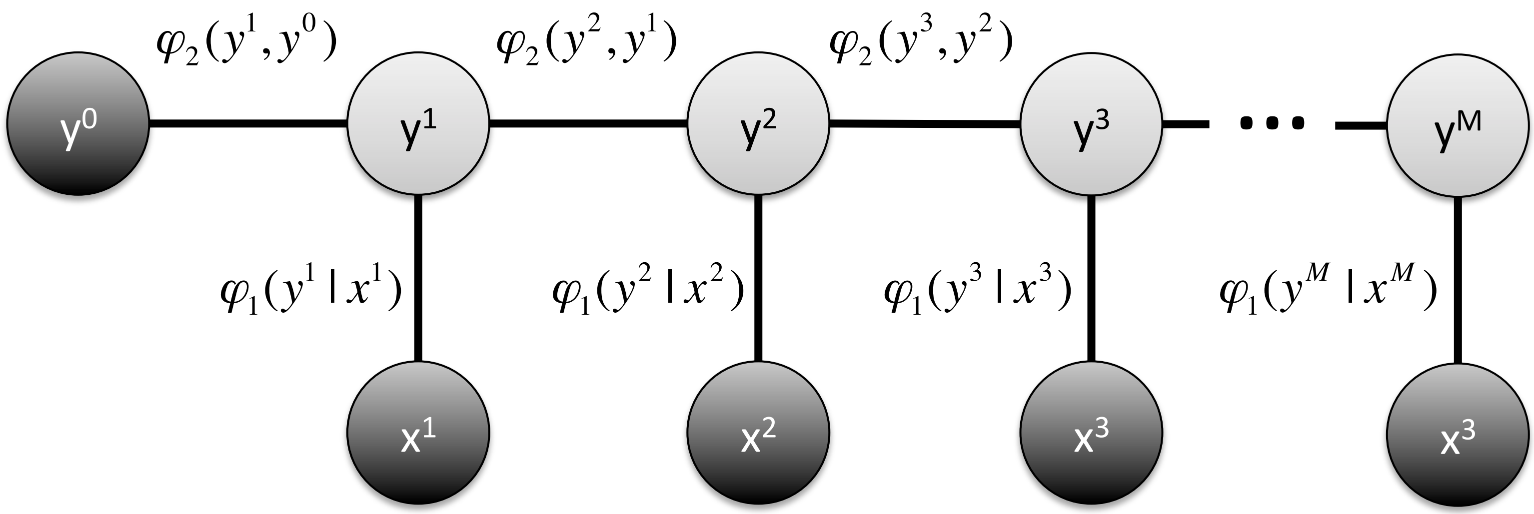
Remainder of Lecture:
Tour of Structured Prediction Models
Some Might be Interesting to You…

- Efficient Prediction:

$$\underset{y}{\operatorname{argmax}}\, F(y,x)$$

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^j!

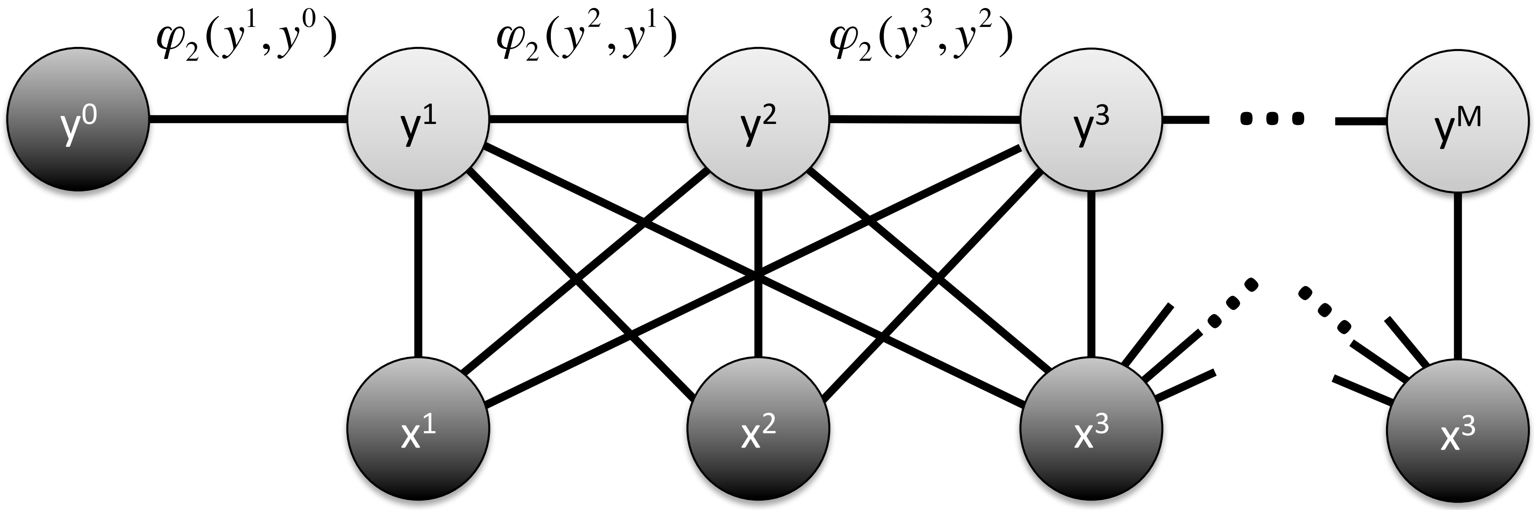
https://piazza.com/cornell/fall2013/btry6790cs6782/resources

http://www.cs.cmu.edu/~guestrin/Class/10708/

https://www.coursera.org/course/pgm

Graphical Models

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



Graph structure encodes structural dependencies between y^j!

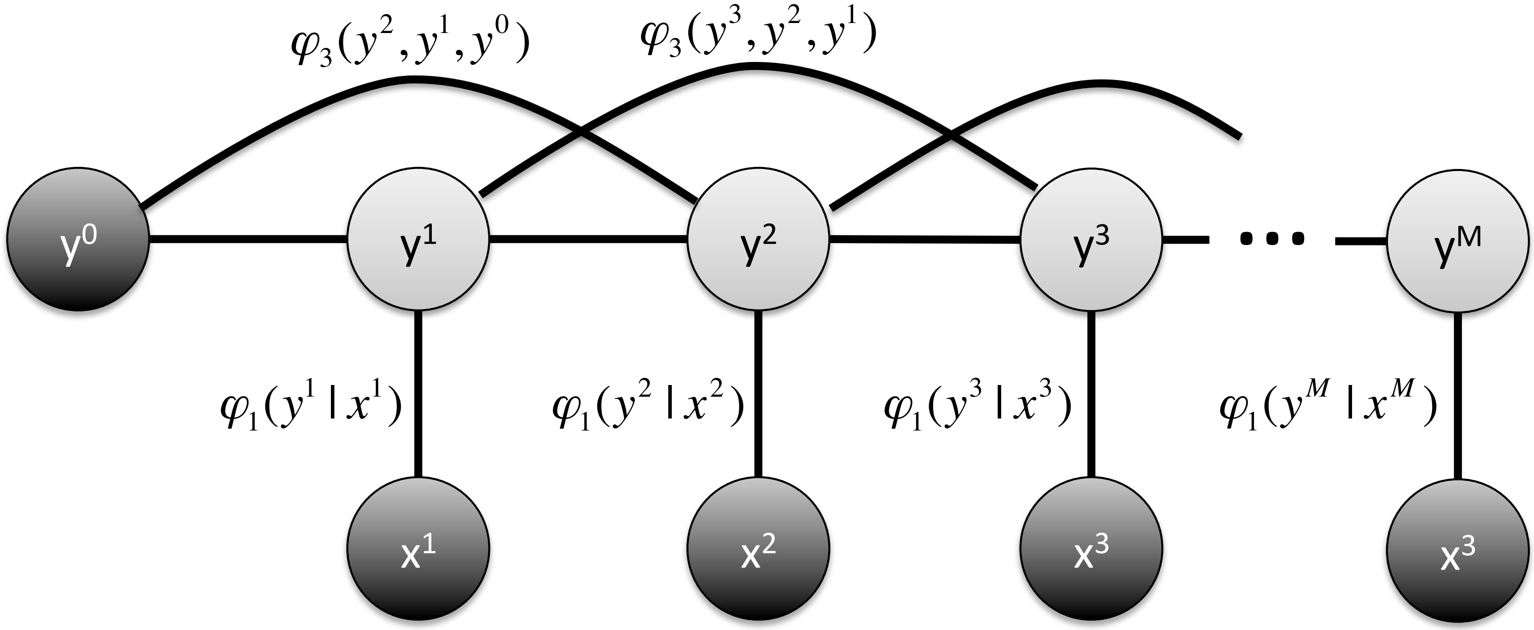
https://piazza.com/cornell/fall2013/btry6790cs6782/resources

http://www.cs.cmu.edu/~guestrin/Class/10708/

https://www.coursera.org/course/pgm

Graphical Models

$$\varphi^{j}(a,b,c \mid x) = \begin{bmatrix} \varphi_1(a \mid x^{j}) \\ \varphi_3(a,b,c) \end{bmatrix}$$



Graph structure encodes structural dependencies between y^j!

Features depend on cliques in graphical model representation.

https://piazza.com/cornell/fall2013/btry6790cs6782/resources

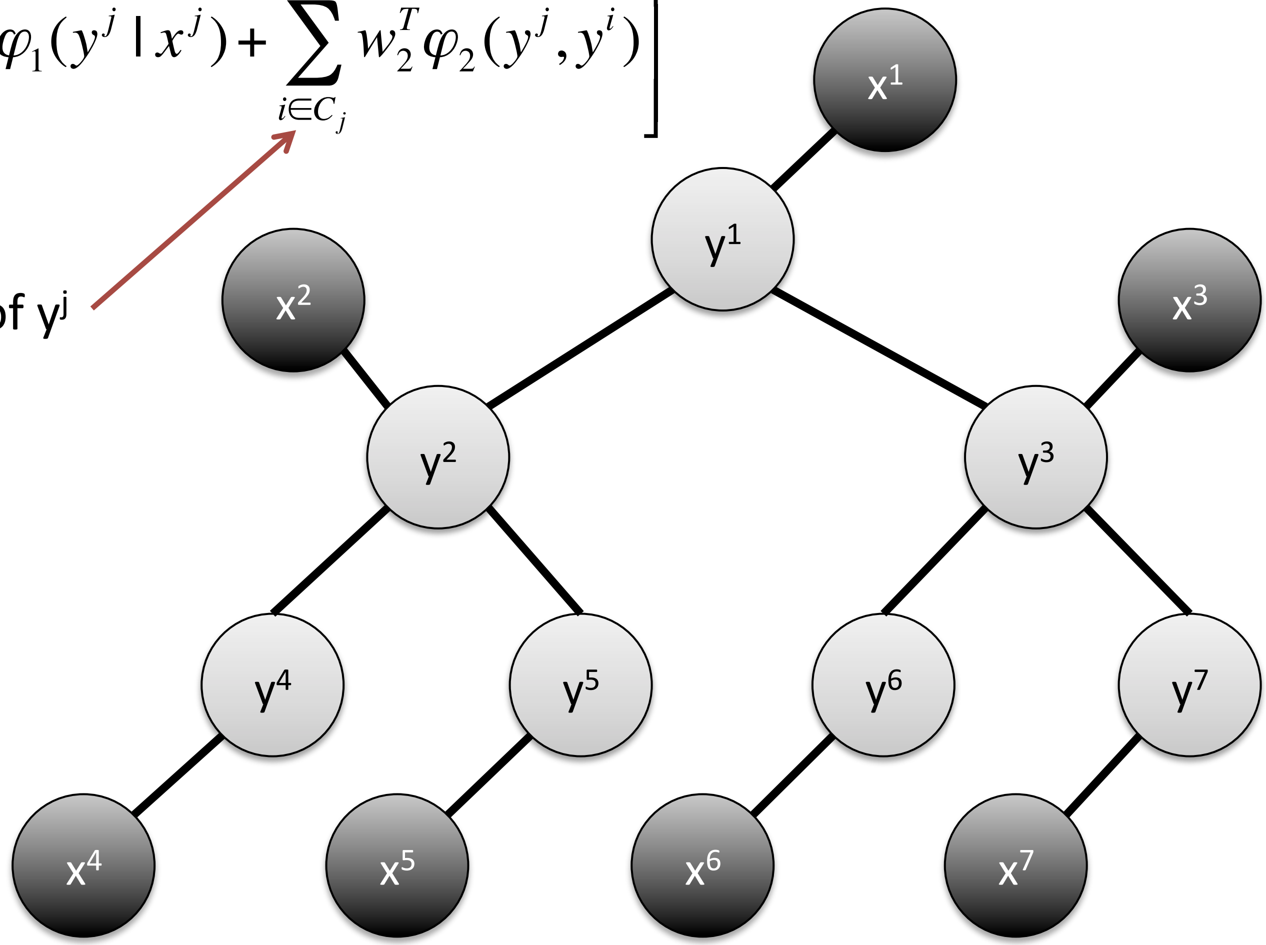
http://www.cs.cmu.edu/~guestrin/Class/10708/

https://www.coursera.org/course/pgm

Tree Structured Models

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

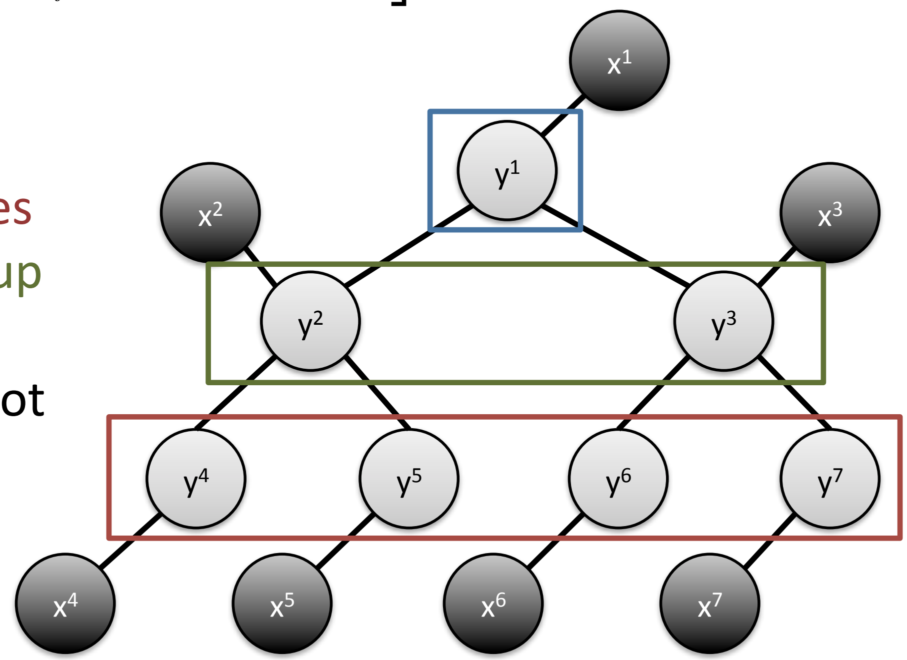
Child nodes of y^j



Prediction via Dynamic Programming

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

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

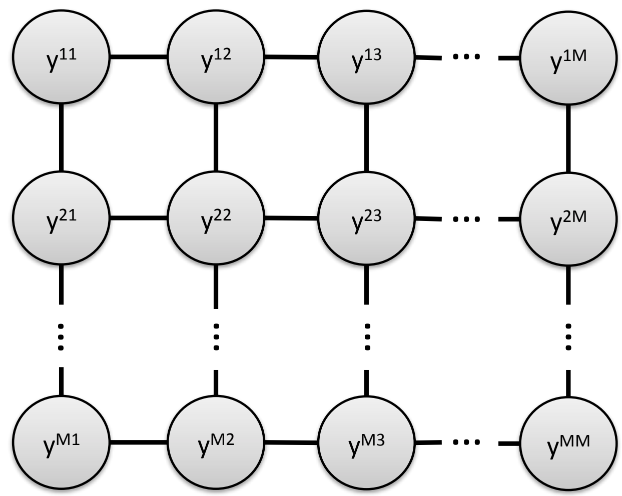
Loopy Graphical Models

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!

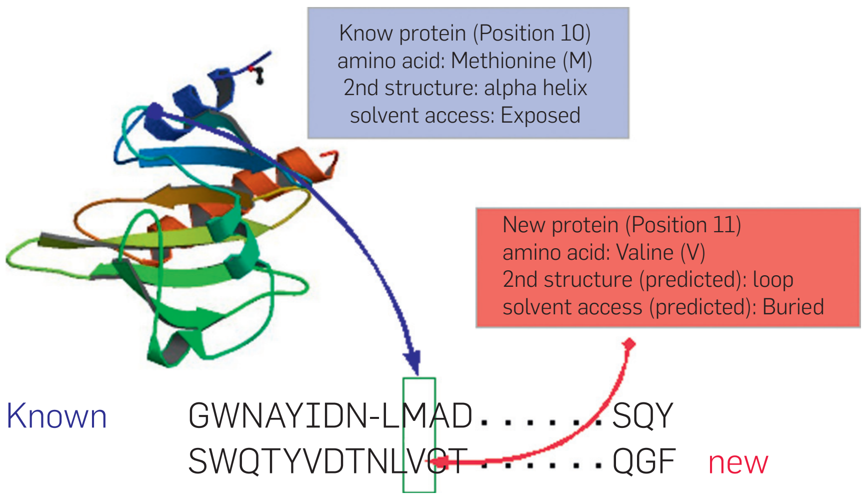
$$\underset{y}{\operatorname{argmax}}\, F(y,x)$$



x suppressed for brevity

http://vision.middlebury.edu/MRF/
http://www.seas.upenn.edu/~taskar/pubs/mmamn.pdf
http://www.cs.cornell.edu/~rdz/Papers/SZ-visalg99.pdf

String Alignment



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

Predict Folding Structure & Function of Protein

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

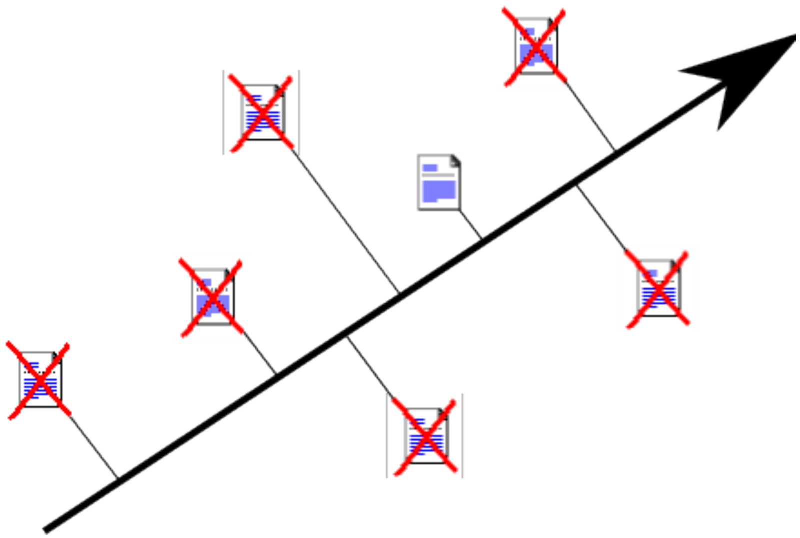
Larger Database **G** of Homologies (proteins w/ known similarities to **D**)

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

$F(y,x)$	encodes score of different types of substitutions, insertions & deletions

http://www.cs.cornell.edu/People/tj/publications/yu_etal_06a.pdf
See Also: http://journals.plos.org/ploscompbiol/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.

$$y^{ij} \in \{-1,+1\}$$

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

x = query & set of results

y = ranking

$$\underset{y}{\operatorname{argmax}}\, F(y,x) = \operatorname{sort}\left\{w^T\varphi(x^j)\right\}_j$$

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

http://research.microsoft.com/en-us/um/people/cburges/tech_reports/MSR-TR-2010-82.pdf

http://www.yisongyue.com/publications/sigir2007_svmmap.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/

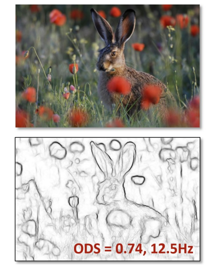
http://www.cs.cornell.edu/Courses/cs778/2006fa/

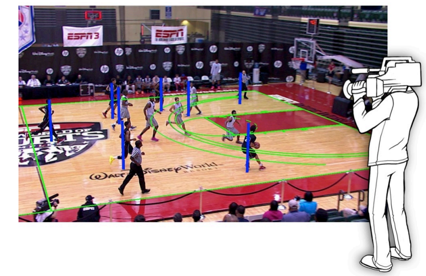
https://www.sites.google.com/site/spflodd/

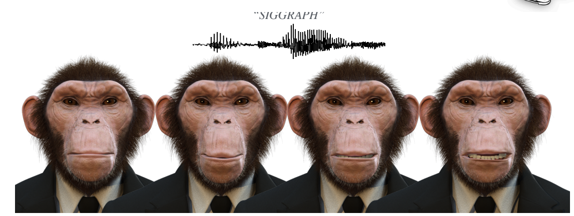
http://www.cs.cornell.edu/People/tj/publications/joachims_06b.pdf

Next Week

- **Lecture Tuesday:**
  - Learning Reductions
  - Recent Applications







- **NO Lecture Thursday:**
  - Student-Faculty Conference

- **Recitation Thursday:**
  - Review of Conditional Random Fields