Caltech

Machine Learning & Data Mining

CS/CNS/EE 155

Lecture 7:

Recap of CRFs & Structural SVMs

Announcements

- Homework 2 due in ~1 week
  - Discussion Forums on Moodle
- Homework 3 released in ~1 week
  - Will be easier than HW2, but has 1 proof question.
- First mini-project released in ~1 week
  - Kaggle competition that will run for 3 weeks.
  - You can work in groups of up to 3.

Today

- Recap of Conditional Random Fields
  - Simpler notation
  - Harder to relate back to earlier lectures on HMMs
  - Easier to relate to more general structured prediction problems
    - I.e., beyond pairwise linear chain models

- Introduction to Structural SVMs

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)

Recap: General Multiclass

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

$\underbrace{\hspace{6em}}_{L=6}$

- x = “Fish sleep”
- y = (N, V)
- Multiclass prediction:
  – All possible length-M sequences as different class
  – (D, D), (D, N), (D, V), (D, Adj), (D, Adv), (D, Pr) (N, D), (N, N), (N, V), (N, Adj), (N, Adv), ...
- **L^M classes!**
  – **Length 2: 6^2 = 36!**

Recap: General Multiclass

- x = “Fish sleep”
- y = (N, V)

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

$\underbrace{\hspace{6em}}_{L=6}$

- M
  - …ss
  - …)
- **L**
  - g

Can Model Everything!
(In Theory)

Exponential Explosion in #Classes!
(Not Tractable)

Recap: Independent Multiclass

x="I fish often"

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

L=6

- Treat each word independently (assumption)
  - Independent multiclass prediction per word
  - Predict for x="I" independently
  - Predict for x="fish" independently
  - Predict for x="often" independently
  - Concatenate predictions.

Assume pronouns are nouns for simplicity.

Recap: Independent Multiclass

x="I fish often"

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

L=6

#Classes = #POS Tags
(6 in our example)

Solvable using standard multiclass prediction.
But ignores context!

Assume pronouns are nouns for simplicity.

1st-Order Sequence Models

- General multiclass:
  - Unique scoring function per entire seq.
  - Very intractable
- Independent multiclass
  - Scoring function per token, apply to each token in seq.
  - Ignores context, low accuracy
- **First-order models**
  - **Scoring function per pair of tokens.**
  - **"Sweet spot" between fully general & ind. multiclass**

1st-Order Sequence Model

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

Scoring Function

Scoring transitions

Scoring input features

- x = "Fish Sleep"

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

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

$u_{N,V}$

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

$w_{V,Fish}$

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

y^0 = special start state

New Notation

Duplicate word features for each label.

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

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

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

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

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

New Notation

One feature for every transition.

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

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

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

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

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

New Notation

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

Scoring transitions

Scoring input features

Old Scoring Function

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

Stacked Weight Vector

Stacked Feature Vector

New Scoring Function

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

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

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

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

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

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

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

Old Notation:

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

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

Old Notation:

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

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

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

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

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

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

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

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

x = “Fish Sleep” y = (N,V)

$$w_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \varphi_1^j(a \mid x) = \begin{bmatrix} 1_{[(a=Noun)\wedge(x^j='Fish')]} \\ 1_{[(a=Noun)\wedge(x^j='Sleep')]} \\ 1_{[(a=Verb)\wedge(x^j='Fish')]} \\ 1_{[(a=Verb)\wedge(x^j='Sleep')]} \end{bmatrix} \quad w_2 = \begin{bmatrix} 1 \\ -2 \\ 2 \\ -1 \\ 1 \\ -2 \end{bmatrix} \quad \varphi_2^j(a,b) = \begin{bmatrix} 1_{[(a=Noun)\wedge(b=Start)]} \\ 1_{[(a=Noun)\wedge(b=Noun)]} \\ 1_{[(a=Noun)\wedge(b=Verb)]} \\ 1_{[(a=Verb)\wedge(b=Start)]} \\ 1_{[(a=Verb)\wedge(b=Noun)]} \\ 1_{[(a=Verb)\wedge(b=Verb)]} \end{bmatrix}$$

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

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

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)

)

Reduction to Independent Multiclass

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left\{F(y,x)\right\} \qquad\qquad Z(x) = \sum_{y'} \exp\left\{F(y',x)\right\}$$

- Suppose: $$F(y,x) \equiv \sum_{j=1}^{M} \left[w^T \varphi^j(y^j \mid x) \right] \qquad \varphi^j(y^j \mid x) = \begin{bmatrix} 1_{[y^j=1]}\phi_1(x^j) \\ \vdots \\ 1_{[y^j=L]}\phi_1(x^j) \end{bmatrix}$$

No pairwise features.

Stack features $\varphi_1(x^j)$ L times

- Then: $$P(y \mid x) = \prod_{j=1}^{M} \frac{\exp\left\{w^T \varphi^j(y^j \mid x)\right\}}{\sum_a \exp\left\{w^T \varphi^j(a \mid x)\right\}} = \prod_{j=1}^{M} \frac{1}{Z(x^j)} \exp\left\{w_{y^j}^T \phi_1(x^j)\right\} = \prod_{j=1}^{M} P(y^j \mid x^j)$$

$$Z(x) = \sum_{y'} \prod_{j=1}^{M} \exp\left\{w^T \varphi^j(y'^j \mid x)\right\} = \prod_{j=1}^{M} \sum_a \exp\left\{w^T \varphi^j(a \mid x)\right\} = \prod_{j=1}^{M} \overbrace{\sum_a \exp\left\{w_a^T \phi_1(x^j)\right\}} = \prod_{j=1}^{M} Z(x^j)$$

Computing Predictions (Viterbi)

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

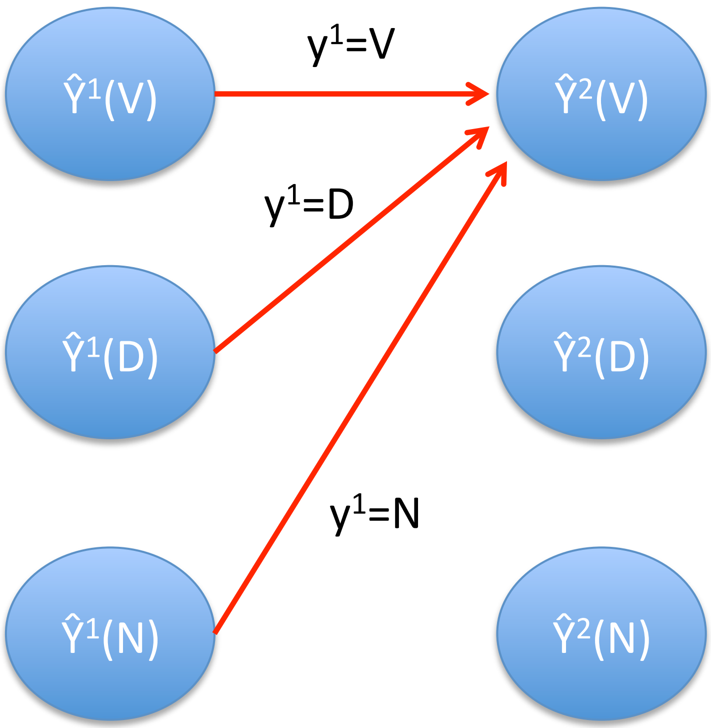
$$F(y^{1:k}, x) \equiv \sum_{j=1}^{k} \left[w^T \varphi^j (y^j, y^{j-1} \mid x) \right]$$

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:

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

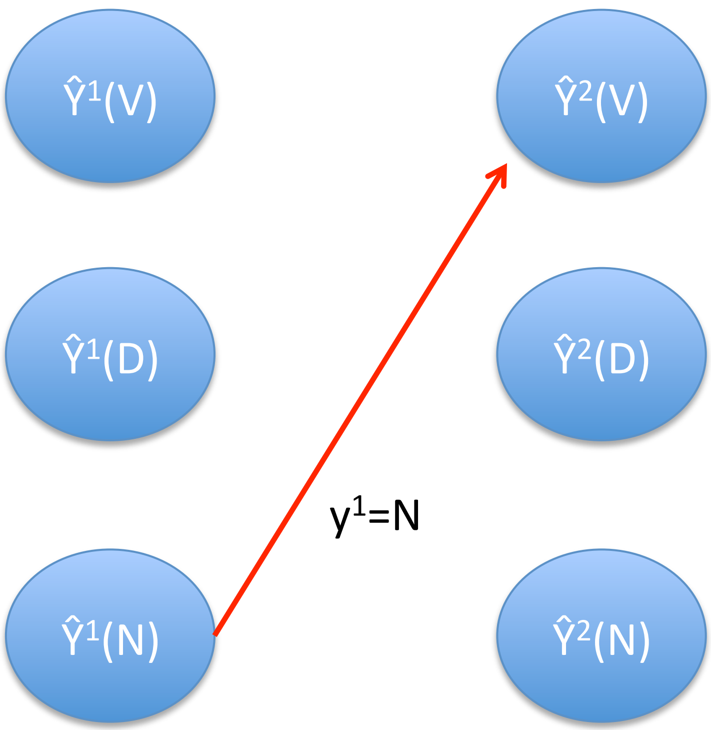
Store each
Ŷ^1(T) & F(Ŷ^1(T),x)



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



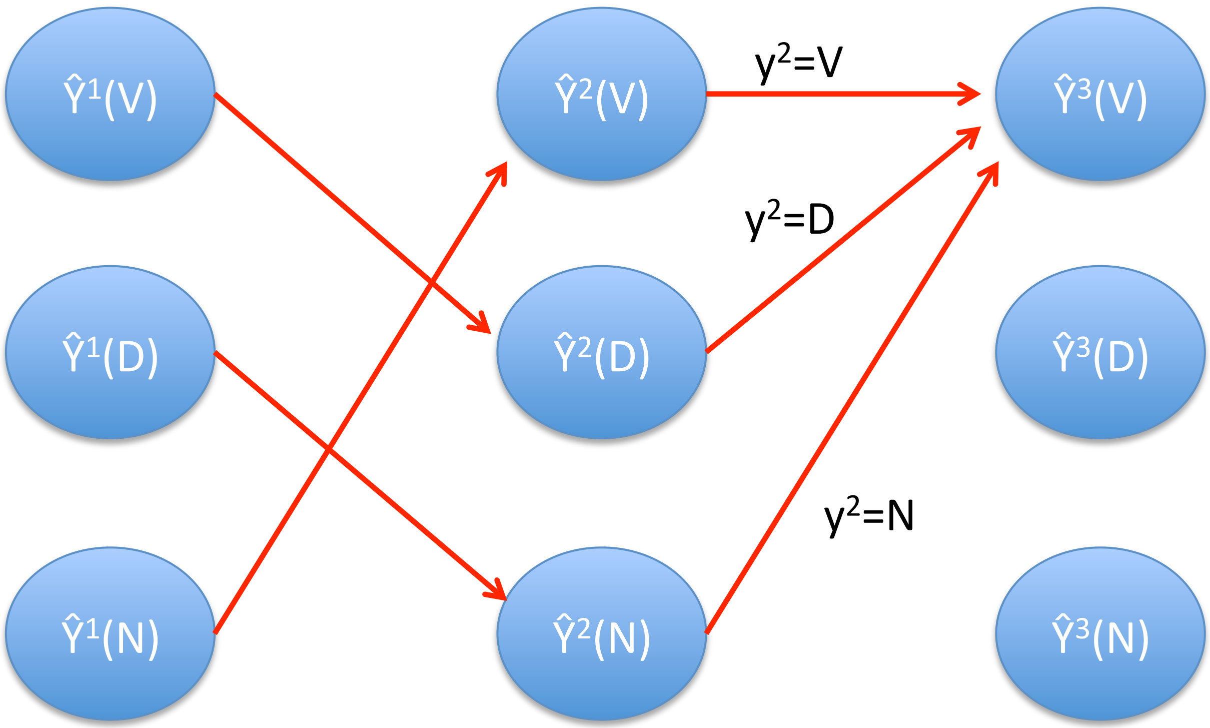
$\hat{Y}^1(T)$ is just T

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

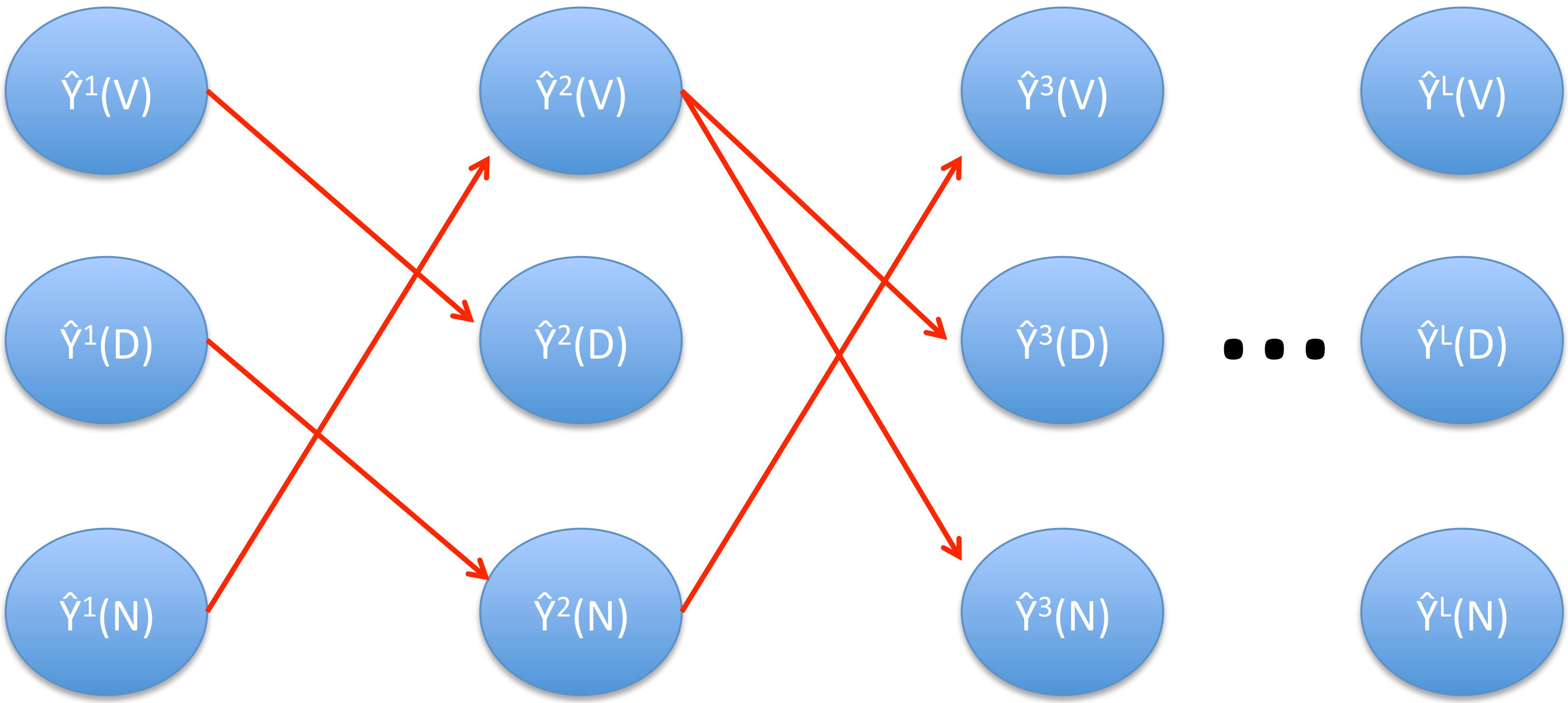
Ex: $\hat{Y}^2(V) = (N, V)$

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

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

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

Matrix Notation: →

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

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

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

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

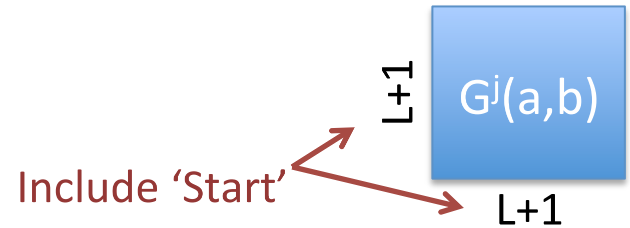
See course notes.

Matrix Semiring

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

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

Matrix Version of G^j



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

$$G^{1:2} = G^2 \; G^1$$

$$G^{i:j}(a,b) \equiv \quad G^{i:j} = G^j \; G^{j-1} \cdots G^{i+1} \; G^i$$

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 $G^{1:M}$

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

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:

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

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

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

Use log probs instead!

See course notes.

Training

- Minimize log loss of training data:

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

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

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

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

Optimality Condition

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

- 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):

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

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

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

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

$$\alpha^j(a) = \sum_b \alpha^{j-1}(b) G^j(a,b)$$

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

$$\beta^j(b) = \sum_a \beta^{j+1}(a) G^{j+1}(a,b)$$

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

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

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 $\phi^j(a,b|x)$
  - Arbitrary features that depend on pairs of labels.
- Train via minimizing neg log likelihood
- Dynamic programming to train and predict

Structural SVMs

1^{st} Order Sequential Model

- Linear in pairwise features $\phi^j(a,b|x)$:

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

- Prediction via maximizing F:

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

- How to train w?

CRF Learning Revisited

- CRFs minimize log loss:

$$\underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} -F(y_i, x_i) + \log\left(Z(x_i)\right)$$

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

- Reduces to independent Logistic Regression
  – When F only depends on single tokens at a time

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

Other Loss Functions?

- General Form: $$\underset{w}{\operatorname{argmin}} \frac{\lambda}{2}\|w\|^2 + \sum_{i=1}^{N} \ell\left(y_i, x_i, F\right)$$
- Log Loss: $$\ell\left(y,x,F\right) = -F(y,x) + \log\left(Z(x)\right)$$
- 0/1 Loss: $$\ell\left(y,x,F\right) = 1_{[h(x)\neq y]} \qquad h(x) = \underset{y}{\operatorname{argmax}}\, F(y,x)$$

  (But not all mis-predictions equally bad...)

- True y = (D,N,V,D,N)
  - y' = (D,N,V,N,N)
  - y'' = (V,D,N,V,V)

Equal 0/1 Loss...
...but clearly y'' is worse

Should have loss function that distinguishes between y' and y''!

Hamming Loss

$$\underset{w}{\operatorname{argmin}} \frac{\lambda}{2}\|w\|^2 + \sum_{i=1}^{N} \ell(y_i, x_i, F)$$

- Hamming Loss: $$\ell(y, x, F) = \sum_{j=1}^{M} 1_{\left[h(x)^j \neq y^j\right]}$$

- True y = (D,N,V,D,N)
  - y' = (D,N,V,N,N)
  - y'' = (V,D,N,V,V)

y'' has much worse hamming loss (loss of 5 vs loss of 1)

(But not continuous!)

Need to define continuous surrogate of Hinge Loss!

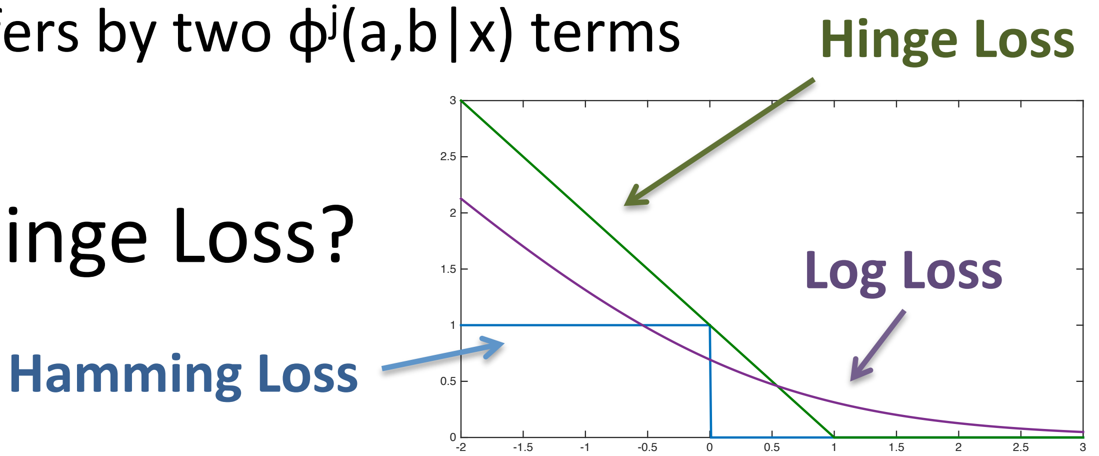
Surrogates of Hamming Loss

- Log Loss ≈ surrogate for Hamming Loss:

$$\ell(y,x,F) = -F(y,x) + \log(Z(x)) \qquad \ell(y,x,F) = \sum_{j=1}^{M} 1_{\left[h(x)^j \neq y^j\right]}$$

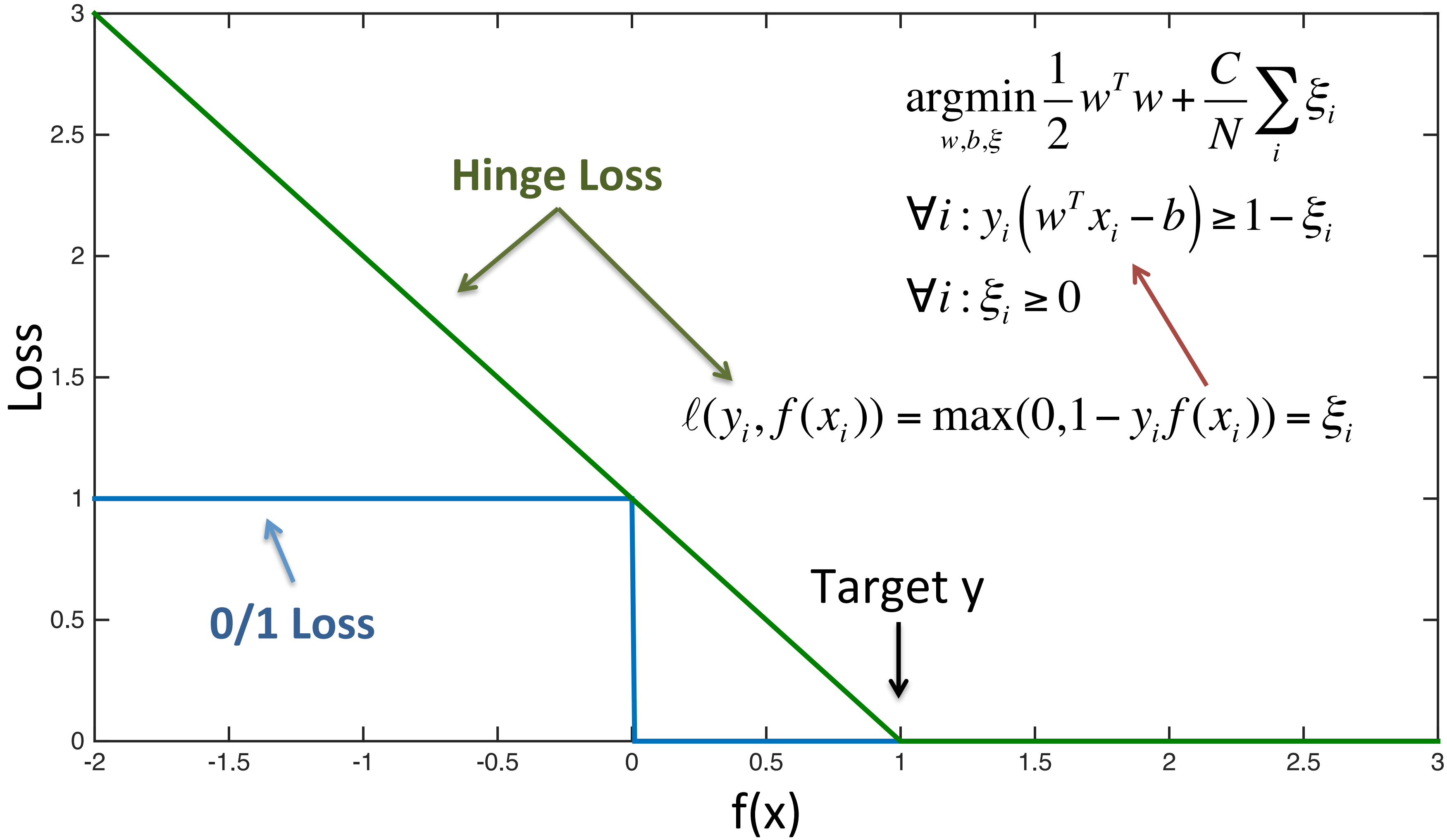
  - Suppose y & y' differ by 1 token
    - Hamming loss differs by 1
    - Log Loss differs by two $\phi^j(a,b|x)$ terms

- What about Hinge Loss?



Original Hinge Loss

(Support Vector Machine)

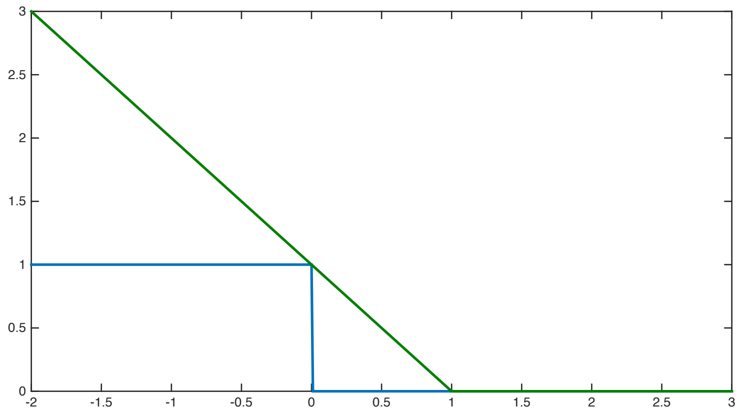


Property of Hinge Loss

$$\underset{w,b,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i : y_i \left(w^T x_i - b \right) \geq 1 - \xi_i$$

$$\forall i : \xi_i \geq 0$$



$$h(x) = \underset{y \in \{-1,+1\}}{\operatorname{argmax}} \, y f(x) = sign(f(x)) \quad \Rightarrow \quad \xi_i \geq 1_{[h(x_i) \neq y_i]}$$

$$\ell(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) = \xi_i$$

Hinge loss = continuous upper bound on 0/1 loss

Hamming Hinge Loss

(Structural SVM)

$$\operatorname*{argmin}_{w,\xi} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

Sometimes normalize by M

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \xi_i \qquad \forall i : \xi_i \geq 0$$

$$h(x) = \operatorname*{argmax}_y F(y, x) \quad \rightarrow \quad F(y_i, x_i) - F(h(x_i), x_i) \leq 0$$

Learned Predictor

$$\rightarrow \quad \xi_i \geq \sum_j 1_{\left[h(x_i)^j \neq y_i^j\right]}$$

$$\ell(y_i, x_i, F) = \max_{y'} \left\{ \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \left(F(y_i, x_i) - F(y', x_i)\right) \right\} = \xi_i$$

Continuous upper bound on Hamming Loss!

Hamming Hinge Loss

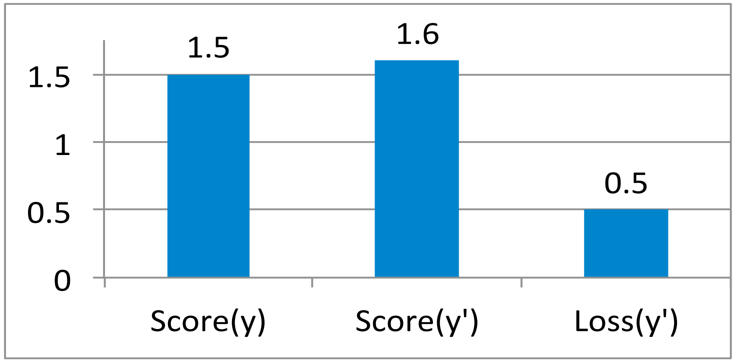
$$\underset{w,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, y' : \underbrace{F(y_i, x_i)}_{\text{Score}(y_i)} - \underbrace{F(y', x_i)}_{\text{Score}(y')} \geq \underbrace{\frac{1}{M_i} \sum_j 1_{\left[y'^j \neq y_i^j\right]}}_{\text{Loss}(y')} - \underbrace{\xi_i}_{\text{Slack}} \qquad \forall i : \xi_i \geq 0$$

Suppose for incorrect $y' = h(x_i)$:

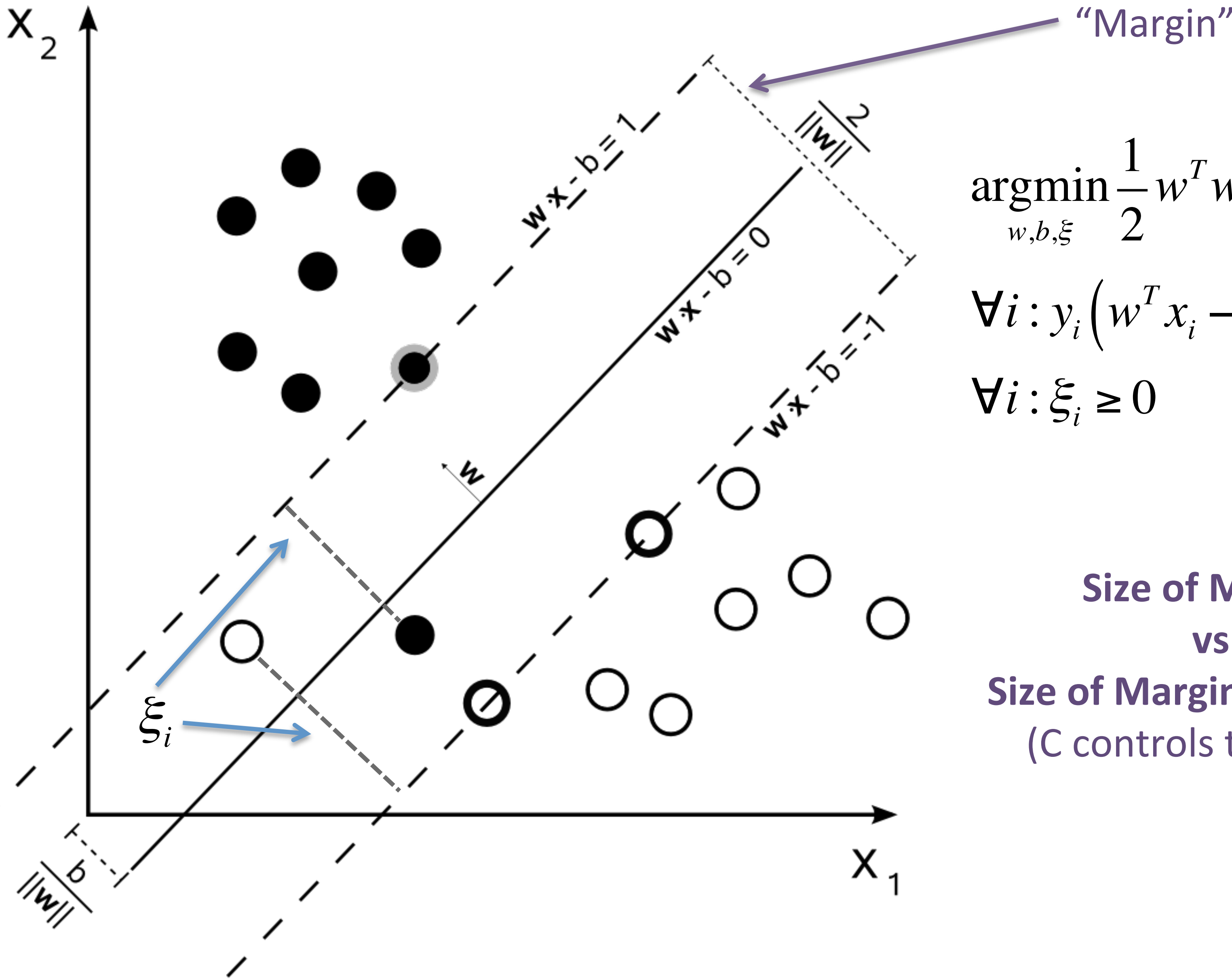
Then:

$$\boxed{\xi_i = 0.6 \geq 0.5 = \frac{1}{M_i} \sum_j 1_{\left[h(x_i)^j \neq y_i^j\right]}}$$



Slack variable upper bounds Hamming Loss!

Recall: Soft-Margin Support Vector Machine



$$\underset{w,b,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i : y_i \left(w^T x_i - b \right) \geq 1 - \xi_i$$

$$\forall i : \xi_i \geq 0$$

Size of Margin
vs
Size of Margin Violations
(C controls trade-off)

Image Source: http://en.wikipedia.org/wiki/Support_vector_machine

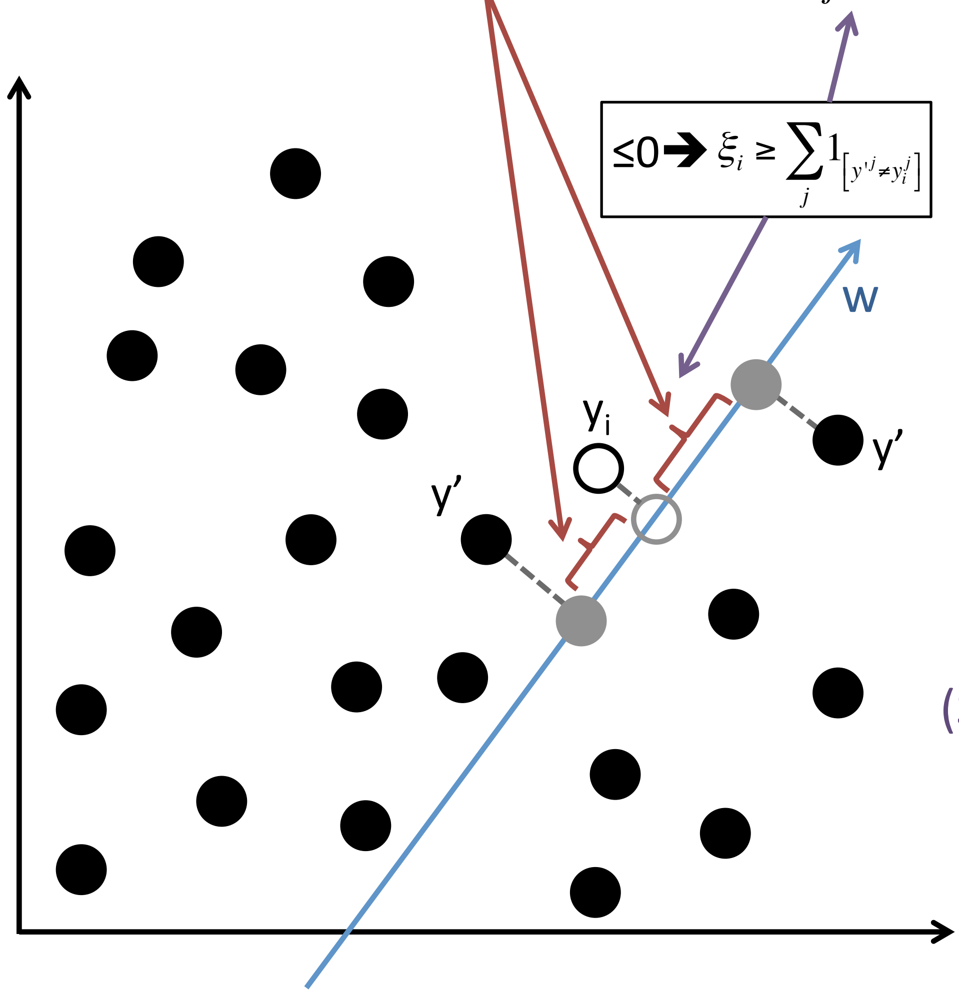
Soft-Margin Structural Support Vector Machine

$$\underset{w,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, y' : \underbrace{F(y_i, x_i) - F(y', x_i)} \geq \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \xi_i \qquad \forall i : \xi_i \geq 0$$

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

High Dimensional Point



Size of Margin

vs

Size of Margin Violations

(C controls trade-off)

(Scale margin by Hamming Loss)

Reduction to Independent Multiclass

$$\operatorname*{argmin}_{w,\xi} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \xi_i \qquad \forall i : \xi_i \geq 0$$

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

$$\varphi^j (y^j \mid x) = \begin{bmatrix} 1_{\left[y^j = 1\right]} \phi_1(x^j) \\ \vdots \\ 1_{\left[y^j = L\right]} \phi_1(x^j) \end{bmatrix}$$

No pairwise features.

Stack features $\varphi_1(x^j)$ L times

$$\forall i, j, a : w_{y_i^j}^T \phi_1(x^j) - w_a^T \phi_1(x^j) \geq 1 - \xi_{ij}$$

Decompose constraints to multiclass hinge loss per token!

Story So Far: Structural SVMs

- Same model class as CRFs:

Linear w.r.t. pairwise features

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

- Same prediction (given trained model):

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

- Train to minimize Hamming Hinge Loss

When is Hinge Loss Better?

- Original Hinge Loss
  - Encourages margin between positive & negative examples
- Hamming Hinge Loss:
  - Encourages margin between correct y & suboptimal y'
  - Margin scaled by hamming loss
- Tend to outperform CRFs when margin exists
  - (with small slack…)

Training Structural SVMs

$$\underset{w,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \xi_i \qquad \forall i : \xi_i \geq 0$$

- How many y' are there?
  - **Exponential!**

- Intractable to enumerate constraints…
  - …let alone solve the optimization problem

Approximate Hinge Loss

- Choose tolerate ε>0:

$$\underset{w,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \frac{1}{M_i} \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \xi_i - \varepsilon \qquad \forall i : \xi_i \geq 0$$

$$h(x) = \underset{y}{\operatorname{argmax}} F(y, x) \quad \rightarrow \quad F(y_i, x_i) - F(h(x_i), x_i) \leq 0$$

Learned Predictor

$$\rightarrow \xi_i \geq \sum_j 1_{\left[h(x_i)^j \neq y_i^j\right]} - \varepsilon$$

$$\ell(y_i, x_i, F) = \max_{y'} \left\{ \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \left(F(y_i, x_i) - F(y', x_i) \right) \right\} = \xi_i + \varepsilon$$

Approximate Hinge Loss

$$\underset{w,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

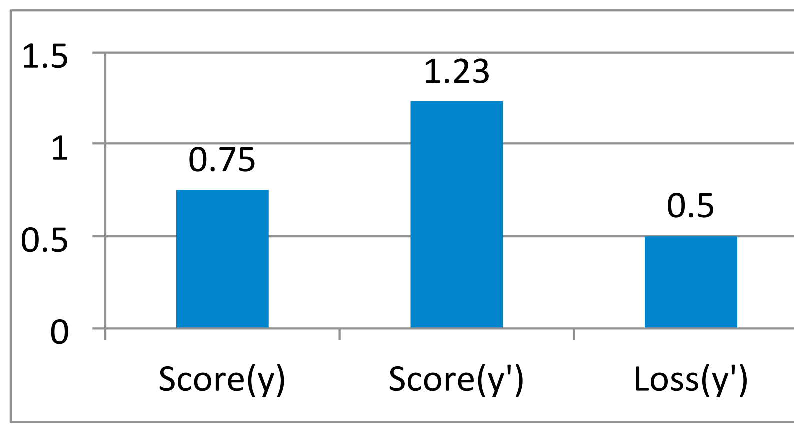
ε=0.02

$$\forall i, y' : \underbrace{F(y_i, x_i)}_{\text{Score}(y_i)} - \underbrace{F(y', x_i)}_{\text{Score}(y')} \geq \underbrace{\frac{1}{M_i} \sum_j 1_{\left[y'^j \neq y_i^j\right]}}_{\text{Loss}(y')} - \underbrace{\xi_i}_{\text{Slack}} - \varepsilon \qquad \forall i : \xi_i \geq 0$$

Suppose for incorrect $y' = h(x_i)$:

Then:

$$\boxed{\xi_i = 0.48 \geq \frac{1}{M_i} \sum_j 1_{\left[h(x_i)^j \neq y_i^j\right]} - \varepsilon}$$



Sneak Preview: Linear Convergence Rate

- Guarantee for any ε>0:

$$\underset{w,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \frac{1}{M_i} \sum_j 1_{\left[y'^j \neq y_i^j\right]} - \xi_i - \varepsilon$$

$$\forall i : \xi_i \geq 0$$

Complexity of Viterbi

- Complexity: $O\left(\frac{h}{\varepsilon}\right)$

Approximately minimize
Hinge Loss w/ tolerance ε

http://www.cs.cornell.edu/people/tj/publications/joachims_etal_09a.pdf

Recap: Structural SVMs

- Minimize Hamming Hinge Loss
  - Maximizes margin between y^i and incorrect y'
  - Desired Margin Depends on Hamming Loss
- Reduces to Conventional Multiclass SVMs
  - If scoring function F decomposes
- Naïve Training takes exponential time
  - Next Lecture: Efficient Approximate Training

Next Lecture

- **Lecture Thursday:**
  - Structural SVM Training
  - General Structured Prediction

- **Recitation Tomorrow:**
  - Gradient Descent for Non-Differentiable Obj.
    - E.g., Lasso, Hinge Loss
  - Accelerated Gradient Descent for Faster Training