

# Machine Learning & Data Mining

## CS/CNS/EE 155

Lecture 8:  
Structural SVMs Part 2 &  
General Structured Prediction

# Announcements

- Homework 2 due next Tuesday
  - 2pm via Moodle
- Homework 3 out next week
  - Due 2 weeks later
  - (Easier than HW2)
- Kaggle Mini-Project out next week
  - Due ~3 weeks later

# Kaggle Mini-Project

- Training set of ~5K labeled data points
  - ~50 features
- Test set of unlabeled data points
  - Submit predictions on test set
- You choose the methods, loss functions, feature manipulations, etc.
  - Expected to do cross validation & model selection
  - Written report
    - Clearly & concisely documenting your process
    - Template will be provided
  - Groups of up to 3
- Due after ~3 weeks

# Today

- Structural SVMs
  - Recap of Previous Lecture
  - Training
- General Structured Prediction
  - Brief Overview

# Recap: 1<sup>st</sup> Order Sequential Model

- Input:  $x = (x^1, \dots, x^M)$
- Predict:  $y = (y^1, \dots, y^M)$ 
  - Each  $y^i$  one of  $L$  labels.
- Linear Model w.r.t. pairwise features  $\phi^j(a, b | x)$ :

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

- Prediction via maximizing  $F$ :

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

**POS Tags:**  
Det, Noun, Verb, Adj, Adv, Prep  
 $L=6$

# Recap: Simple Example

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

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

- “Unary Features”
- “Pairwise Transition Features”

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

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

$x = \text{"Fish Sleep"}$

$y = (N, V)$

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

$$w_2 = \begin{bmatrix} 1 \\ -2 \\ 2 \\ -1 \\ 1 \\ -2 \end{bmatrix}$$

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

$$F(y=(N,V), x = \text{"Fish Sleep"}) = w_1^T \varphi_1^1(N, x) + w_2^T \varphi_2^1(N, Start) + w_1^T \varphi_1^2(V, x) + w_2^T \varphi_2^2(V, N)$$

$$= w_{1,1} + w_{2,1} + w_{1,4} + w_{2,5} = 2 + 1 + 0 + 1 = 4$$

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

$y$

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

# Structured Prediction

- Complex output spaces
  - All possible Part-of-Speech sequences
- Naïve prediction is often exponential time:

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

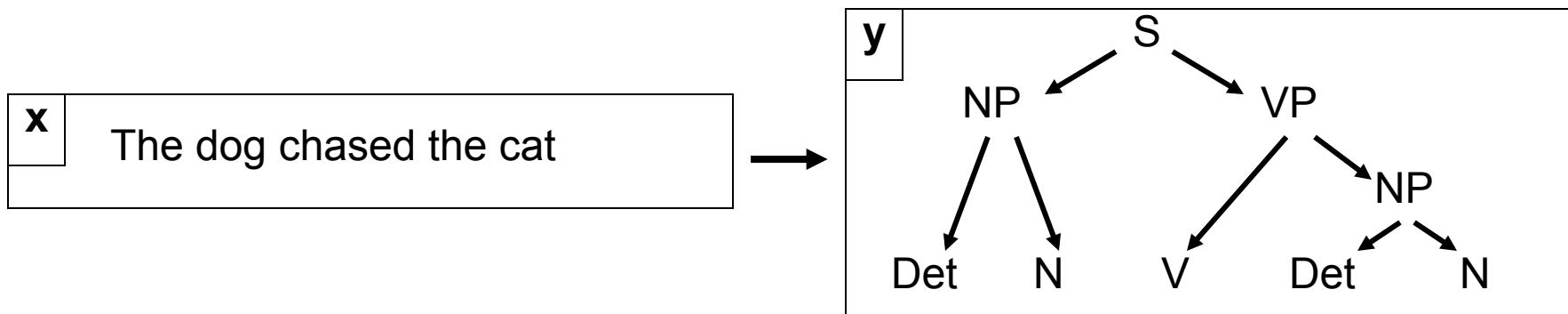
- Evaluation is also multivariate:

$$\ell(y, x, F) = \sum_{j=1}^M \mathbf{1}_{[h(x)^j \neq y^j]}$$

E.g., Hamming Loss

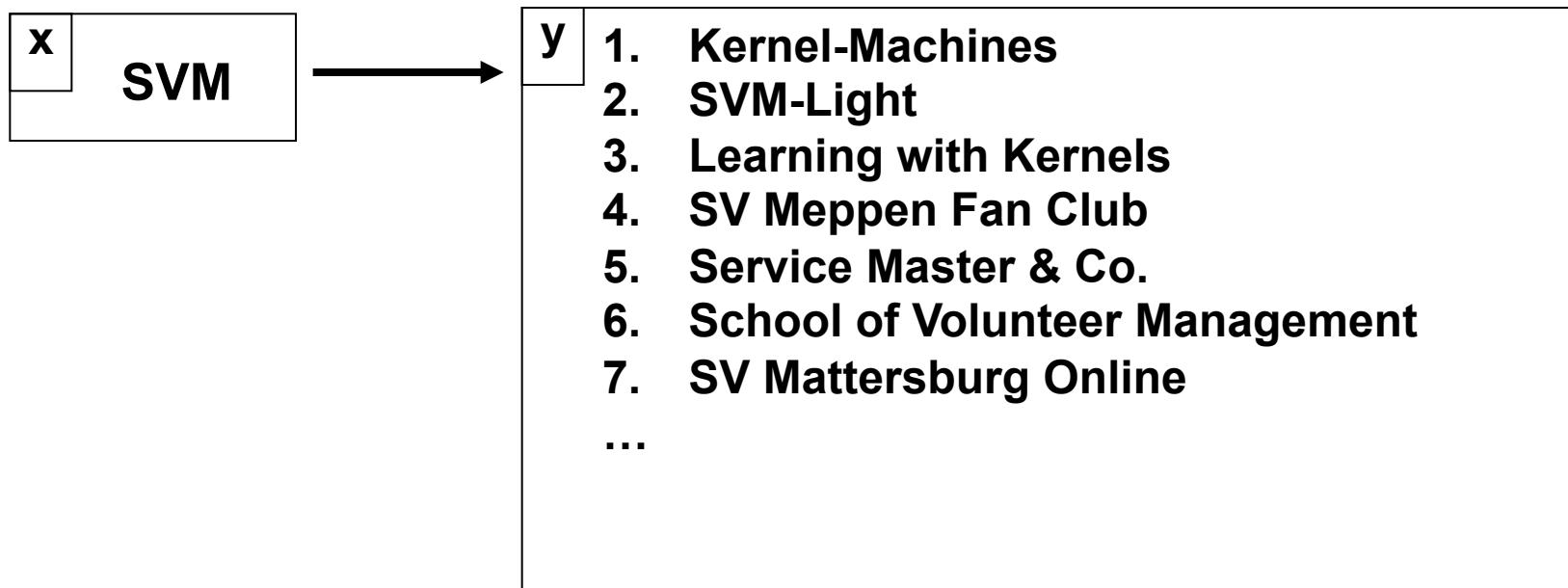
# Examples of Complex Output Spaces

- Natural Language Parsing
  - Given a sequence of words  $x$ , predict the parse tree  $y$ .
  - Dependencies from structural constraints, since  $y$  has to be a tree.

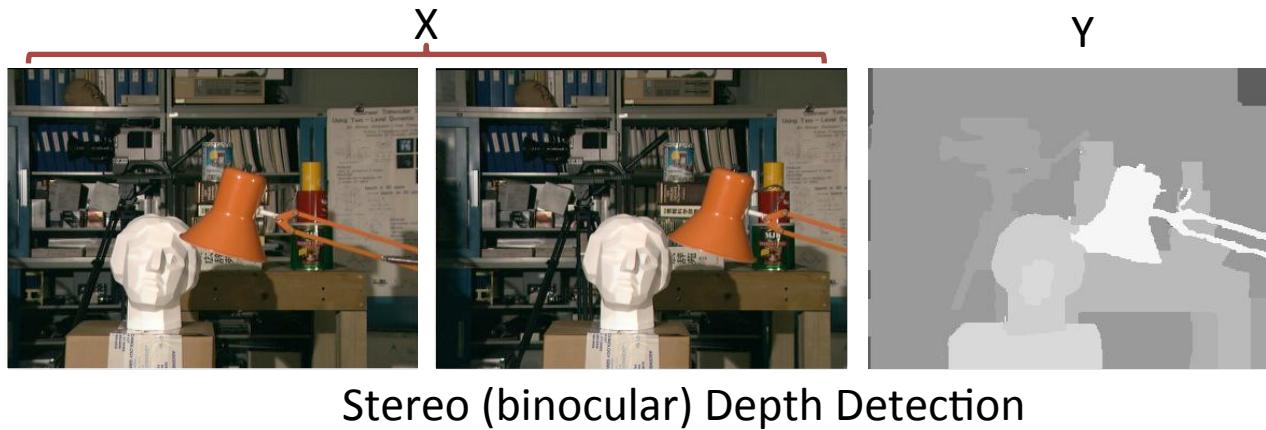
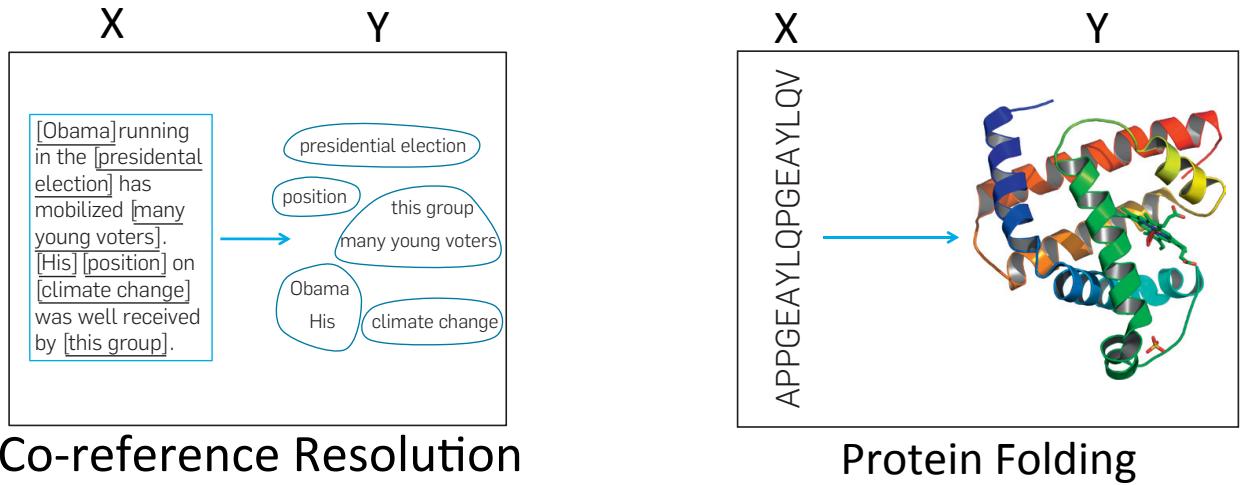


# Examples of Complex Output Spaces

- Information Retrieval
  - Given a query  $x$ , predict a ranking  $y$ .
  - Dependencies between results (e.g. avoid redundant hits)
  - Loss function over rankings (e.g. Average Precision)



# Examples of Complex Output Spaces



Will see examples later in lecture.

# Training Structured Prediction Models

- General Form:

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

- Or equivalently:

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

- Require continuous surrogate of evaluation measure.

# Structural SVM

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

*Sometimes normalize by M*

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i \quad \begin{matrix} \downarrow \\ \text{“Slack”} \end{matrix}$$
$$\forall i : \xi_i \geq 0$$

Consider:  $y' = \operatorname{argmax}_y F(y, x) \rightarrow F(y_i, x_i) - F(y', x_i) \leq 0$

Prediction of Learned Model

Slack is continuous upper bound on Hamming Loss!

$$y' \neq y_i \rightarrow \xi_i \geq \sum_j 1_{[y'^j \neq y_i^j]}$$
$$y' = y_i \rightarrow \xi_i \geq 0$$

# Example 1

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

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

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$\xi_i = 0$



$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss
(N,N)	2	2	1
(N,V)	4	0	0
(V,N)	1	3	2
(V,V)	1	3	1

# Example 2

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

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

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$\xi_i = 2$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss
(N,N)	4	-1	1
(N,V)	3	0	0
(V,N)	0	3	2
(V,V)	1	2	1

# Example 3

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

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

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$\xi_i = 1$  

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss
(N,N)	2	2	1
(N,V)	4	0	0
(V,N)	3	1	2
(V,V)	1	3	1

# When is Slack Positive?

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

$$\operatorname{argmin}_{w, \xi} \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_i \ell(y_i, x_i, F)$$

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

All  $i$ :  $\xi_i \geq 0$

Hamming Hinge Loss

- Whenever margin not big enough!

$$\xi_i > 0 \iff \exists y' : F(y_i, x_i) - F(y', x_i) < \sum_j \mathbb{1}_{[y'^j \neq y_i^j]}$$

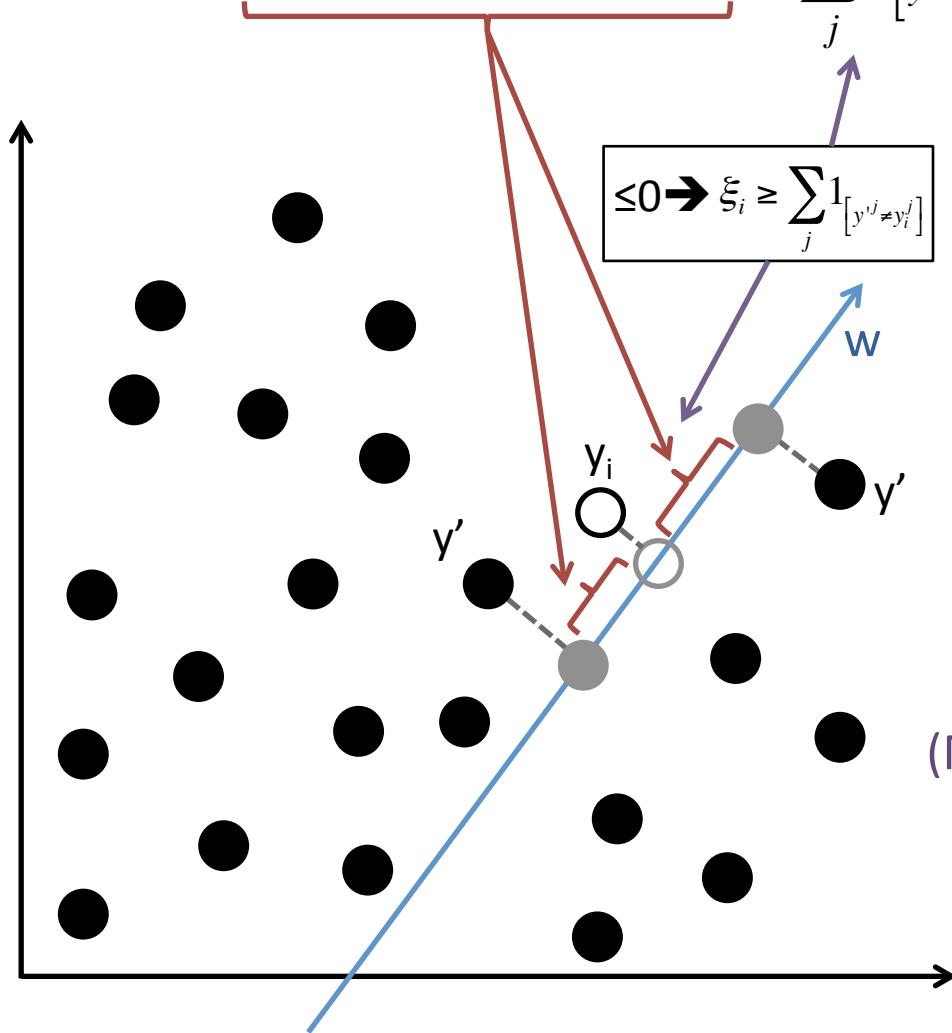
$$\xi_i = \max_{y'} \left\{ \sum_j \mathbb{1}_{[y'^j \neq y_i^j]} - (F(y_i, x_i) - F(y', x_i)) \right\} = \ell(y_i, x_i, F)$$

Verify that above definition  $\geq 0$

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

## Structural SVM Geometric Interpretation

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



Size of Margin  
vs  
Size of Margin Violations  
(C controls trade-off)  
(Margin scaled by Hamming Loss)

# Structural SVM Training

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

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

Often Exponentially Many!

- Strictly convex optimization problem
  - Same form as standard SVM optimization
  - Easy right?
- Intractable # of constraints!

# Structural SVM Training

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

- The trick is to not enumerate all constraints.
- Only solve the SVM objective over a small subset of constraints (**working set**).
  - Efficient!
- But some constraints might be violated.

# Example

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

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

$x_i = \text{"Fish Sleep"}$

$y_i = (N, V)$

$\xi_i = 0$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss
(N,N)	2	2	1
(N,V)	4	0	0
(V,N)	3	1	2
(V,V)	1	3	1



# Approximate Hinge Loss

- Choose tolerate  $\varepsilon > 0$ :

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

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

Consider:  $y' = \operatorname{argmax}_y F(y, x) \rightarrow F(y_i, x_i) - F(y', x_i) \leq 0$

Prediction of Learned Model

Slack is continuous upper bound on Hamming Loss -  $\varepsilon$ !

$$y' \neq y_i \rightarrow \xi_i \geq \sum_j 1_{[y'^j \neq y_i^j]} - \varepsilon$$
$$y' = y_i \rightarrow \xi_i \geq 0$$

# Example

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

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

$x_i = \text{"Fish Sleep"}$

$y_i = (N, V)$

$\xi_i = 0$

$\varepsilon = 1$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss
(N,N)	2	2	1
(N,V)	4	0	0
(V,N)	3	1	2
(V,V)	1	3	1



# Structural SVM Training

- **STEP 0:** Specify tolerance  $\varepsilon$
- **STEP 1:** Solve SVM objective function using only working set of constraints  $\mathbf{W}$  (initially empty). The trained model is  $w$ .
- **STEP 2:** Using  $w$ , find the  $y'$  whose constraint is most violated.
- **STEP 3:** If constraint is violated by more than  $\varepsilon$ , add it to  $\mathbf{W}$ .

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

- **Repeat STEP 1-3** until no additional constraints are added.  
Return most recent model  $w$  trained in STEP 1.

\*This is known as a “cutting plane” method.

# Example

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

Choose  $\varepsilon=0.1$

$$\forall i, y' \in W_i : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i - \varepsilon \quad \forall i : \xi_i \geq 0$$

**Init:**  $W_i = \emptyset$

**Solve:**  $\xi_i = 0$

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss	Viol.
(N,N)	0	0	1	1
(N,V)	0	0	0	0
(V,N)	0	0	2	2
(V,V)	0	0	1	1



**Constraint Violation:** Loss – Slack – (  $F(y, x) - F(y', x)$  ) = Viol

# Example

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

Choose  $\varepsilon=0.1$

$$\forall i, y' \in W_i : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i - \varepsilon \quad \forall i : \xi_i \geq 0$$

**Update:**  $W_i = \{(V, N)\}$

**Solve:**  $\xi_i = 0$

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss	Viol.
(N,N)	0	0	1	1
(N,V)	0	0	0	0
(V,N)	0	0	2	2
(V,V)	0	0	1	1



**Constraint Violation:** Loss – Slack – (  $F(y, x) - F(y', x)$  ) = Viol

# Example

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

**Choose  $\varepsilon=0.1$**

$$\forall i, y' \in W_i : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i - \varepsilon \quad \forall i : \xi_i \geq 0$$

**Update:**  $W_i = \{(V, N)\}$

**Solve:**  $\xi_i = 0.5$

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss	Viol.
(N,N)	0.7	0.2	1	0.2
(N,V)	0.9	0	0	0
(V,N)	-0.6	1.5	2	0
(V,V)	0	0.9	1	0.4



**Constraint Violation:** Loss – Slack – (  $F(y, x) - F(y', x)$  ) = Viol

# Example

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

**Choose  $\varepsilon=0.1$**

$$\forall i, y' \in W_i : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i - \varepsilon \quad \forall i : \xi_i \geq 0$$

**Update:**  $W_i = \{(V,N), (N,N)\}$

**Solve:**  $\xi_i = 0.5$

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss	Viol.	
(N,N)	0.7	0.2	1	0.2	✗
(N,V)	0.9	0	0	0	✓
(V,N)	-0.6	1.5	2	0	✓
(V,V)	0	0.9	1	0.4	✗

**Constraint Violation:** Loss – Slack – (  $F(y, x) - F(y', x)$  ) = Viol

# Example

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

**Choose  $\varepsilon=0.1$**

$$\forall i, y' \in W_i : F(y_i, x_i) - F(y', x_i) \geq \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i - \varepsilon \quad \forall i : \xi_i \geq 0$$

**Update:**  $W_i = \{(V,N), (N,N)\}$

**Solve:**  $\xi_i = 0.55$

$x_i$  = “Fish Sleep”

$y_i = (N, V)$

$y'$	$F(y', x_i)$	$F(y_i, x_i) - F(y', x_i)$	Loss	Viol.
(N,N)	0.55	0.45	1	0
(N,V)	1	0	0	0
(V,N)	-0.65	1.65	2	0
(V,V)	-0.05	0.95	1	0.05



**Constraint Violation:** Loss – Slack – (  $F(y, x) - F(y', x)$  ) = Viol

# Example

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

Choose  $\varepsilon=0.1$

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

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

No constraint is violated  
by more than  $\varepsilon$

Solve:  $\xi_i = 0.55$

i	Loss	Viol.
1	1	0
2	0	0
3	2	0
(V,V)	-0.05	0.95
	1	0.05

Constraint Violation:  $\text{Loss} - \text{Slack} - (F(y, x) - F(y', x)) = \text{Viol}$

# Geometric Interpretation

Scoring  $y$  corresponds to dot product of high dimensional point.

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

High Dimensional Point

$$\Psi_{i,y'} = \sum_{j=1}^M [\varphi^j(y_i^j, y_i^{j-1} | x_i) - \varphi(y'^j, y'^{j-1} | x_i)]$$

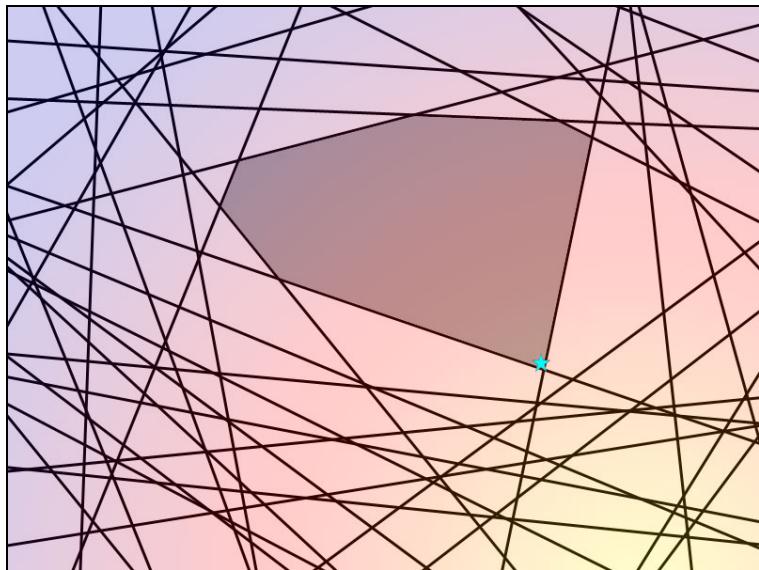
$$\underset{w, \xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

Quad. Optimization Problem  
w/ Linear Constraints!

$$\forall i, y': w^T \Psi_{i,y'} \geq \Delta_{i,y'} - \xi_i$$

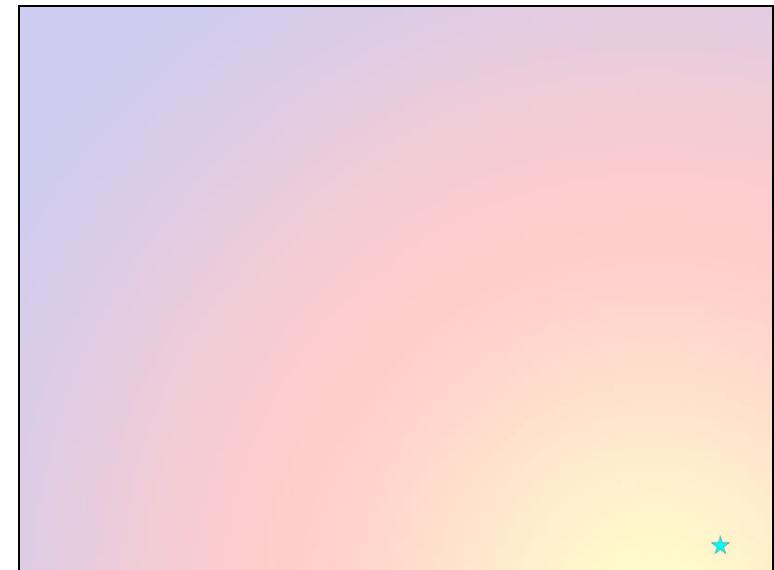
$$\forall i : \xi_i \geq 0$$

# Geometric Example



## Naïve SVM Problem

- Exponential constraints
- Most are dominated by a small set of “important” constraints

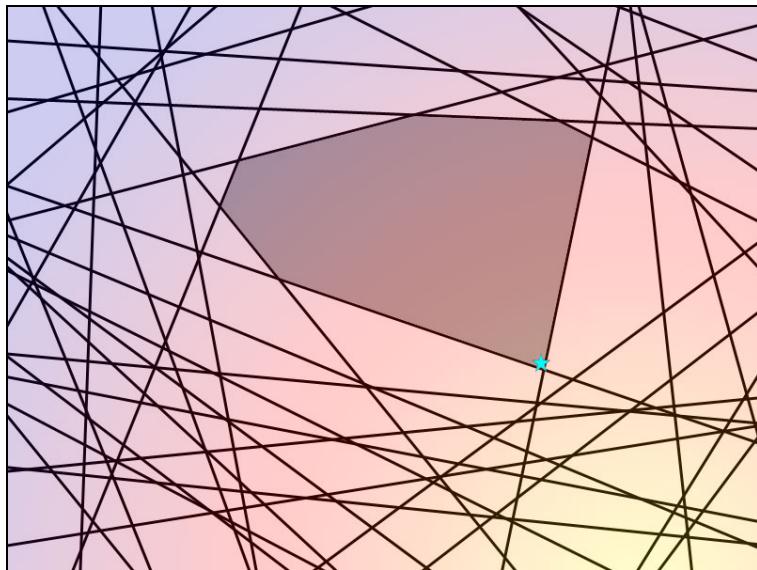


## Structural SVM Approach

- Repeatedly finds the next most violated constraint...
- ...until set of constraints is a good approximation.

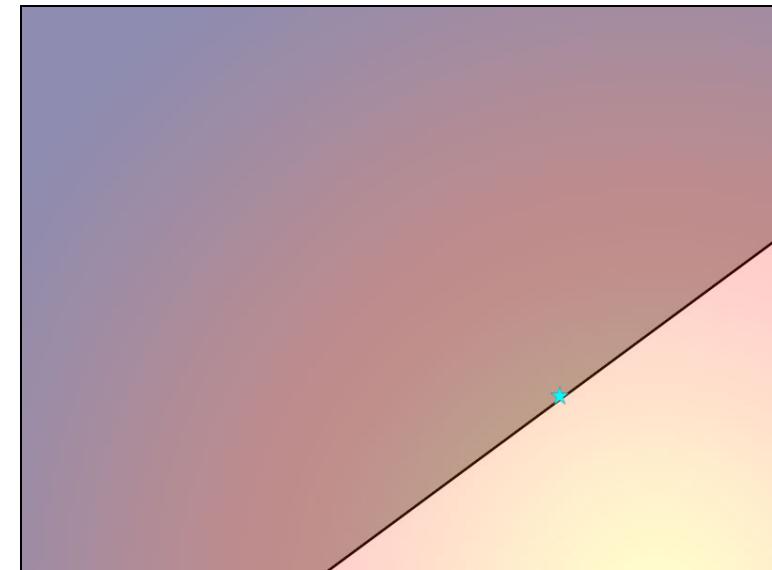
\*This is known as a “cutting plane” method.

# Geometric Example



## Naïve SVM Problem

- Exponential constraints
- Most are dominated by a small set of “important” constraints

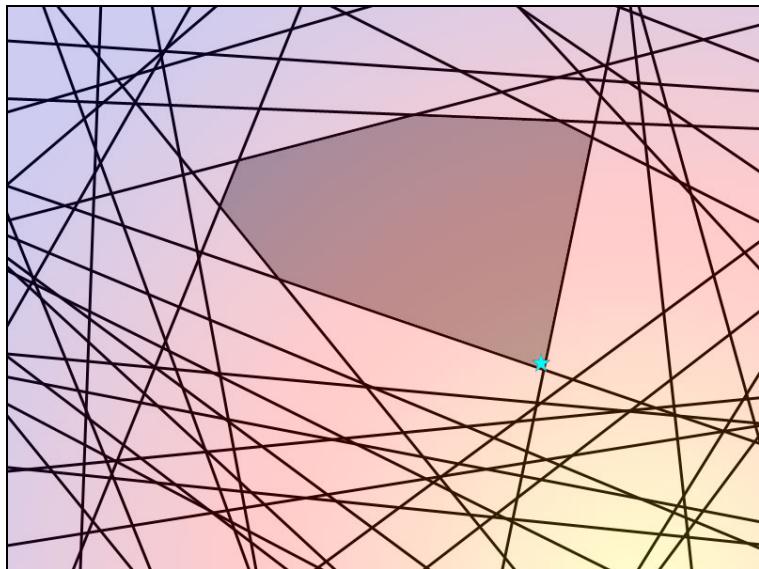


## Structural SVM Approach

- Repeatedly finds the next most violated constraint...
- ...until set of constraints is a good approximation.

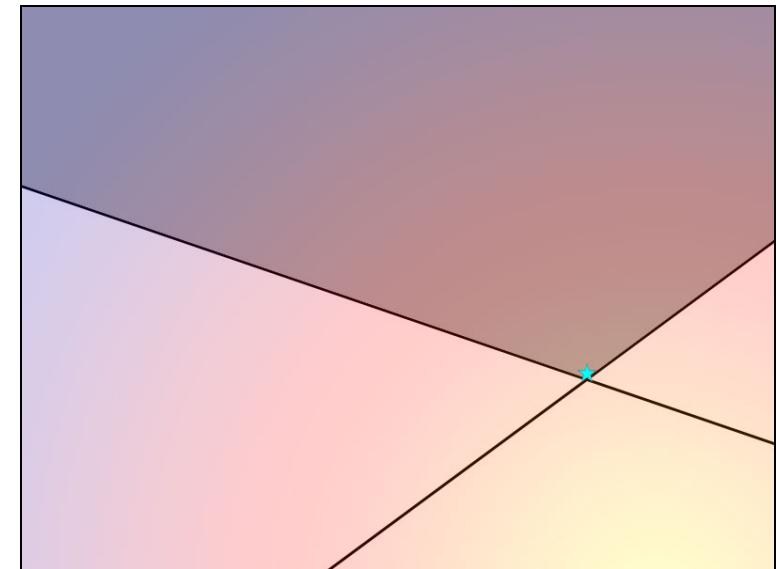
\*This is known as a “cutting plane” method.

# Geometric Example



## Naïve SVM Problem

- Exponential constraints
- Most are dominated by a small set of “important” constraints

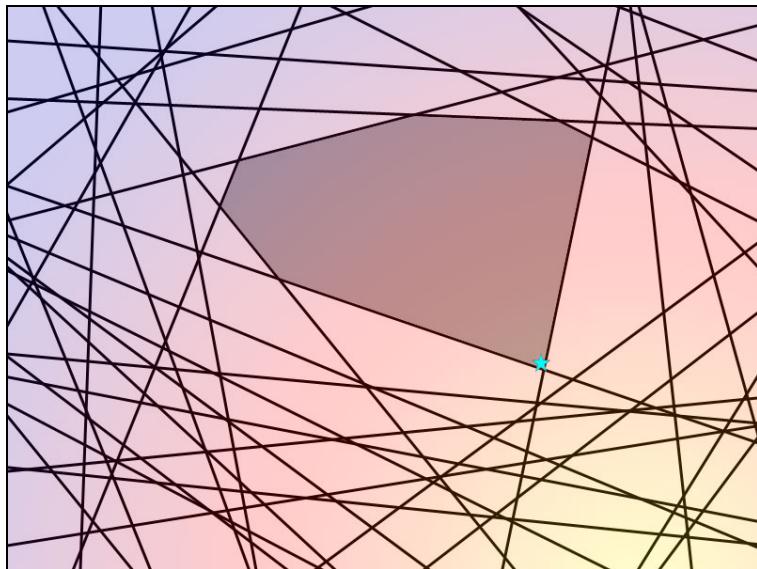


## Structural SVM Approach

- Repeatedly finds the next most violated constraint...
- ...until set of constraints is a good approximation.

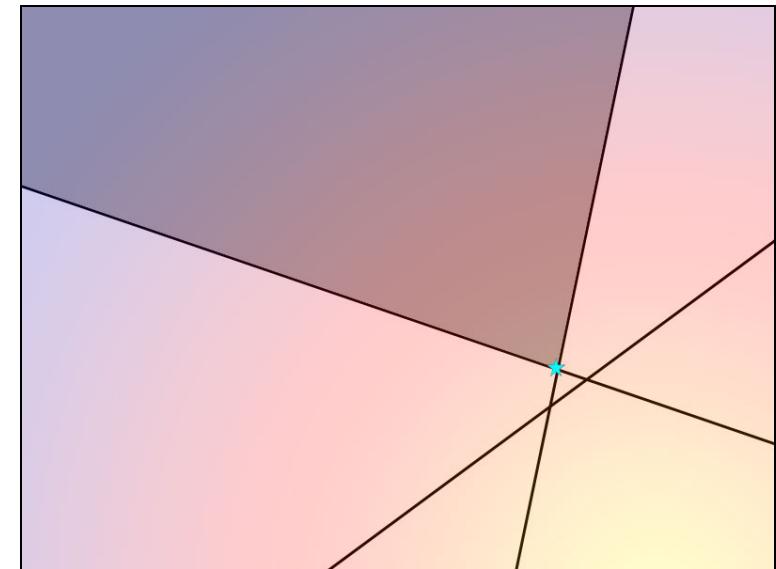
\*This is known as a “cutting plane” method.

# Geometric Example



## Naïve SVM Problem

- Exponential constraints
- Most are dominated by a small set of “important” constraints



## Structural SVM Approach

- Repeatedly finds the next most violated constraint...
- ...until set of constraints is a good approximation.

\*This is known as a “cutting plane” method.

# Linear Convergence Rate

- Guarantee for any  $\varepsilon > 0$ :

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

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

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

- Terminates after #iterations:  $O\left(\frac{1}{\varepsilon}\right)$

Proof found in:

[http://www.cs.cornell.edu/people/tj/publications/joachims\\_etal\\_09a.pdf](http://www.cs.cornell.edu/people/tj/publications/joachims_etal_09a.pdf)

# Finding Most Violated Constraint

- A constraint is violated when:

$$F(y', x_i) - F(y_i, x_i) + \sum_j 1_{[y'^j \neq y_i^j]} - \xi_i > 0$$

- Finding most violated constraint reduces to

$$\operatorname{argmax}_{y'} F(y', x_i) + \sum_j 1_{[y'^j \neq y_i^j]}$$

“Loss augmented inference”

- Highly related to prediction:

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

# “Augmented” Scoring Function

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

**Goal:**

$$\operatorname{argmax}_{y'} F(y', x_i) + \sum_j 1_{[y'^j \neq y_i^j]}$$

Solve Using Viterbi!

$$\tilde{F}(y, x_i, y_i) = \sum_{j=1}^M [\tilde{w}^T \tilde{\varphi}^j(y^j, y^{j-1} | x_i, y_i)]$$

$$\tilde{\varphi}^j(a, b | x_i, y_i) = \begin{bmatrix} \varphi^j(a, b | x_i) \\ 1_{[a \neq y_i^j]} \end{bmatrix}$$

Additional  
Unary Feature!

$$\tilde{w} = \begin{bmatrix} w \\ 1 \end{bmatrix}$$

**Goal:**  $\operatorname{argmax}_{y'} \tilde{F}(y', x_i, y_i)$

# Recap: Structural SVM

- Define structured scoring function:  $F(y, x)$ 
  - E.g., 1<sup>st</sup> order sequential model
  - Efficient prediction algorithm
- Define error function:  $\Delta(y, y')$ 
  - E.g., Hamming Loss:  $\Delta(y, y') = \sum_j 1_{[y'^j \neq y^j]}$
- Train by iteratively finding most violated constraint:  
$$\operatorname{argmax}_{y'} F(y', x_i) + \Delta(y_i, y')$$
  - Requires efficient algorithm (often same as prediction)

# Structural SVMs vs CRFs

- **SVM Objective:**

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

$$\forall i, y' : F(y_i, x_i) - F(y', x_i) \geq \Delta(y_i, y') - \xi_i \quad \forall i : \xi_i \geq 0$$

SVM only cares about  $y'$  that violates margin the most!

- Scales margin by loss of  $y'$

- **CRF Objective:**

$$\operatorname{argmin}_w \frac{1}{2} w^T w + \frac{C}{N} \sum_i -\log P(y_i | x_i)$$

CRF cares about all  $y'$  so that:

- Incorrect  $P(y' | x)$  is minimized
- Correct  $P(y | x)$  is maximized

$$-\log P(y_i | x_i) = F(y_i, x_i) - \log \left( \sum_{y'} \exp \{F(y', x_i)\} \right)$$

# General Structured Prediction

# More Elaborate Scoring Functions

- Structure encoded by linear scoring function:

$$F(y, x)$$

- 2<sup>nd</sup> 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:

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

# More Elaborate Scoring Functions

- Structure encoded by linear scoring function:

$$F(y, x)$$

Remainder of Lecture:

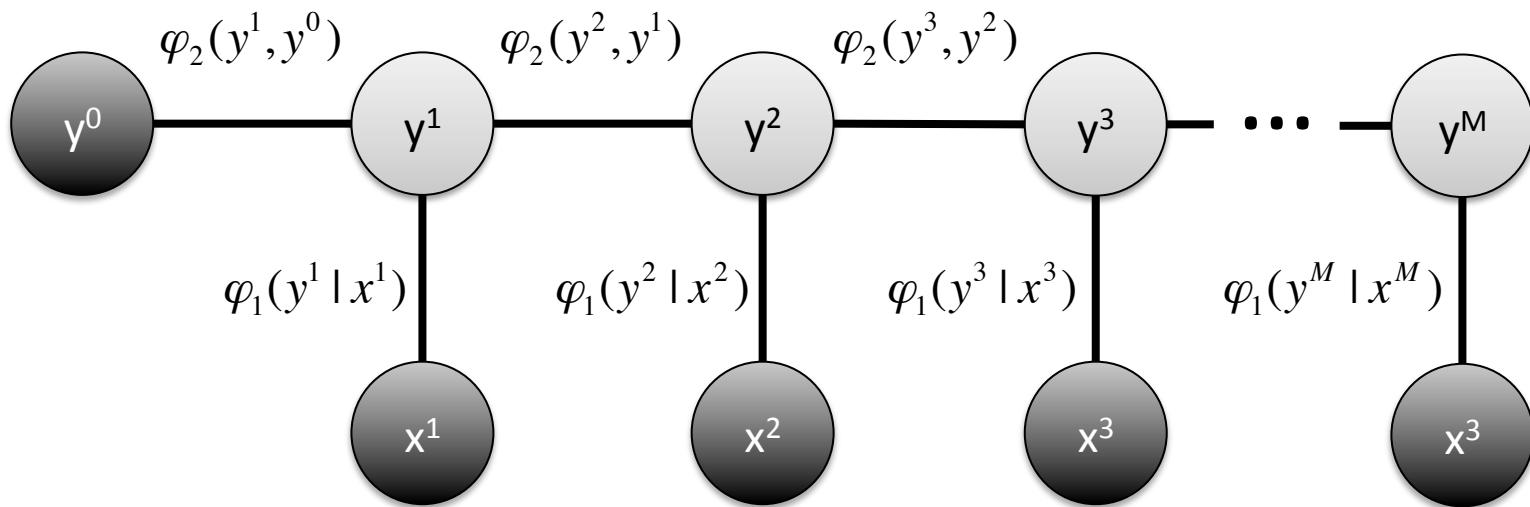
Tour of Structured Prediction Models  
Some Might be Interesting to You...

- Efficient Prediction:

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

# Graphical Models

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



**Graph structure encodes structural dependencies between  $y^j$ !**

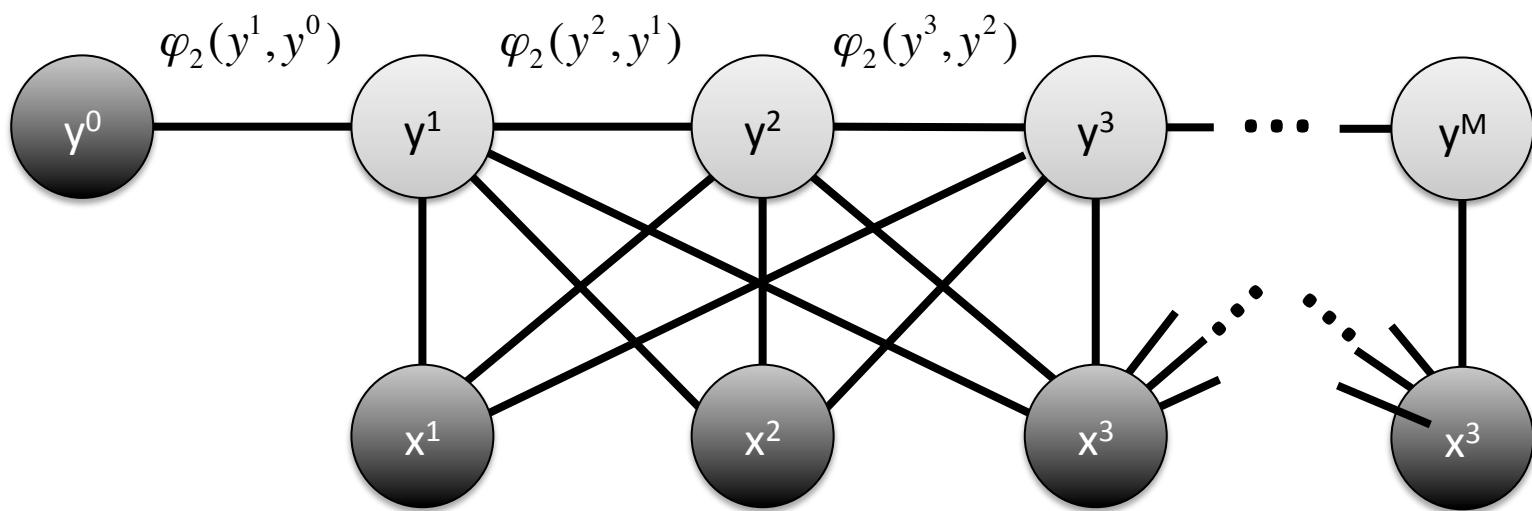
<https://piazza.com/cornell/fall2013/btry6790cs6782/resources>

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

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

# Graphical Models

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



**Graph structure encodes structural dependencies between  $y^j$ !**

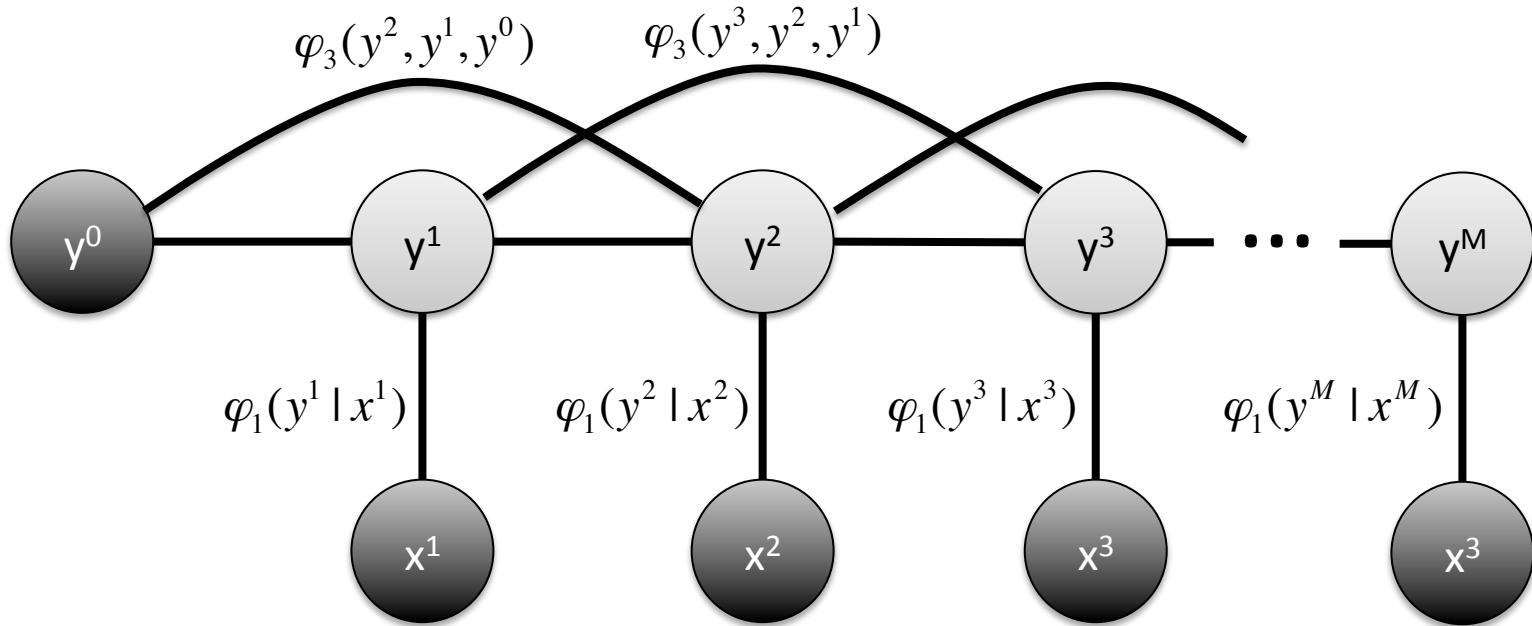
<https://piazza.com/cornell/fall2013/btry6790cs6782/resources>

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

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

# Graphical Models

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



**Graph structure encodes structural dependencies between  $y^i$ !**

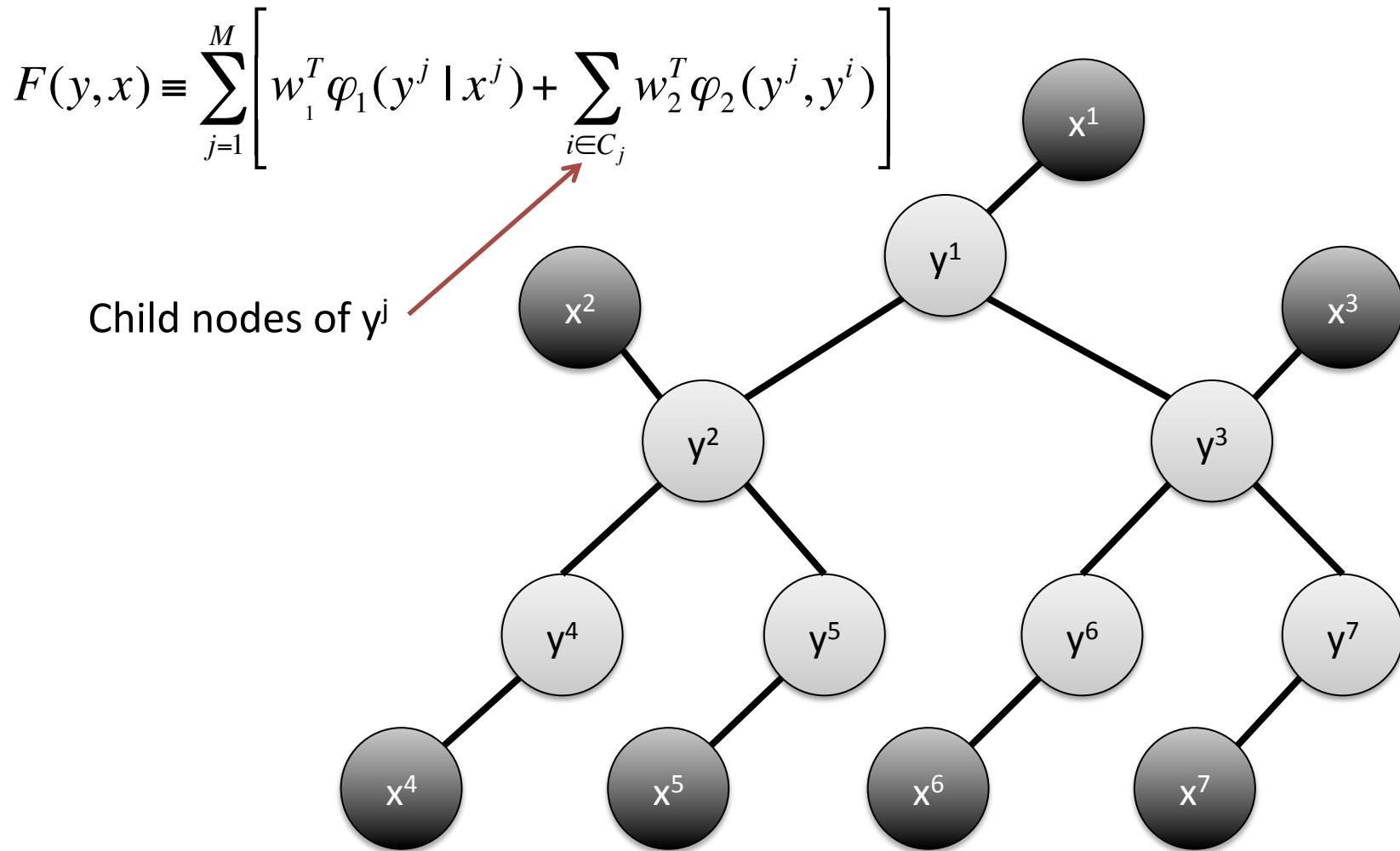
**Features depend on cliques in graphical model representation.**

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

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

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

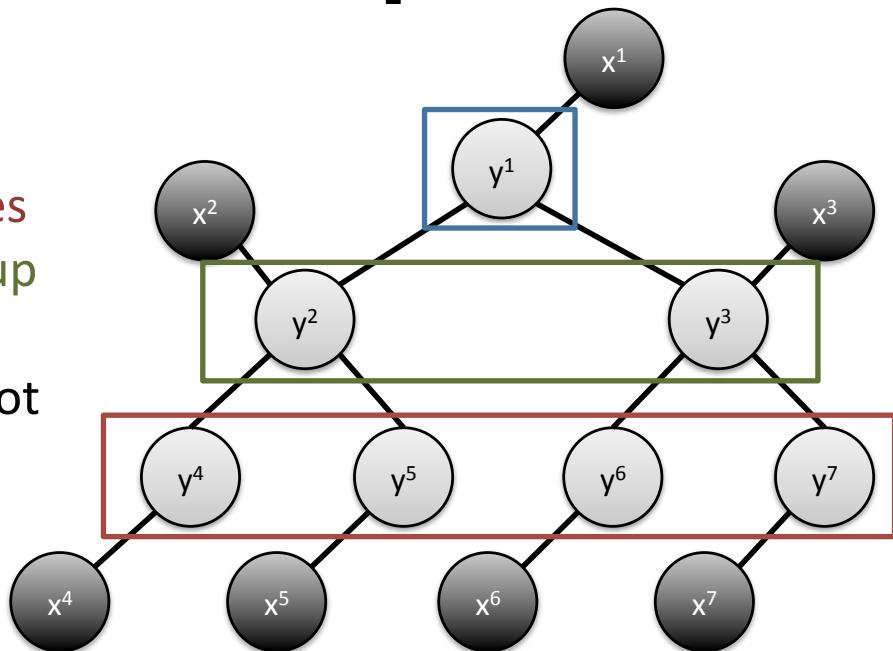
# Tree Structured Models



# Prediction via Dynamic Programming

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

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

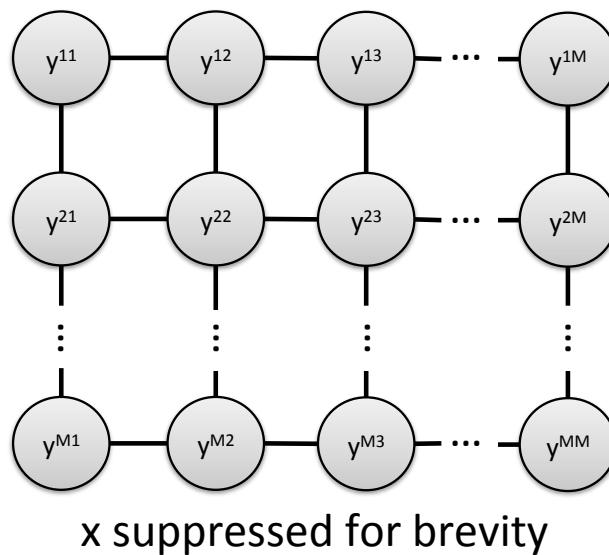
# Loopy Graphical Models

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!

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

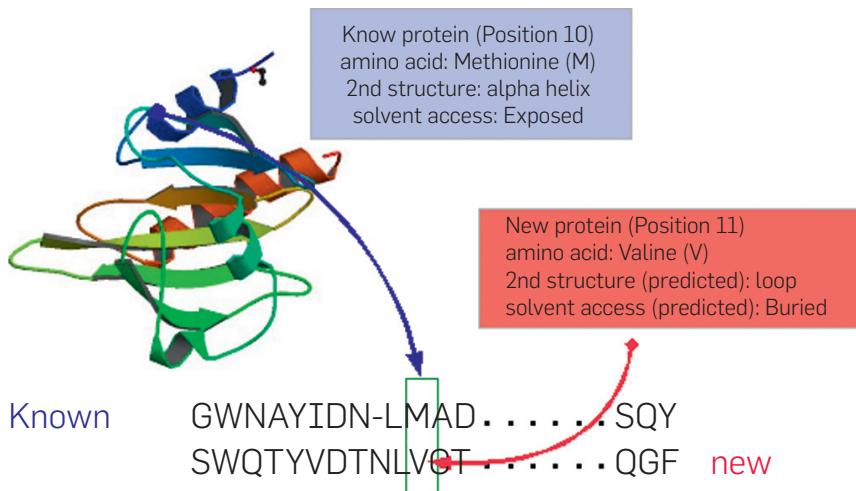


<http://vision.middlebury.edu/MRF/>

<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**)

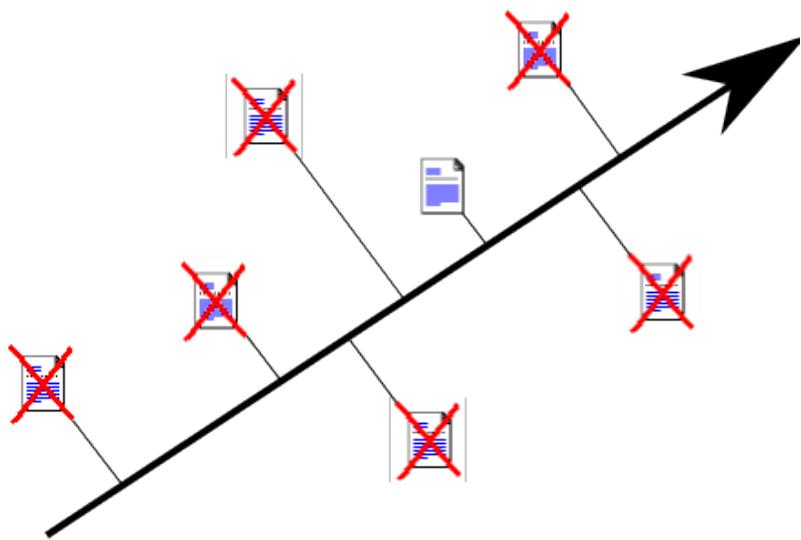
Train on **G**: learn how to align any amino acid seq to proteins in **D**

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

[http://www.cs.cornell.edu/People/tj/publications/yu\\_etal\\_06a.pdf](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



x = query & set of results

y = ranking

Find w that predicts best ranking of search results.

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

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

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

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

[http://www.cs.cornell.edu/People/tj/publications/joachims\\_05a.pdf](http://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://research.microsoft.com/en-us/um/people/cburges/tech_reports/MSR-TR-2010-82.pdf)

[http://www.yisongyue.com/publications/sigir2007\\_svmmapper.pdf](http://www.yisongyue.com/publications/sigir2007_svmmapper.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/spflood/>

[http://www.cs.cornell.edu/People/tj/publications/joachims\\_06b.pdf](http://www.cs.cornell.edu/People/tj/publications/joachims_06b.pdf)

# Next Week

- Decision Trees
- Bagging
- Random Forests
- Boosting
- Ensemble Selection
- Often the most accurate methods in practice.
  - (Hint: try them for the Kaggle mini-project)