# Advanced Machine Learning & Perception

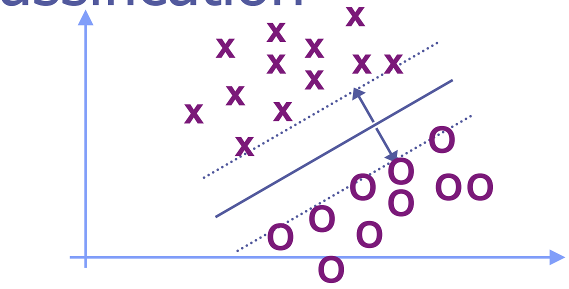
Instructor: Tony Jebara

# Topic 8

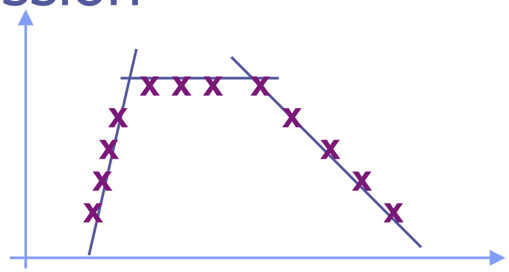
- Beyond binary output…
- Based on T. Joachims’ slides
- Multi-Class SVM
- Structured Prediction
- Cutting Plane Algorithms

# SVM Extensions

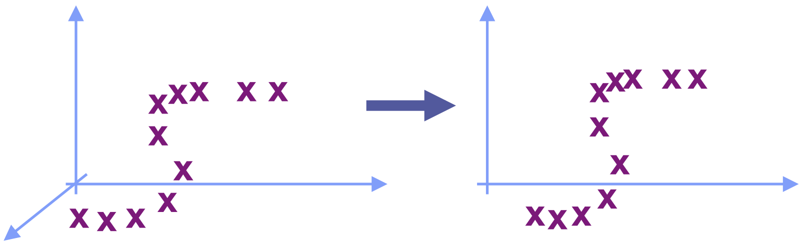
## Classification



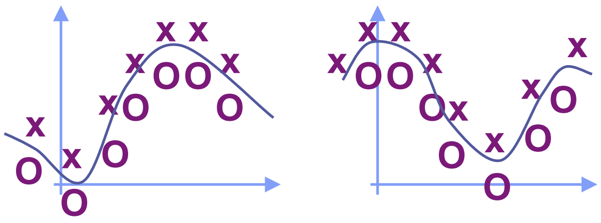
## Regression



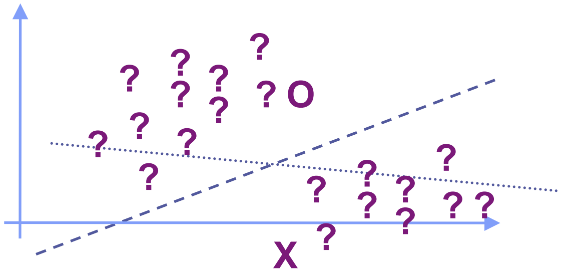
## Feature/Kernel Selection



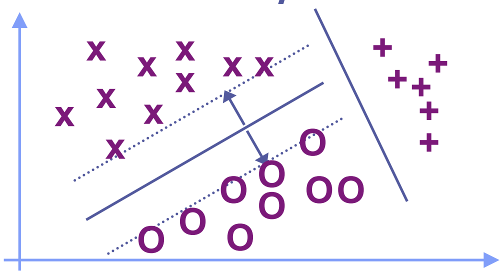
## Meta/Multi-Task Learning



## Transduction



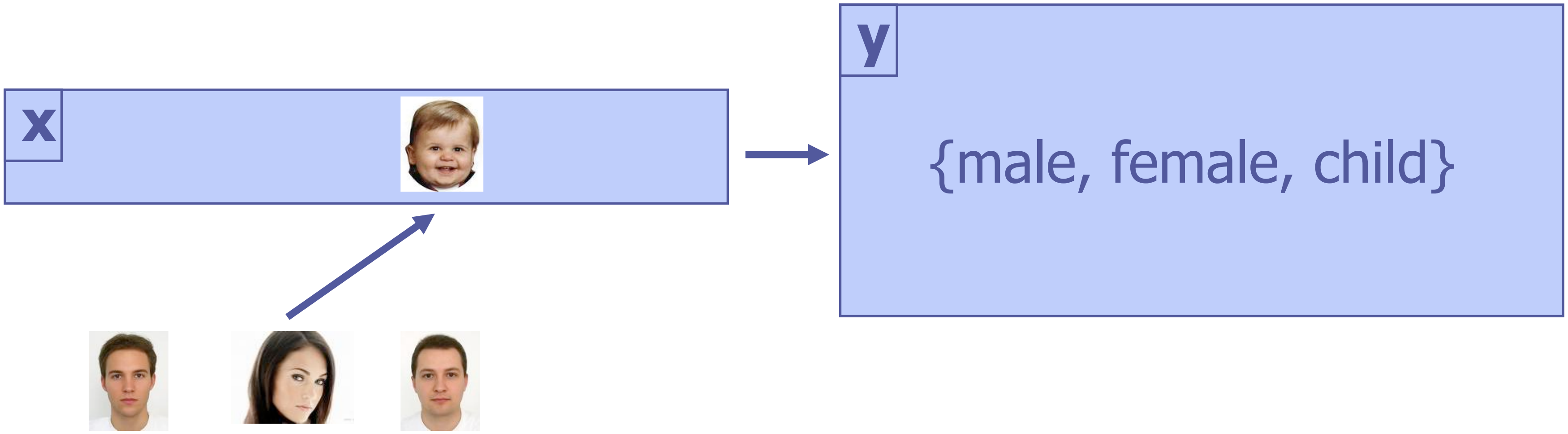
## Multi-Class / Structured



# Multi-Class & Structured Output

- Support vector machines predict only a binary output

- Can SVMs handle multi-class labels??

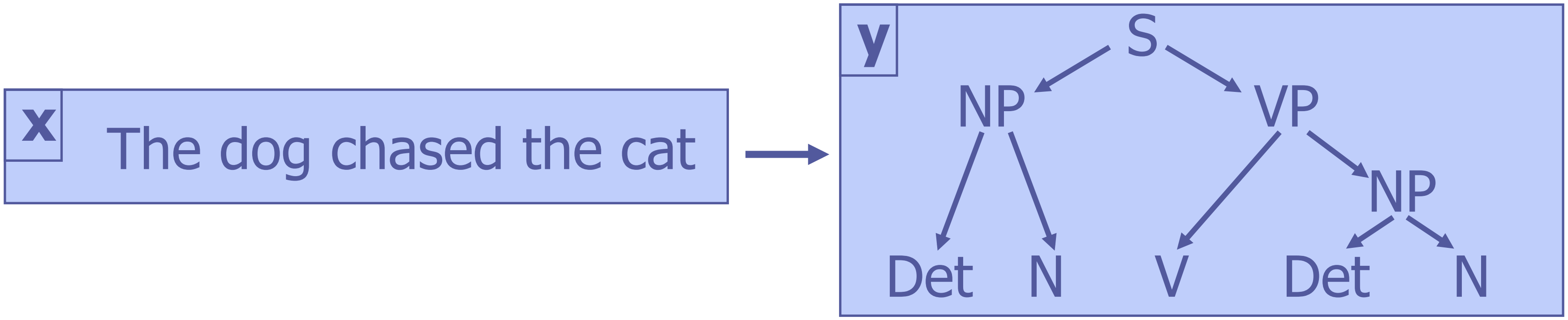


{male, female, child}

# Multi-Class & Structured Output

- Or, (almost any) structured output?
- For example: 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.

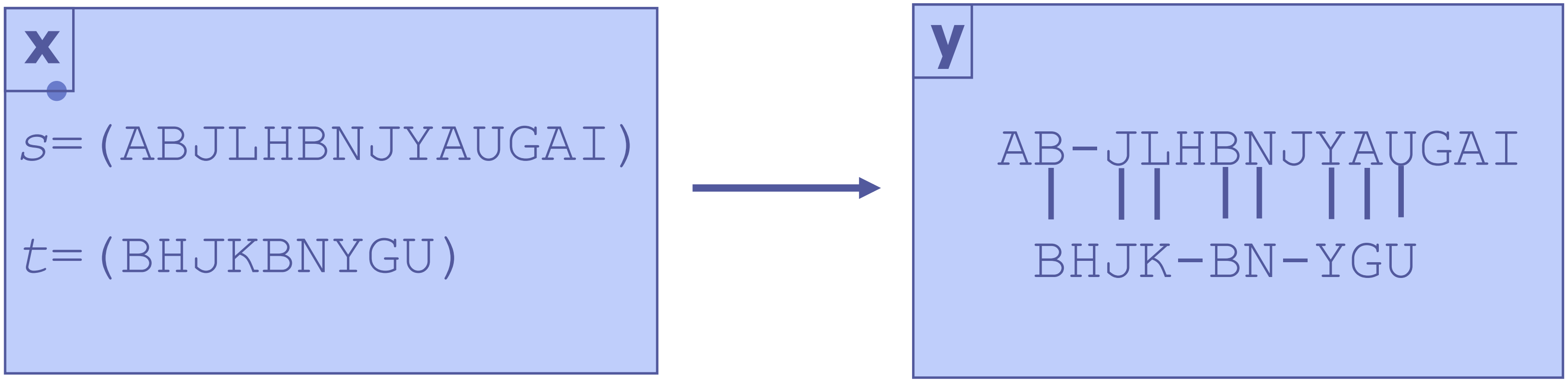


# Multi-Class & Structured Output

- Or, (almost any) structured output?

For example: Protein Sequence Alignment

> Given two sequences *x=(s,t)*, predict an alignment *y*.
>
> Structural dependencies, since prediction has to be a valid global/local alignment.



*s*= (ABJLHBNJYAUGAI)

*t*= (BHJKBNYGU)

```
AB-JLHBNJYAUGAI
 | || || |||
 BHJK-BN-YGU
```

# Multi-Class & Structured Output

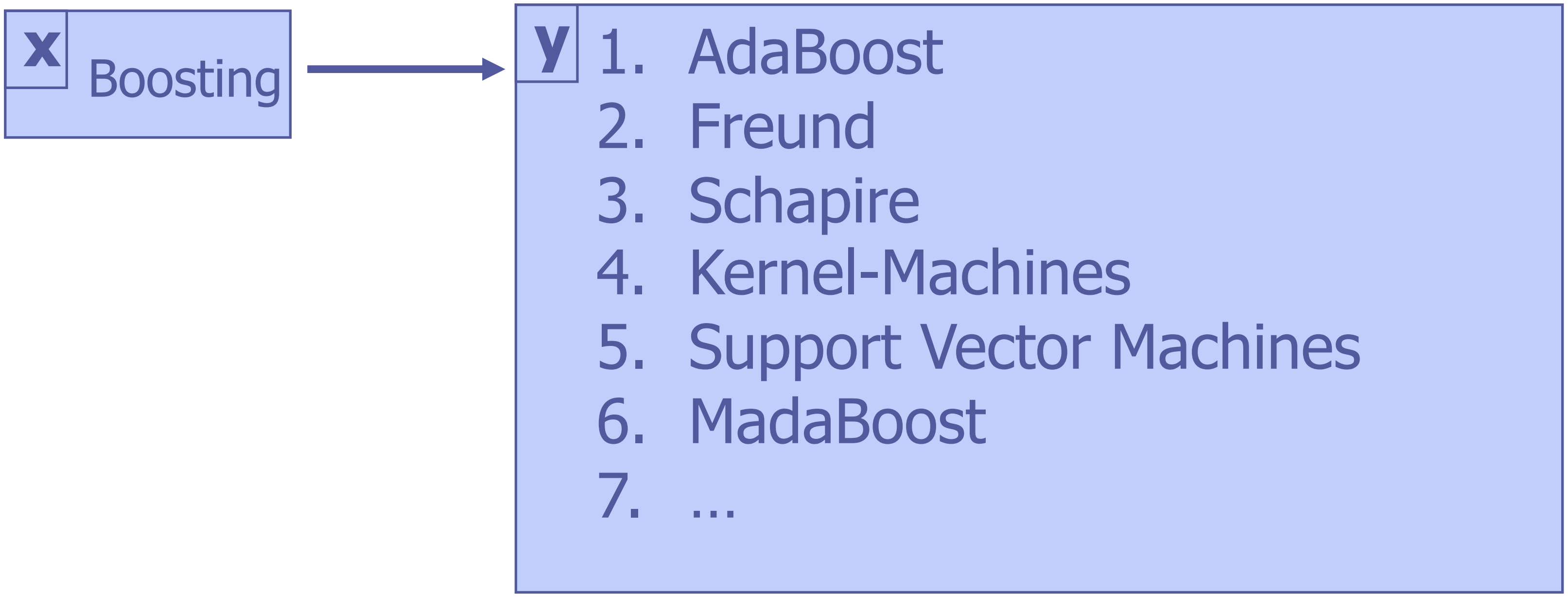
•Or, (almost any) structured output?

For example: Information Retrieval

Given a query x, predict a ranking *y*.

Dependencies between results (e.g. avoid redundant hits)

Loss function over rankings (e.g. AvgPrec)



**x** Boosting

**y**

1. AdaBoost
2. Freund
3. Schapire
4. Kernel-Machines
5. Support Vector Machines
6. MadaBoost
7. …

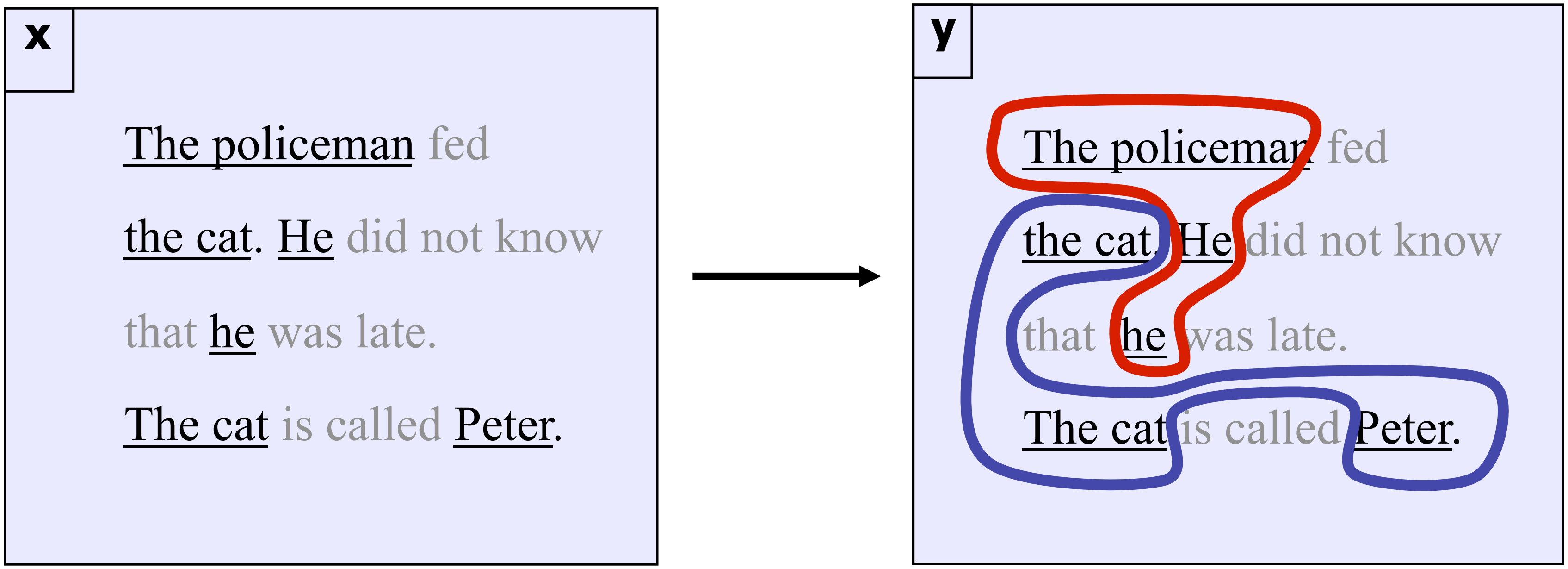
# Multi-Class & Structured Output

- Or, (almost any) structured output?
- For Example, Noun-Phrase Co-reference

  Given a set of noun phrases *x*, predict a clustering *y*.

  Structural dependencies, since prediction has to be an equivalence relation.

  Correlation dependencies from interactions.

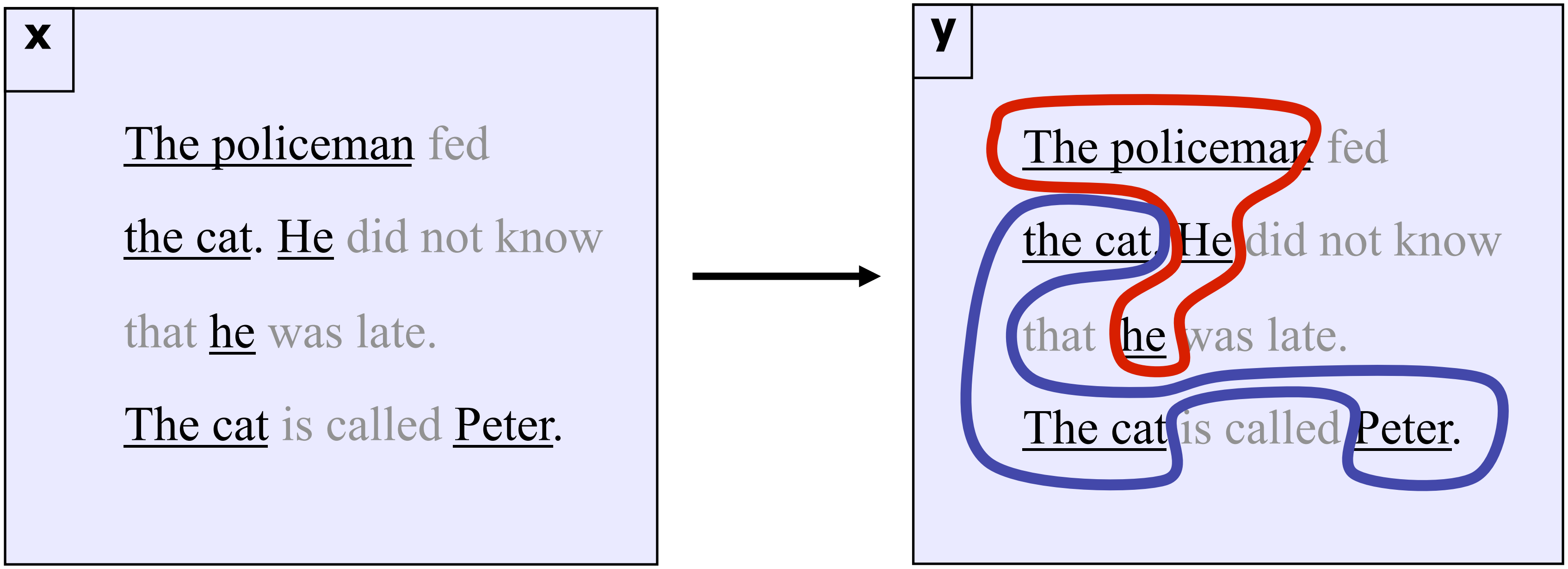


The policeman fed the cat. He did not know that he was late. The cat is called Peter.

The policeman fed the cat. He did not know that he was late. The cat is called Peter.

# Multi-Class & Structured Output

- These problems are usually solved via maximum likelihood
- Or via Bayesian Networks and Graphical Models
- Problem: these methods are not discriminative!
- They learn p(x,y) instead of x→y like an SVM…
- We will adapt the SVM approach to these domains…

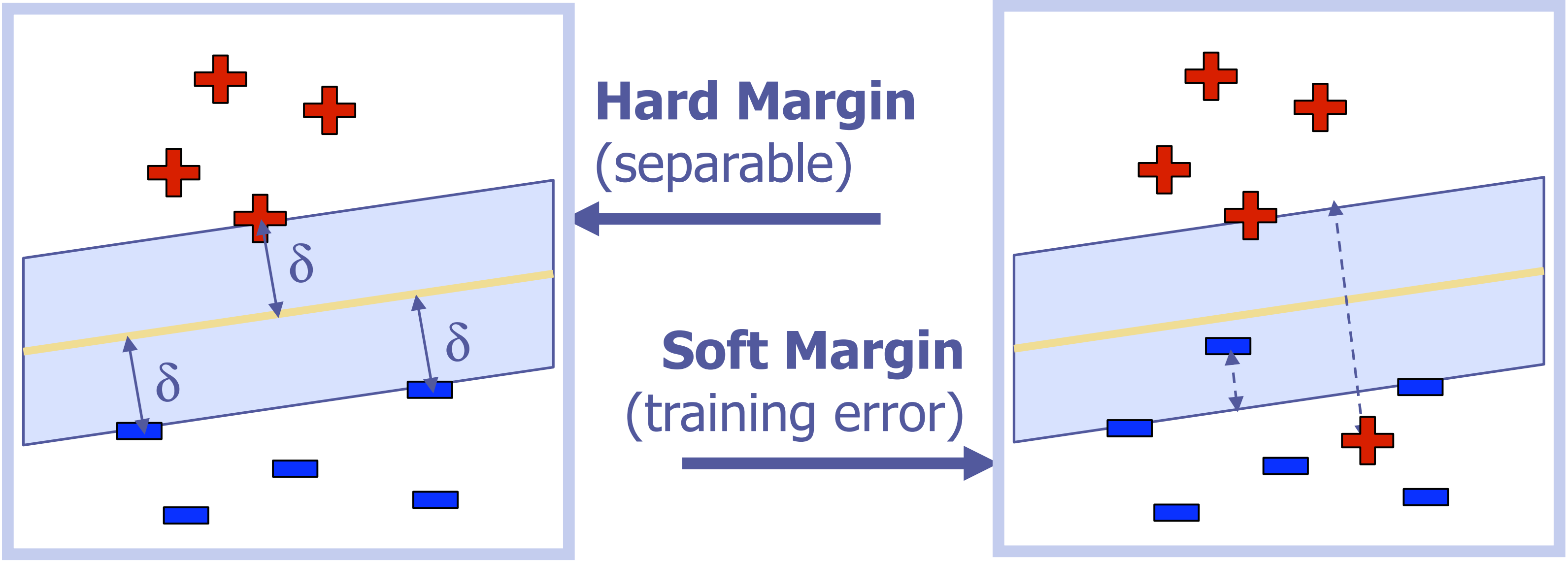


The policeman fed
the cat. He did not know
that he was late.
The cat is called Peter.

The policeman fed
the cat. He did not know
that he was late.
The cat is called Peter.

# Support Vector Machine

- Binary classification: $\left\{\left(\mathbf{x}_1, y_1\right), \ldots, \left(\mathbf{x}_n, y_n\right)\right\} \rightarrow f\left(\mathbf{x}\right) = \mathbf{w}^T\mathbf{x} + b$



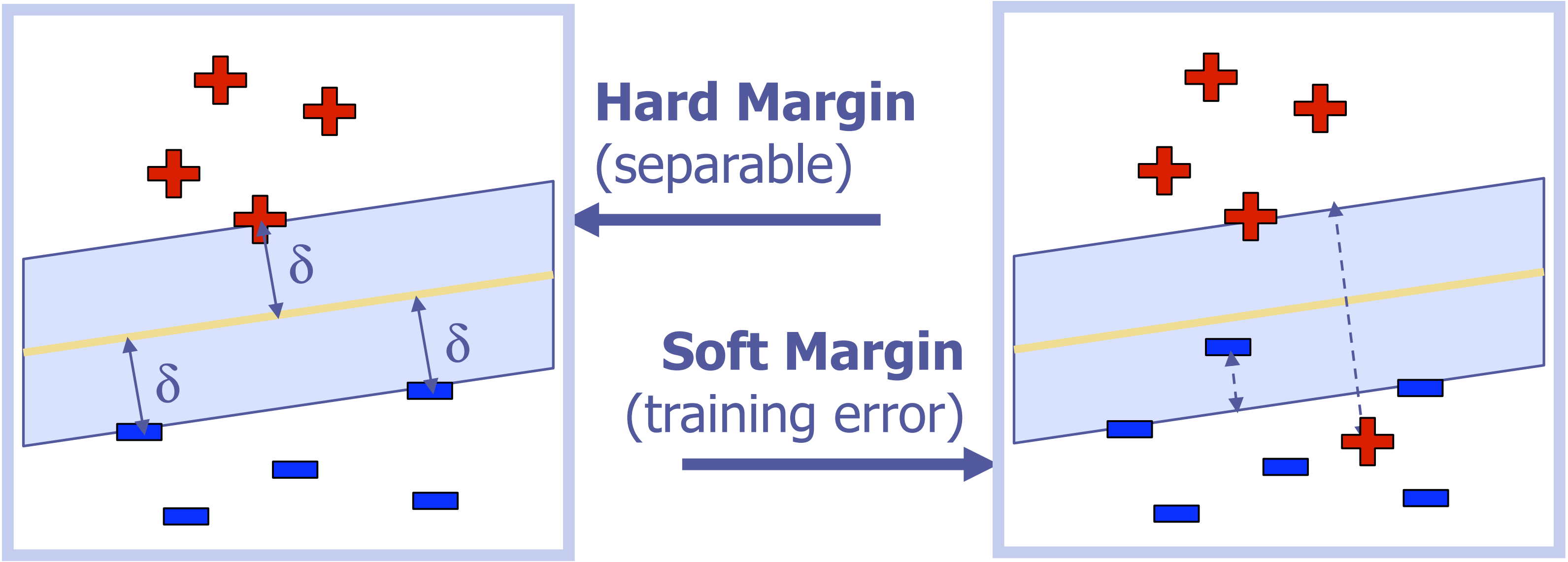
**Hard Margin**
(separable)

**Soft Margin**
(training error)

- P: $\min_{w,b,\xi\geq 0} \frac{1}{2}\left\|\mathbf{w}\right\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \quad s.t. \quad y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i$
- D: $\max_{\lambda} \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{n}\lambda_i\lambda_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \;\; s.t. 0 \leq \lambda_i \leq \frac{C}{n}, \sum_{i=1}^{n}\lambda_i y_i = 0$
- Primal (P) and dual (D) give same solution $\mathbf{w}^* = \sum_{i=1}^{n}\lambda_i^* y_i \mathbf{x}_i$

# Support Vector Machine & b=0

- Binary classification: $\left\{\left(\mathbf{x}_1, y_1\right), \ldots, \left(\mathbf{x}_n, y_n\right)\right\} \rightarrow f\left(\mathbf{x}\right) = \mathbf{w}^T\mathbf{x}$



**Hard Margin** (separable)

**Soft Margin** (training error)

- P: $\min_{w,b,\xi\geq 0} \frac{1}{2}\left\|\mathbf{w}\right\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \quad s.t. \quad y_i\left(\mathbf{w}^T\mathbf{x}_i\right) \geq 1 - \xi_i$

- D: $\max_\lambda \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{n}\lambda_i\lambda_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \;\; s.t. 0 \leq \lambda_i \leq \frac{C}{n}$

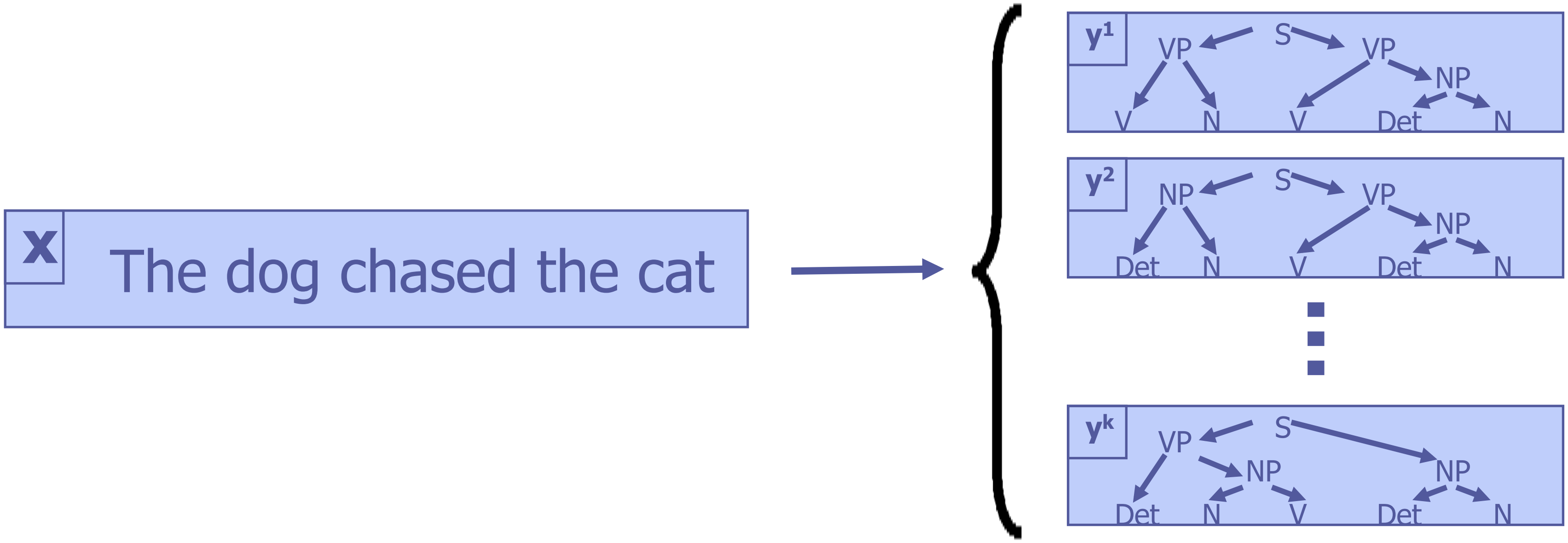
- Solution through origin $\mathbf{w}^* = \sum_{i=1}^{n}\lambda_i^* y_i \mathbf{x}_i$ (or just pad x with 1)

# Multi-Class & Structured Output

- View the problem as a list of all possible answers
- Approach: view as multi-class classification task
- Every complex output $y_i \in Y$ is one class
- Problems: Exponentially many classes!

  How to predict efficiently? How to learn efficiently?

  Potentially huge model! Manageable number of features?

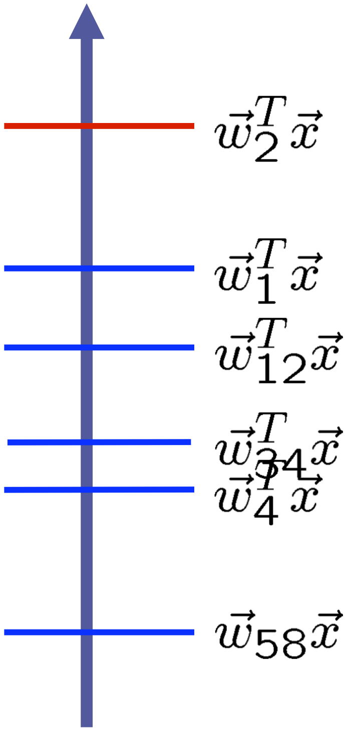


x The dog chased the cat

# Multi-Class Output

- View the problem as a list of all possible answers
- Approach: view as multi-class classification task
- Every complex output $y_i \in \{1,\ldots,k\}$ is one of K classes
- Enumerate many constraints (slow)…

$$\{(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_n, y_n)\} \to f(\mathbf{x}) = \arg\max_{i\in\{1,\ldots,k\}} \mathbf{w}_i^T \mathbf{x}$$



$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\xi\geq 0} \sum_{i=1}^{k} \|\mathbf{w}_i\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad \forall j \neq y_1 : (\mathbf{w}_{y_1}^T \mathbf{x}_1) \geq (\mathbf{w}_j^T \mathbf{x}_1) + 1 - \xi_1$$

$$s.t. \quad \ldots$$

$$s.t. \quad \forall j \neq y_n : (\mathbf{w}_{y_n}^T \mathbf{x}_n) \geq (\mathbf{w}_j^T \mathbf{x}_n) + 1 - \xi_n$$

# Joint Feature Map

- Instead of solving for K different w's, make 1 long w
- Replace each x with $\phi\left(\mathbf{x}, y = i\right) = \left[0^T\ 0^T \ldots 0^T\ \mathbf{x}^T\ 0^T \ldots 0^T\ \right]^T$
- Put the x vector in the i'th position
- The feature vectors is DK dimensional

$$y_i \in \left\{1, \ldots, k\right\}$$

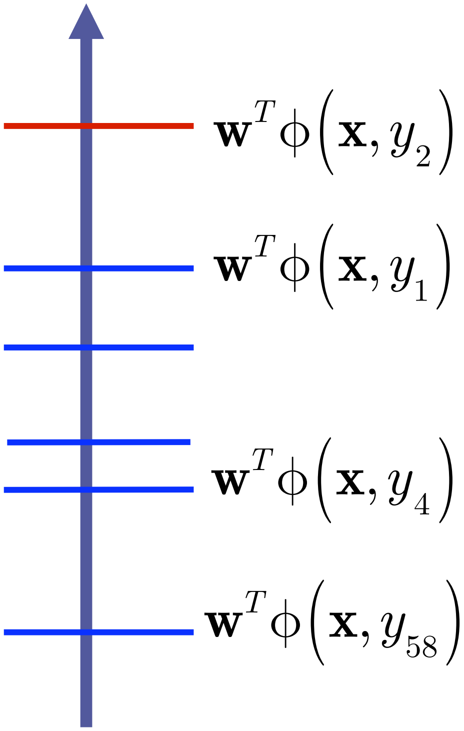
$$\left\{\left(\mathbf{x}_1, y_1\right), \ldots, \left(\mathbf{x}_n, y_n\right)\right\} \rightarrow f\left(\mathbf{x}\right) = \arg\max_{y \in Y} \mathbf{w}^T \phi\left(\mathbf{x}, y\right)$$

$$\min_{\mathbf{w}, \xi \geq 0} \left\|\mathbf{w}\right\|^2$$

$$s.t. \quad \forall y \cup Y \setminus y_1 : \mathbf{w}^T \phi\left(\mathbf{x}_1, y_1\right) \geq \mathbf{w}^T \phi\left(\mathbf{x}_1, y\right) + 1$$

$$s.t. \quad \ldots$$

$$s.t. \quad \forall y \cup Y \setminus y_n : \mathbf{w}^T \phi\left(\mathbf{x}_n, y_n\right) \geq \mathbf{w}^T \phi\left(\mathbf{x}_n, y\right) + 1$$



$\mathbf{w}^T \phi\left(\mathbf{x}, y_2\right)$
$\mathbf{w}^T \phi\left(\mathbf{x}, y_1\right)$
$\mathbf{w}^T \phi\left(\mathbf{x}, y_4\right)$
$\mathbf{w}^T \phi\left(\mathbf{x}, y_{58}\right)$

# Joint Feature Map

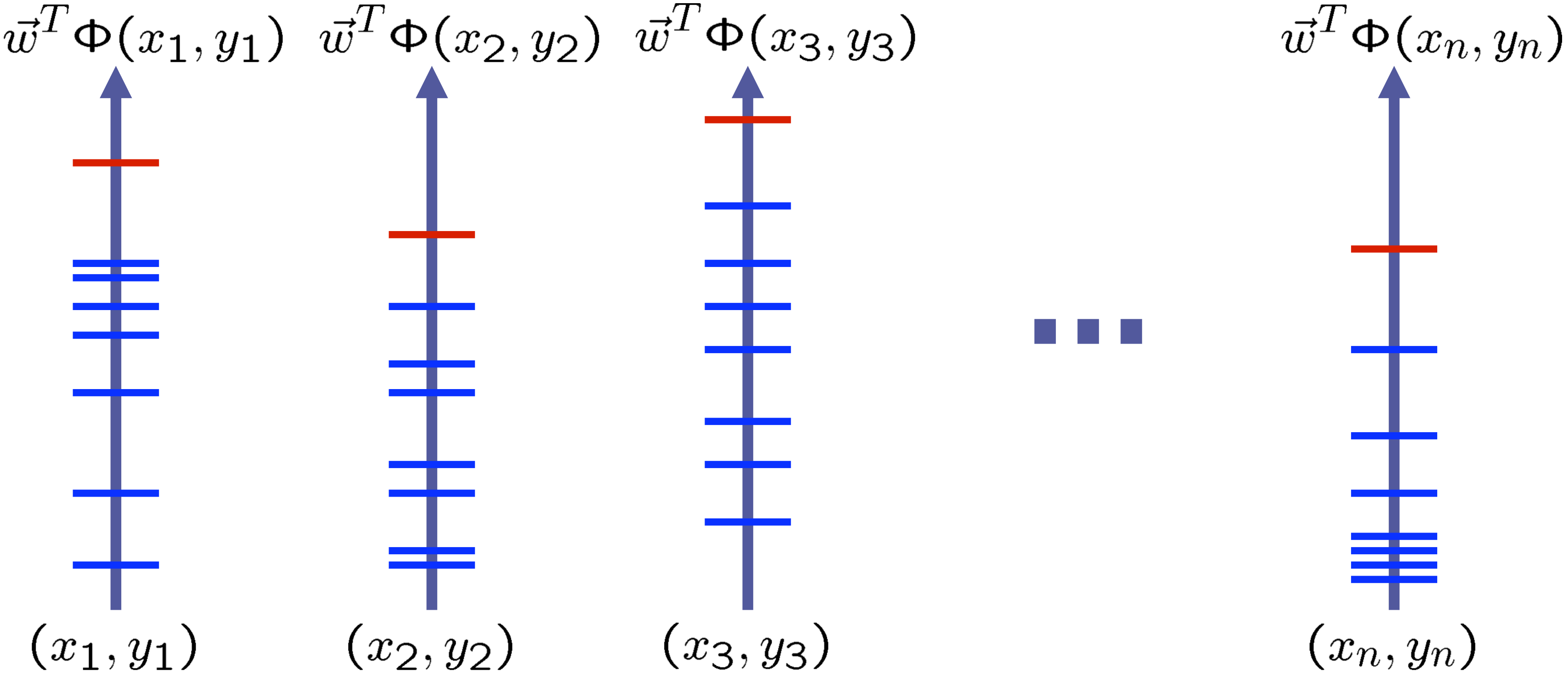
- Learn weight vector so that $\mathbf{w}^T \phi\left(\mathbf{x}_i, y\right)$ is max for correct y

$$\min_{\mathbf{w},\xi\geq 0} \left\|\mathbf{w}\right\|^2$$

$$s.t.\quad \forall y \cup Y \setminus y_1 : \mathbf{w}^T \phi\left(\mathbf{x}_1, y_1\right) \geq \mathbf{w}^T \phi\left(\mathbf{x}_1, y\right) + 1$$

$$s.t.\quad \ldots$$

$$s.t.\quad \forall y \cup Y \setminus y_n : \mathbf{w}^T \phi\left(\mathbf{x}_n, y_n\right) \geq \mathbf{w}^T \phi\left(\mathbf{x}_n, y\right) + 1$$


$\vec{w}^T \mathbf{\Phi}(x_1, y_1)$
$\vec{w}^T \mathbf{\Phi}(x_2, y_2)$
$\vec{w}^T \mathbf{\Phi}(x_3, y_3)$
$\vec{w}^T \mathbf{\Phi}(x_n, y_n)$
...
$(x_1, y_1)$
$(x_2, y_2)$
$(x_3, y_3)$
$(x_n, y_n)$

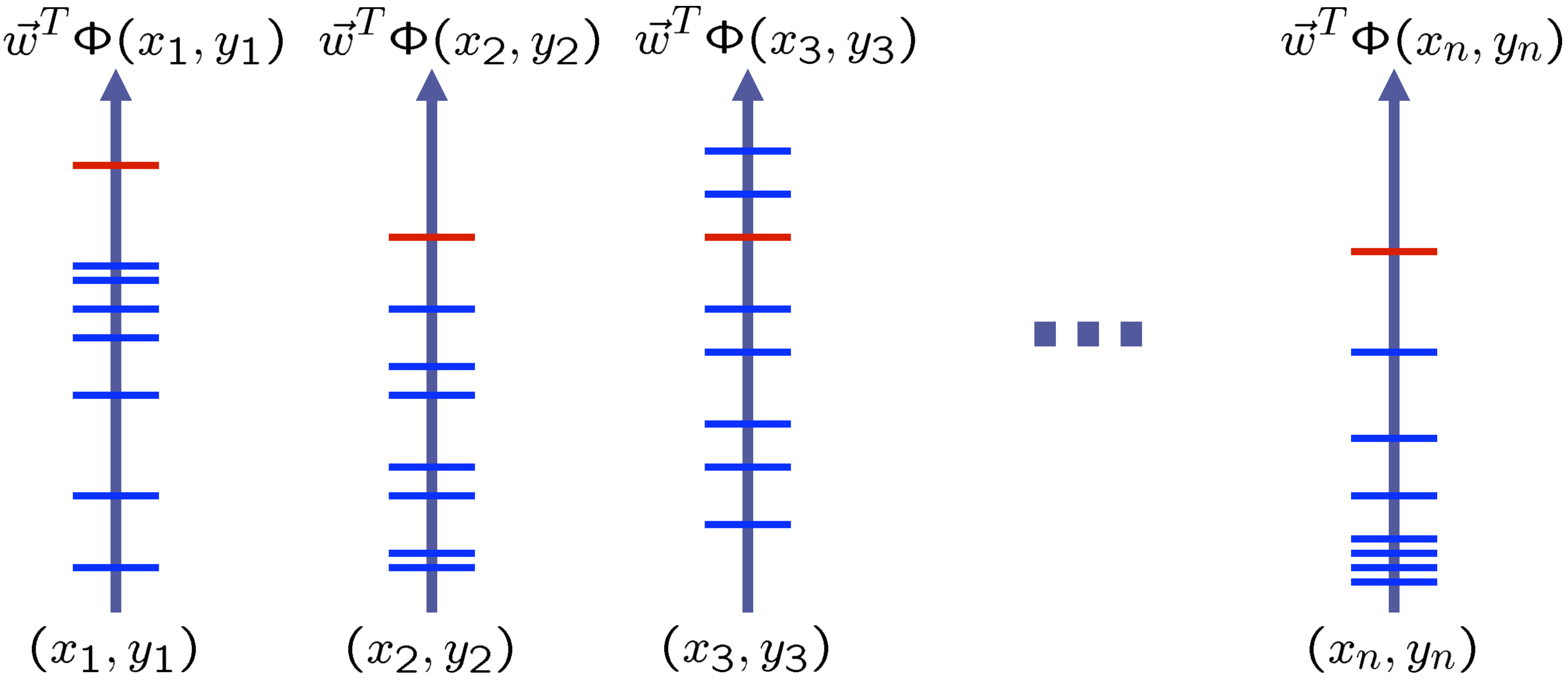
# Joint Feature Map with Slack

$$\min_{\mathbf{w},\xi\geq 0} \frac{1}{2}\left\|\mathbf{w}\right\|^2 + \frac{C}{n}\sum\nolimits_{i=1}^{n}\xi_i$$

$$s.t.\quad \forall y \cup Y \setminus y_1 : \mathbf{w}^T\phi\left(\mathbf{x}_1, y_1\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_1, y\right) + 1 - \xi_1$$

$$s.t.\quad \ldots$$

$$s.t.\quad \forall y \cup Y \setminus y_n : \mathbf{w}^T\phi\left(\mathbf{x}_n, y_n\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_n, y\right) + 1 - \xi_n$$



$\vec{w}^T\Phi(x_1, y_1)$
$\vec{w}^T\Phi(x_2, y_2)$
$\vec{w}^T\Phi(x_3, y_3)$
$\vec{w}^T\Phi(x_n, y_n)$
...
$(x_1, y_1)$
$(x_2, y_2)$
$(x_3, y_3)$
$(x_n, y_n)$

# The label loss function

- Not all classes are created equal, why clear each by 1?

$$\min_{\mathbf{w},\xi\geq 0} \tfrac{1}{2}\left\|\mathbf{w}\right\|^2 + \tfrac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$\Delta\left(y, y_1\right)$$

$$s.t. \quad \forall y \cup Y \setminus y_1 : \mathbf{w}^T\phi\left(\mathbf{x}_1, y_1\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_1, y\right) + 1 - \xi_1$$

$$s.t. \quad \ldots$$

$$s.t. \quad \forall y \cup Y \setminus y_n : \mathbf{w}^T\phi\left(\mathbf{x}_n, y_n\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_n, y\right) + 1 - \xi_n$$

- Instead of a constant 1 value, clear some classes more

$$\Delta\left(y, y_1\right) = Loss\ for\ predicting\ y\ instead\ of\ y_1$$

- For example, if y can be {lion, tiger, cat}

$$\Delta\left(tiger, lion\right) = \Delta\left(lion, tiger\right) = 1$$

$$\Delta\left(cat, lion\right) = \Delta\left(lion, cat\right) = 999$$

$$\Delta\left(tiger, tiger\right) = \Delta\left(cat, cat\right) = \Delta\left(lion, lion\right) = 0$$

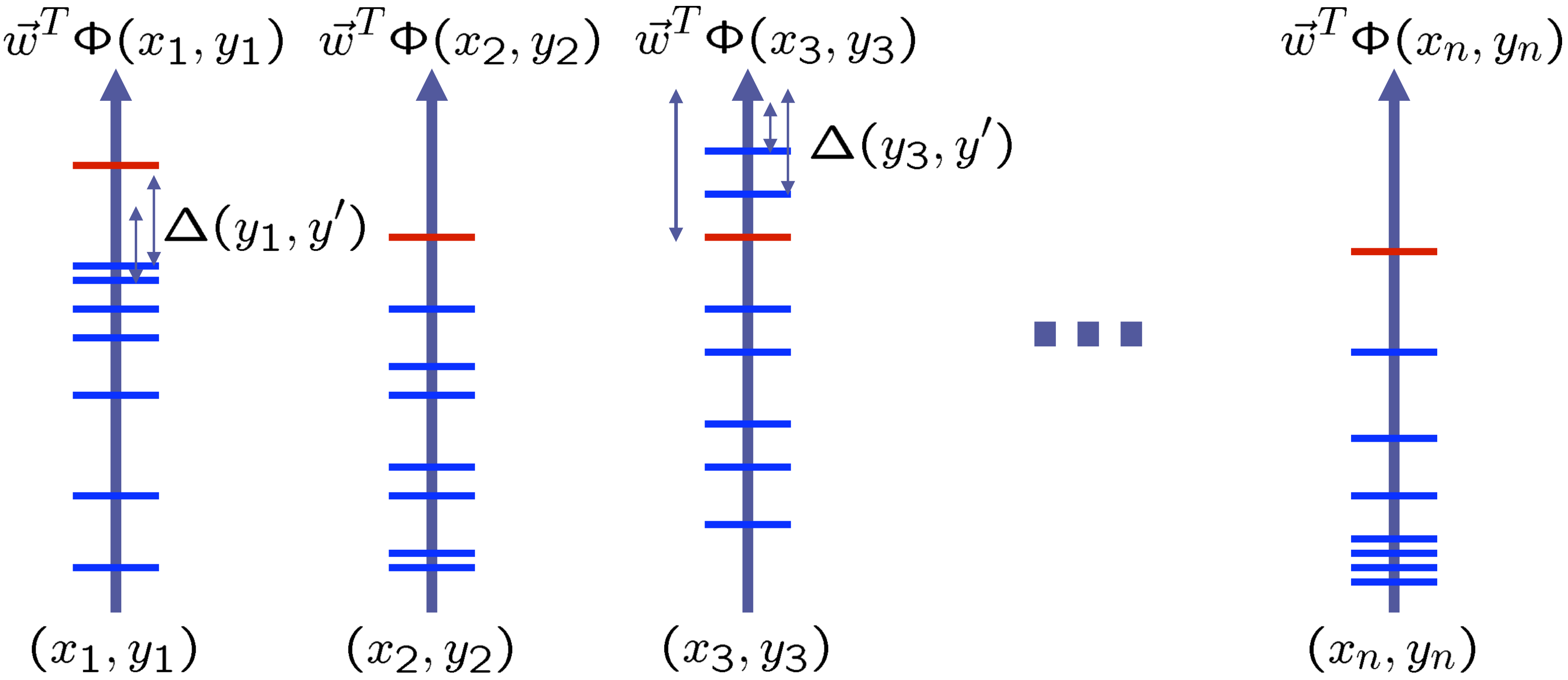
# Joint Feature Map with Any Loss

$$\min_{\mathbf{w},\xi\geq 0} \frac{1}{2}\left\|\mathbf{w}\right\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$s.t.\quad \forall y \cup Y \setminus y_1 : \mathbf{w}^T\phi\left(\mathbf{x}_1, y_1\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_1, y\right) + \Delta\left(y, y_1\right) - \xi_1$$

$$s.t.\quad \ldots$$

$$s.t.\quad \forall y \cup Y \setminus y_n : \mathbf{w}^T\phi\left(\mathbf{x}_n, y_n\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_n, y\right) + \Delta\left(y, y_n\right) - \xi_n$$



# Joint Feature Map with Slack

- Loss function $\Delta$ measures match between target & prediction

$$\begin{aligned}
&\min_{\mathbf{w},\xi\geq 0} \tfrac{1}{2}\left\|\mathbf{w}\right\|^2 + \tfrac{C}{n}\sum\nolimits_{i=1}^{n}\xi_i \\
&s.t. \quad \forall y \cup Y \setminus y_1 : \mathbf{w}^T\phi\left(\mathbf{x}_1, y_1\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_1, y\right) + \Delta\left(y, y_1\right) - \xi_1 \\
&s.t. \quad \ldots \\
&s.t. \quad \forall y \cup Y \setminus y_n : \mathbf{w}^T\phi\left(\mathbf{x}_n, y_n\right) \geq \mathbf{w}^T\phi\left(\mathbf{x}_n, y\right) + \Delta\left(y, y_n\right) - \xi_n
\end{aligned}$$

**Lemma: The training loss is upper bounded by**

$$Err_S(h) = \frac{1}{n}\sum_{i=1}^{n}\Delta(y_i, h(\vec{x}_i)) \leq \frac{1}{n}\sum_{i=1}^{n}\xi_i$$

# Generic Structural SVM (slow!)

- Application Specific Design of Model
  - Loss function $\Delta(y_i, y)$
  - Representation $\Phi(x, y)$
    - ➔ Markov Random Fields [Lafferty et al. 01, Taskar et al. 04]
- Prediction:

$$\hat{y} = argmax_{y \in Y} \{ \vec{w}^T \Phi(x,y) \}$$

- Training:

$$\begin{aligned} \min_{\vec{w}, \vec{\xi} \geq 0} \quad & \frac{1}{2} \vec{w}^T \vec{w} + \frac{C}{n} \sum_{i=1}^{n} \xi_i \\ s.t. \quad & \forall y \in Y \backslash y_1 : \vec{w}^T \Phi(x_1, y_1) \geq \vec{w}^T \Phi(x_1, y) + \Delta(y_1, y) - \xi_1 \\ & \ldots \\ & \forall y \in Y \backslash y_n : \vec{w}^T \Phi(x_n, y_n) \geq \vec{w}^T \Phi(x_n, y) + \Delta(y_n, y) - \xi_n \end{aligned}$$

- Applications: Parsing, Sequence Alignment, Clustering, etc.

# Reformulating the QP

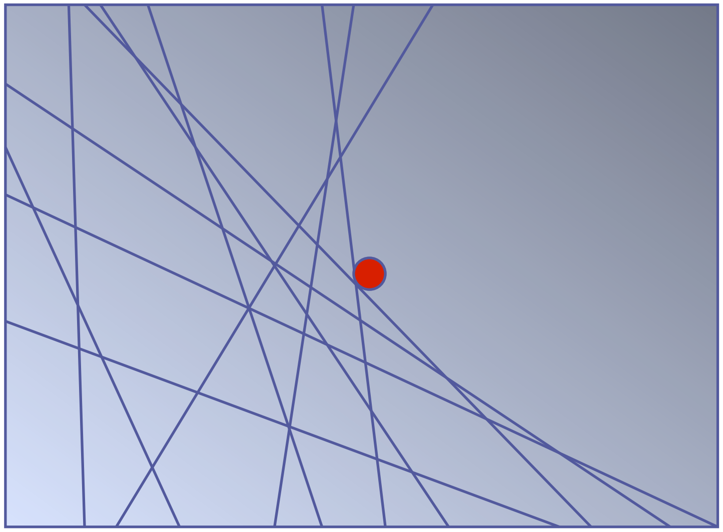
**n-Slack Formulation:** [TsoJoHoAl04]

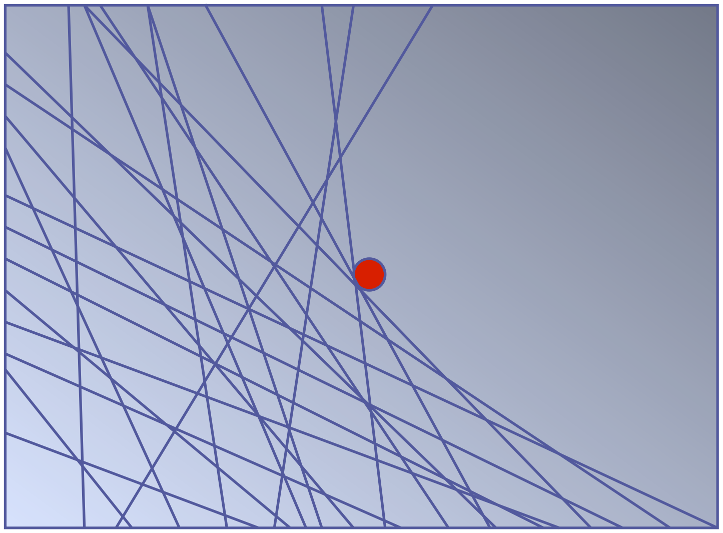
$$\min_{\vec{w},\vec{\xi}} \quad \frac{1}{2}\vec{w}^T\vec{w} + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad \forall y' \in Y : \vec{w}^T\Phi(x_1,y_1) - \vec{w}^T\Phi(x_1,y') \geq \Delta(y_1,y) - \xi_1$$

...

$$\forall y' \in Y : \vec{w}^T\Phi(x_n,y_n) - \vec{w}^T\Phi(x_n,y') \geq \Delta(y_n,y) - \xi_n$$





# Reformulating the QP

**n-Slack Formulation:** [TsoJoHoAl04]

$$\min_{\vec{w},\vec{\xi}} \quad \frac{1}{2}\vec{w}^T\vec{w} + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad \forall y' \in Y : \vec{w}^T\Phi(x_1,y_1) - \vec{w}^T\Phi(x_1,y') \geq \Delta(y_1,y) - \xi_1$$

$$\dots$$

$$\forall y' \in Y : \vec{w}^T\Phi(x_n,y_n) - \vec{w}^T\Phi(x_n,y') \geq \Delta(y_n,y) - \xi_n$$

$$\Leftrightarrow$$

**1-Slack Formulation:** [JoFinYu08]

$$\min_{\vec{w},\xi} \quad \frac{1}{2}\vec{w}^T\vec{w} + C\xi$$

$$s.t. \quad \forall y'_1 \dots y'_n \in Y : \frac{1}{n}\sum_{i=1}^{n}\left[\vec{w}^T\Phi(x_i,y_i) - \vec{w}^T\Phi(x_i,y'_i)\right] \geq \frac{1}{n}\sum_{i=1}^{n}\left[\Delta(y_i,y'_i)\right] - \xi$$

# Comparing n-Slack & 1-Slack

- Example: $Y = \left\{A, B, C\right\}\ and\ \ y_1 = A, y_2 = A, y_3 = B, y_4 = C$

n-Slack→n(k-1) constraints

$$y_1 \geq B, y_1 \geq C$$
$$y_2 \geq B, y_2 \geq C$$
$$y_3 \geq A, y_3 \geq C$$
$$y_4 \geq A, y_4 \geq B$$

1-Slack→k$^n$ constraints

$$\begin{aligned} y_1y_2y_3y_4 \geq\ & AAAA, AAAB, AAAC, AABA, \\ & AABB, \cdot, AACA, AACB, AACC, \\ & ABAA, ABAB, ABAC, ABBA, \\ & ABBB, ABBC, ABCA, ABCB, \\ & ABCC, ACAA, ACAB, ACAC, \ldots \end{aligned}$$

- Idea: we expect only a few constraints to be active
- Cutting-Plane: a greedy approach to QP
- Solve with only a few constraints at a time
- If solution violates come constraints, add them back in
- If we are smart about which ones to add, may not need k$^n$

# 1-Slack Cutting-Plane Algorithm

```
Input: $(x_1, y_1), \ldots, (x_n, y_n), C, \epsilon$
$S \leftarrow \emptyset, \vec{w} \leftarrow 0, \xi \leftarrow 0$
REPEAT
    FOR $i = 1, \ldots, n$
        Compute $y_i' = argmax_{y \in Y}\{\Delta(y_i,y) + \vec{w}^T\Phi(x_i,y)\}$
    ENDFOR
    IF $\sum_{i=1}^{n}\left[\Delta(y_i,y_i') - \vec{w}^T[\Phi(x_i,y_i) - \Phi(x_i,y_i')])\right] > \xi + \epsilon$
        – $S \leftarrow S \cup \{\vec{w}^T \frac{1}{n}\sum_{i=1}^{n}[\Phi(x_i,y_i) - \Phi(x_i,y_i')] \geq \frac{1}{n}\sum_{i=1}^{n}\Delta(y_i,y_i') - \xi\}$
        optimize StructSVM over S to get w and ξ
    ENDIF
UNTIL solution has not changed during iteration
```

[Jo06] [JoFinYu08]

# Polynomial Sparsity Bound

- Theorem: The cutting-plane algorithm finds a solution to the Structural SVM soft-margin optimization problem in the 1-slack formulation after adding at most

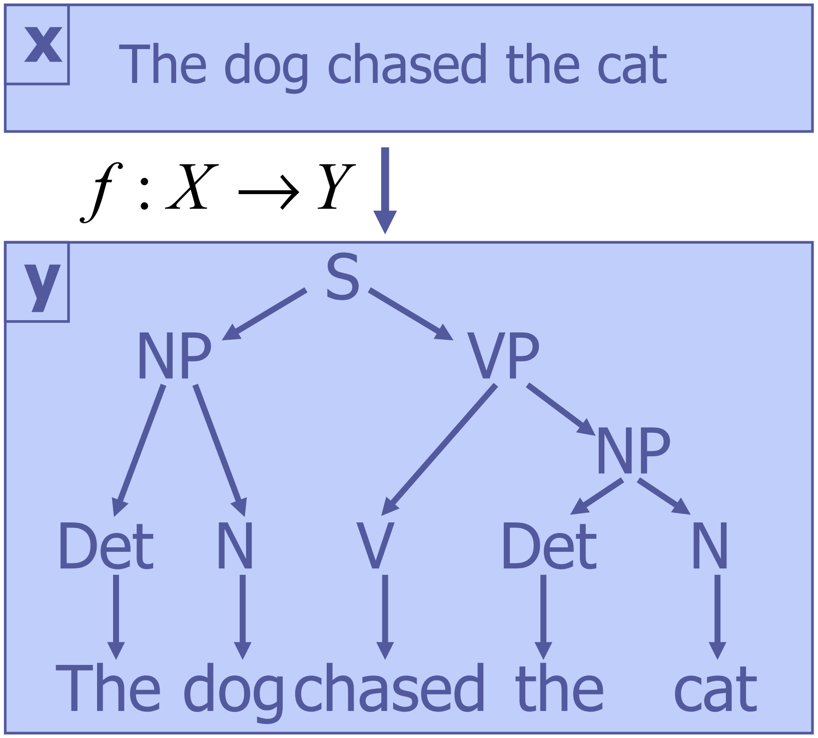
$$\left\lceil \log_2\left(\frac{\Delta}{4R^2C}\right)\right\rceil + \left\lceil \frac{16R^2C}{\varepsilon}\right\rceil$$

constraints to the working set S, so that the primal constraints are feasible up to a precision and the objective on S is optimal. The loss has to be bounded $0 \leq \Delta(y_i,y) \leq \Delta$ , and $2||\Phi(x,y)|| \leq R$ .

[Jo03] [Jo06] [TeoLeSmVi07] [JoFinYu08]

# Joint Feature Map for Trees

- Weighted Context Free Grammar
  - Each rule (e.g. $S \rightarrow NP\ VP$) has a weight
  - Score of a tree is the sum of its weights
  - Find highest scoring tree $h(\vec{x}) = argmax_{y \in Y} \left[ \vec{w}^T \Phi(x,y) \right]$



$$\Phi(\mathbf{x},\mathbf{y}) = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ \vdots \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{array}{c} S \rightarrow NP\ VP \\ S \rightarrow NP \\ NP \rightarrow Det\ N \\ VP \rightarrow V\ \ NP \\ \\ Det \rightarrow dog \\ Det \rightarrow the \\ N \rightarrow dog \\ V \rightarrow chased \\ N \rightarrow cat \end{array}$$

# Experiments: NLP

Implementation

- Incorporated modified version of Mark Johnson's CKY parser
- Learned weighted CFG with $\epsilon = 0.01, C = 1$

Data

- Penn Treebank sentences of length at most 10 (start with POS)
- Train on Sections 2-22: 4098 sentences
- Test on Section 23: 163 sentences

| | Test Accuracy | |
|---|---|---|
| Method | Acc | $F_1$ |
| PCFG with MLE | 55.2 | 86.0 |
| SVM with (1-$F_1$)-Loss | **58.9** | **88.5** |

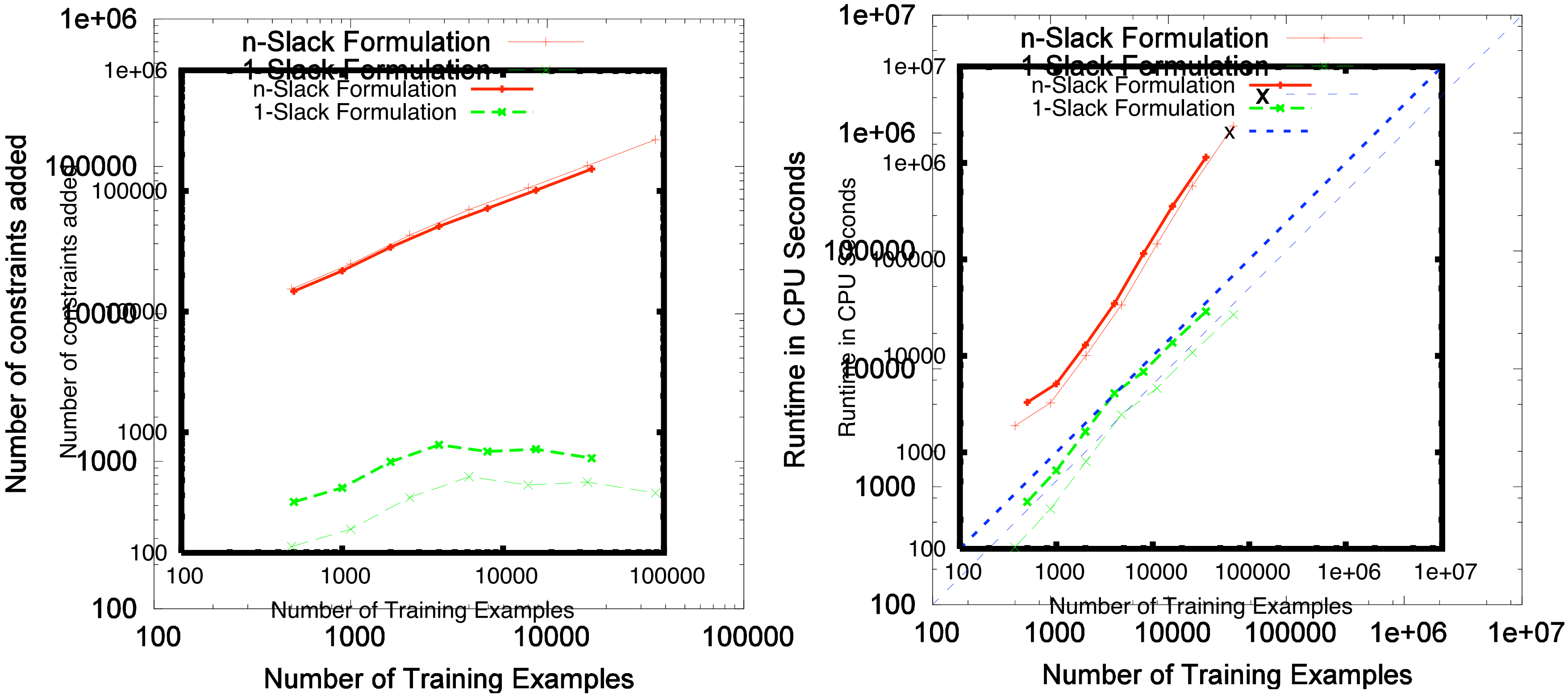
[TsoJoHoAl04]

- more complex features [TaKlCoKoMa04]

# Experiments of 1-Slack versus n-Slack

## Part-of-speech tagging on Penn Treebank

## ~36,000 examples, ~250,000 features in linear HMM model



[JoFinYu08]

# StructSVM for Any Problem

- General
  - SVM-struct algorithm and implementation

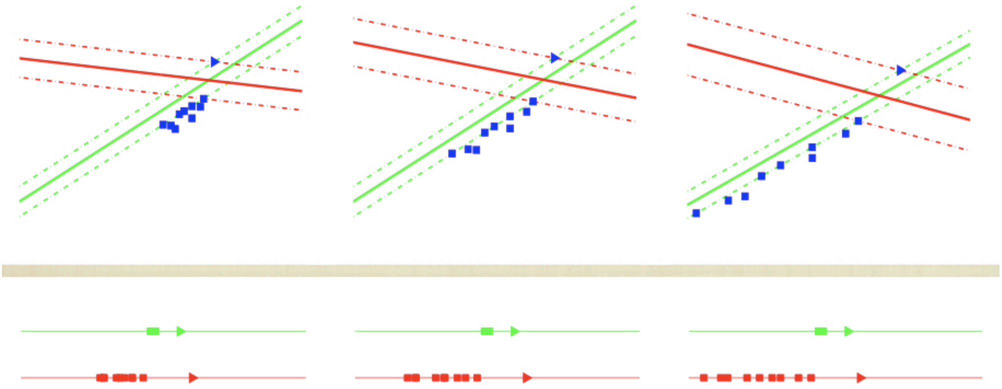
    http://svmlight.joachims.org
  - Theory (e.g. training-time linear in n)
- Application specific
  - Loss function $\Delta(y_i, y)$
  - Representation $\Phi(x, y)$
  - Algorithms to compute

    $$\hat{y} = argmax_{y \in Y} \{ \vec{w}^T \Phi(x_i, y) \}$$

    $$\hat{y} = argmax_{y \in Y} \{ \Delta(y_i, y) + \vec{w}^T \Phi(x_i, y) \}$$
- Properties
  - General framework for discriminative learning
  - Direct modeling, not reduction to classification/regression
  - “Plug-and-play”

# Struct SVM with Relative Margin

- Add relative margin constraints to struct SVM (ShiJeb09)
- Correct beats wrong labels but not by too much (relatively)



$$\min_{\mathbf{w},\xi\geq 0} \tfrac{1}{2}\left\|\mathbf{w}\right\|^2 + \tfrac{C}{n}\sum\nolimits_{i=1}^{n}\xi_i$$

$$s.t. \quad \forall y \cup Y \setminus y_1 : B \geq \mathbf{w}^T\phi\left(\mathbf{x}_1, y_1\right) - \mathbf{w}^T\phi\left(\mathbf{x}_1, y\right) \geq \Delta\left(y, y_1\right) - \xi_1$$

$$s.t. \quad \ldots$$

$$s.t. \quad \forall y \cup Y \setminus y_n : B \geq \mathbf{w}^T\phi\left(\mathbf{x}_n, y_n\right) - \mathbf{w}^T\phi\left(\mathbf{x}_n, y\right) \geq \Delta\left(y, y_n\right) - \xi_n$$

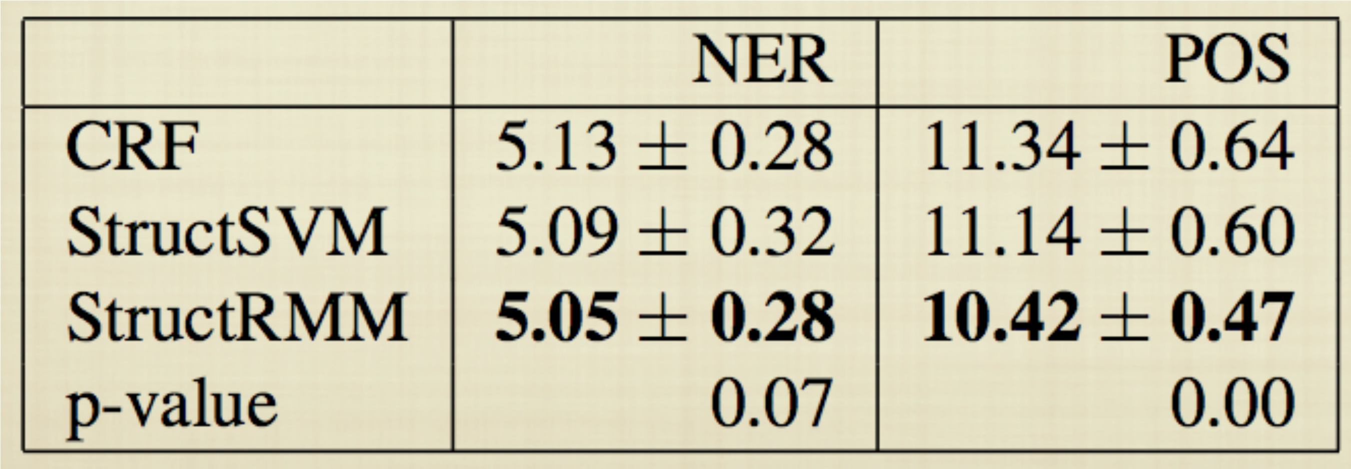
- Needs both $\arg\max_{y\in Y} \mathbf{w}^T\phi\left(\mathbf{x}, y\right)$ and $\arg\min_{y\in Y} \mathbf{w}^T\phi\left(\mathbf{x}, y\right)$

# Struct SVM with Relative Margin

- Similar bound holds for relative margin
- Maximum # of cuts is

$$\max\left\{\frac{2CR^2}{\varepsilon_B^2}, \frac{2n}{\varepsilon}, \frac{8CR^2}{\varepsilon^2}\right\}$$

- Try sequence learning problems for Hidden Markov Modeling
- Consider named entity recognition (NER) task
- Consider part-of-speech (POS) task



| | NER | POS |
|---|---|---|
| CRF | 5.13 ± 0.28 | 11.34 ± 0.64 |
| StructSVM | 5.09 ± 0.32 | 11.14 ± 0.60 |
| StructRMM | **5.05 ± 0.28** | **10.42 ± 0.47** |
| p-value | 0.07 | 0.00 |