Structured Output Prediction with Structural Support Vector Machines

Thorsten Joachims

Cornell University

Department of Computer Science

Joint work with
T. Hofmann, I. Tsochantaridis, Y. Altun (Brown/Google/TTI)
T. Finley, R. Elber, Chun-Nam Yu, Yisong Yue, F. Radlinski
P. Zigoris, D. Fleisher (Cornell)

Supervised Learning

- **Assume:** Data is i.i.d. from

$$P(X, Y)$$

- **Given:** Training sample

$$S = ((x_1, y_1), \ldots, (x_n, y_n))$$

- **Goal:** Find function from input space X to output space Y

$$h : X \longrightarrow Y$$

Complex objects

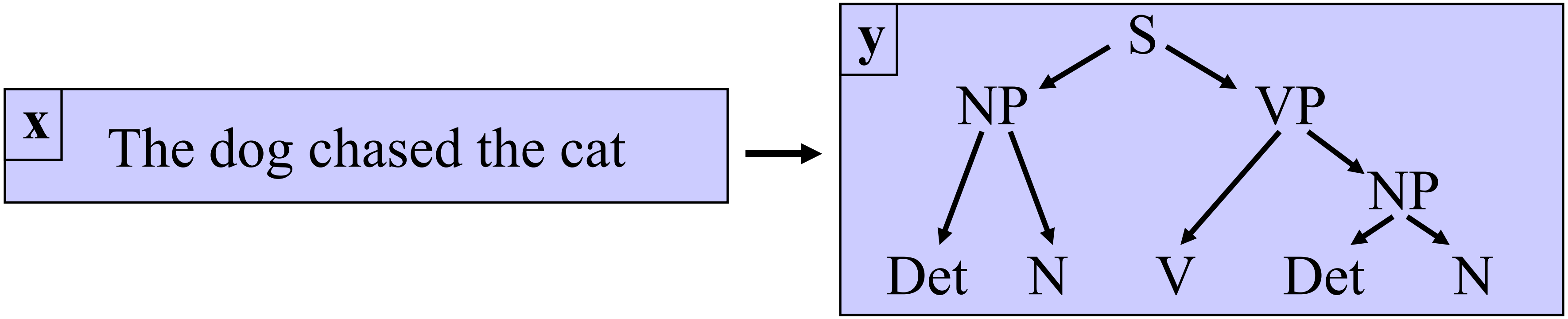
with low risk / prediction error

$$R(h) = \int \Delta(h(x), y) dP(X, Y)$$

- **Methods:** Kernel Methods, SVM, Boosting, etc.

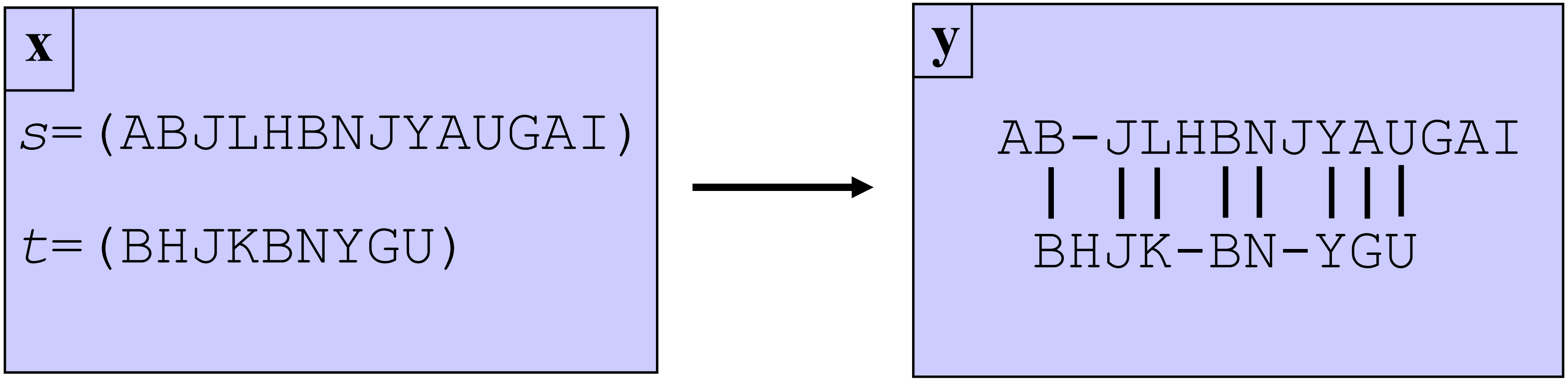
Examples of Complex Output Spaces

- **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

- **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.



x

s=(ABJLHBNJYAUGAI)

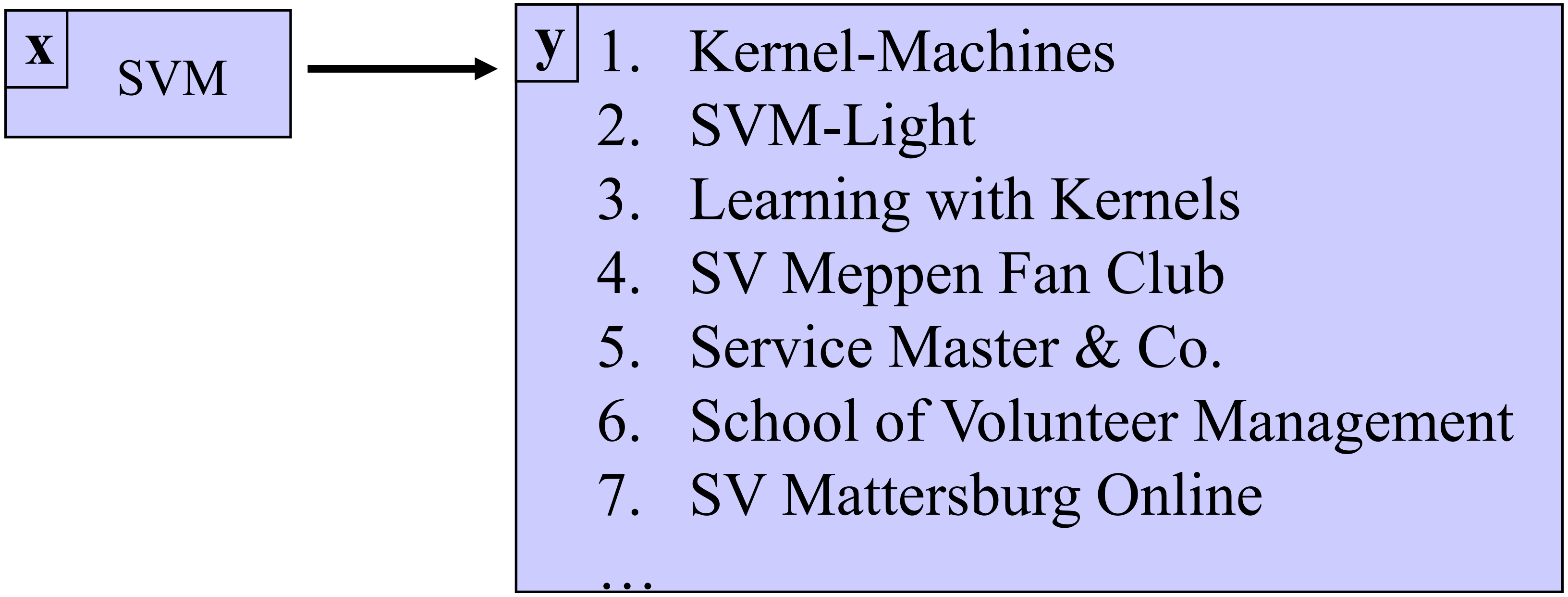
t=(BHJKBNYGU)

y

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

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. AvgPrec)



x

SVM

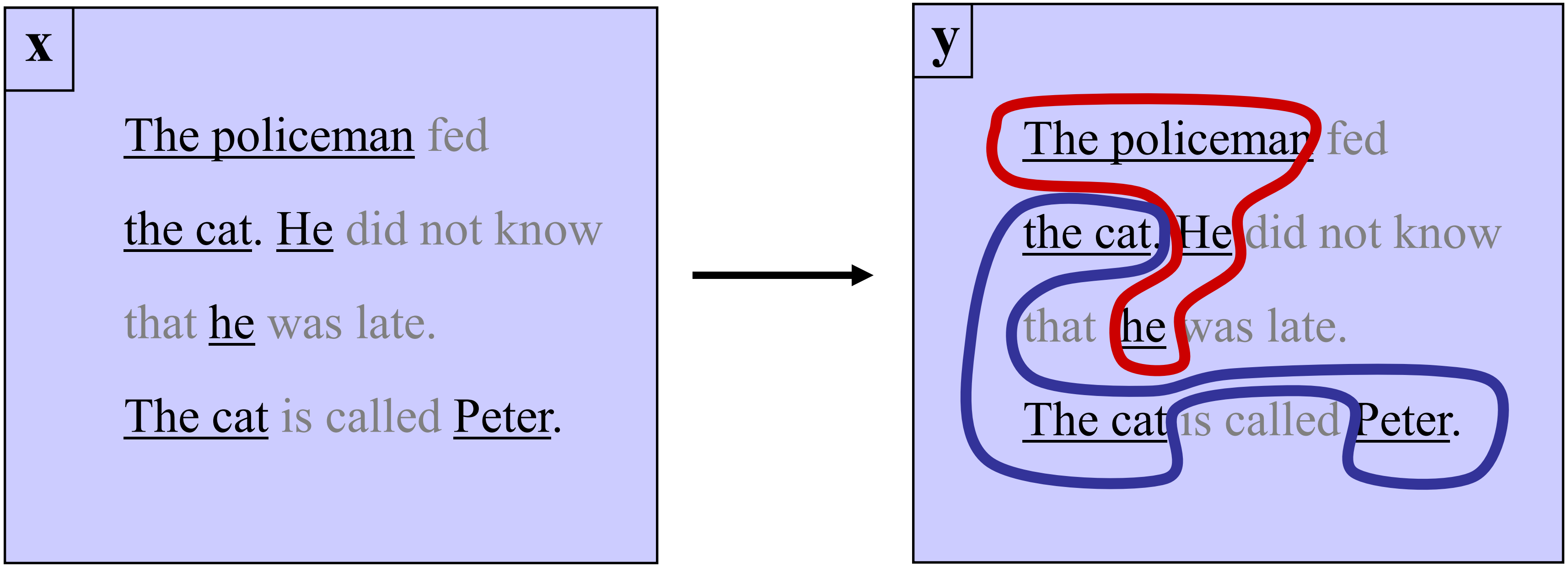
y

1. Kernel-Machines
2. SVM-Light
3. Learning with Kernels
4. SV Meppen Fan Club
5. Service Master & Co.
6. School of Volunteer Management
7. SV Mattersburg Online

…

Examples of Complex Output Spaces

- **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.



x

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

y

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

Examples of Complex Output Spaces

- **and many many more:**
  - Sequence labeling (e.g. part-of-speech tagging, named-entity recognition) [Lafferty et al. 01, Altun et al. 03]
  - Collective classification (e.g. hyperlinked documents) [Taskar et al. 03]
  - Multi-label classification (e.g. text classification) [Finley & Joachims 08]
  - Binary classification with non-linear performance measures (e.g. optimizing F1-score, avg. precision) [Joachims 05]
  - Inverse reinforcement learning / planning (i.e. learn reward function to predict action sequences) [Abbeel & Ng 04]

Overview

- **Task: Discriminative learning with complex outputs**
- ➡ **Related Work**
- **SVM algorithm for complex outputs**
  - Predict trees, sequences, equivalence relations, alignments
  - General non-linear loss functions
  - Generic formulation as convex quadratic program
- **Training algorithms**
  - n-slack vs. 1-slack formulation
  - Correctness and sparsity bound
- **Applications**
  - Sequence alignment for protein structure prediction [w/ Chun-Nam Yu]
  - Diversification of retrieval results in search engines [w/ Yisong Yue]
  - Supervised clustering [w/ Thomas Finley]
- **Conclusions**

Why Discriminative Learning for Structured Outputs?

- **Im** **it!**
  - – er 06]
  - –
- **Di**
  - – ification

Precision/Recall Break-Even Point	Naïve Bayes	Linear SVM
Reuters	72.1	87.5
WebKB	82.0	90.3
Ohsumed	62.4	71.6

- **Improve upon prediction accuracy of existing generative methods!**
  - Natural language parsing: generative models like probabilistic context-free grammars
  - SVM outperforms naïve Bayes for text classification [Joachims, 1998] [Dumais et al., 1998]
- **More flexible models!**
  - Avoid generative (independence) assumptions
  - Kernels for structured input spaces and non-linear functions

Related Work

- **Generative training (i.e. model P(Y,X))**
  - Hidden-Markov models
  - Probabilistic context-free grammars
  - Markov random fields
  - etc.
- **Discriminative training (i.e. model P(Y|X) or minimize risk)**
  - Multivariate output regression [Izeman, 1975] [Breiman & Friedman, 1997]
  - Kernel Dependency Estimation [Weston et al. 2003]
  - Transformer networks [LeCun et al, 1998]
  - Conditional HMM [Krogh, 1994]
  - Conditional random fields [Lafferty et al., 2001]
  - Perceptron training of HMM [Collins, 2002]
  - Maximum-margin Markov networks [Taskar et al., 2003]
  - Structural SVMs [Altun et al. 03] [Joachims 03] [TsoHoJoAl04]

Overview

- **Task: Discriminative learning with complex outputs**
- **Related Work**
- **SVM algorithm for complex outputs**
  - Predict trees, sequences, equivalence relations, alignments
  - General non-linear loss functions
  - Generic formulation as convex quadratic program
- **Training algorithms**
  - n-slack vs. 1-slack formulation
  - Correctness and sparsity bound
- **Applications**
  - Sequence alignment for protein structure prediction [w/ Chun-Nam Yu]
  - Diversification of retrieval results in search engines [w/ Yisong Yue]
  - Supervised clustering [w/ Thomas Finley]
- **Conclusions**

Classification

- **Training Examples:** [redacted]
- **Hypothesis Space:** $h(\mathbf{x}$
- **Training:** Find hyperp

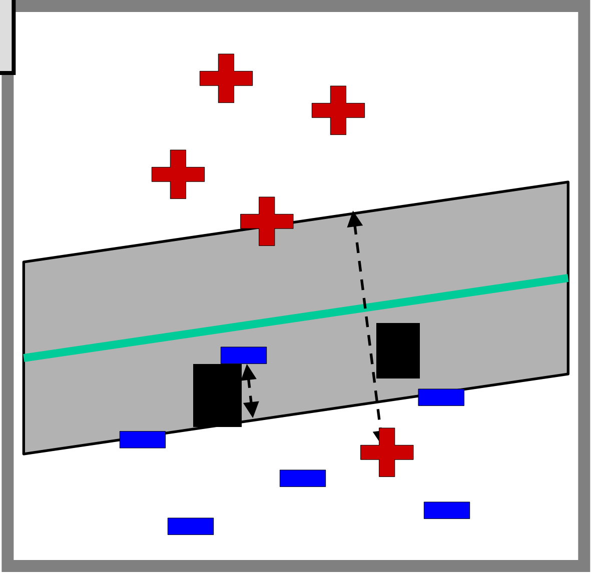
Dual Opt. Problem:

[redacted]

[redacted]

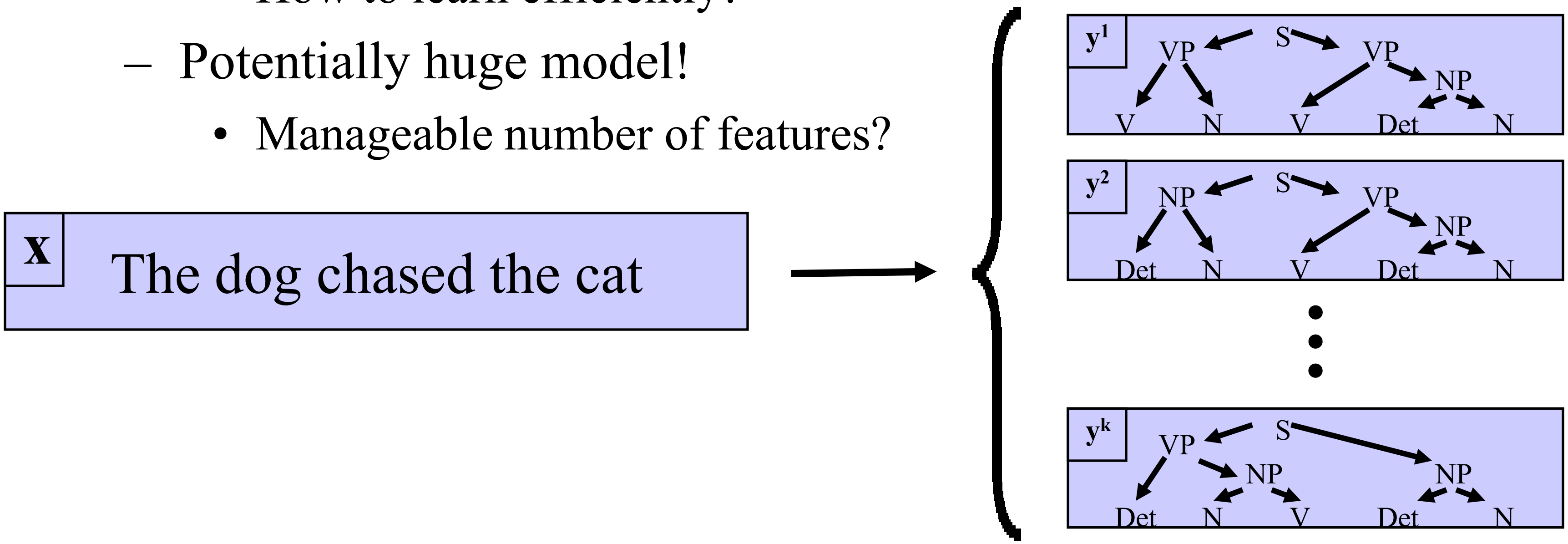
Primal Opt. Problem:

[redacted]



Challenges in Discriminative Learning with Complex Outputs

- **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

Training: Find [redacted] that solve

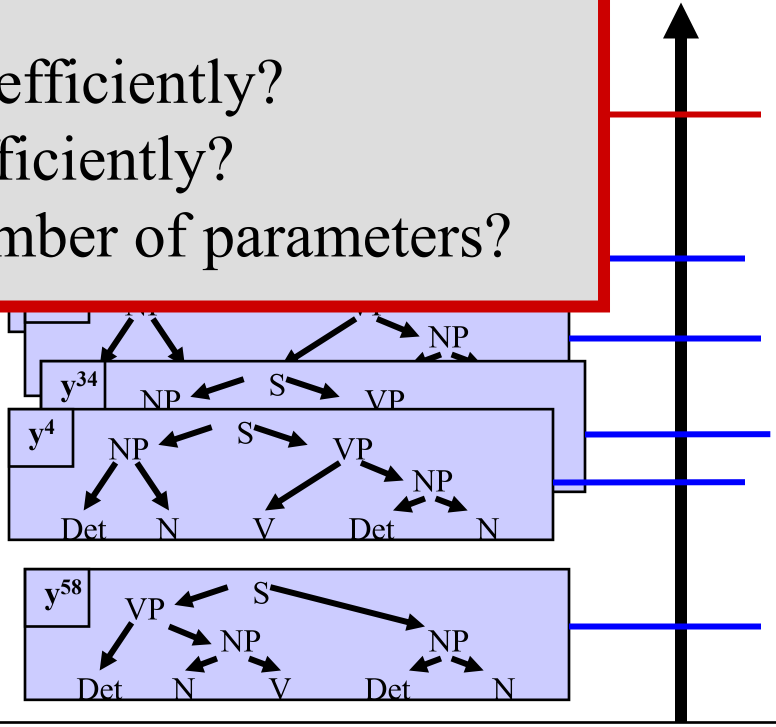
- [redacted]

[redacted]

Problems

- How to predict efficiently?
- How to learn efficiently?
- Manageable number of parameters?

x The dog chased the cat



[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

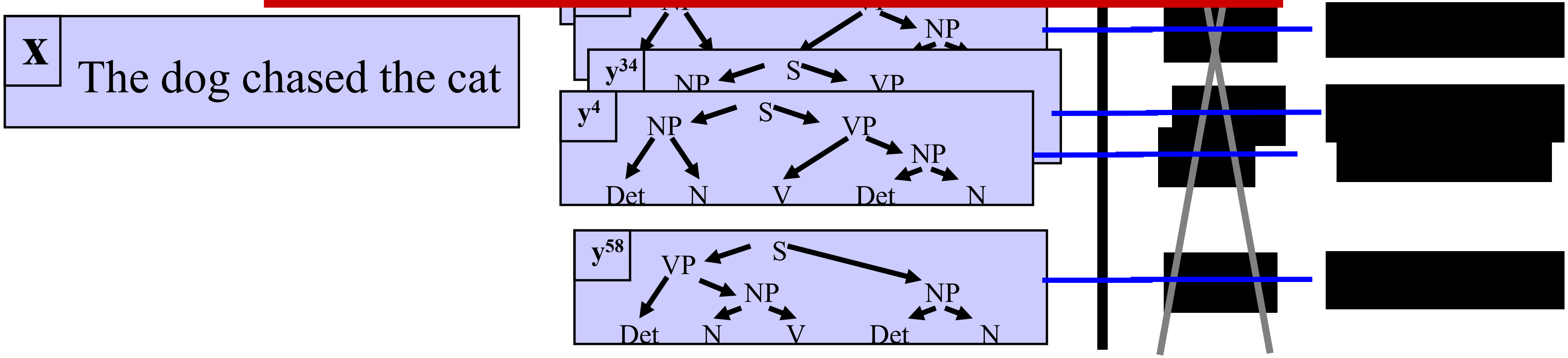
Joint Feature Map

- **Feature vector $\Phi(x, y)$ that describes match between *x* and *y***
- **Learn single weight vector and rank by $\vec{w}^T\Phi(x, y)$**

$$h(\vec{x}) = argmax_{y \in Y} \left[\vec{w}^T \Phi(x, y) \right]$$

Problems

- How to predict efficiently?
- How to learn efficiently?
- Manageable number of parameters? ✓

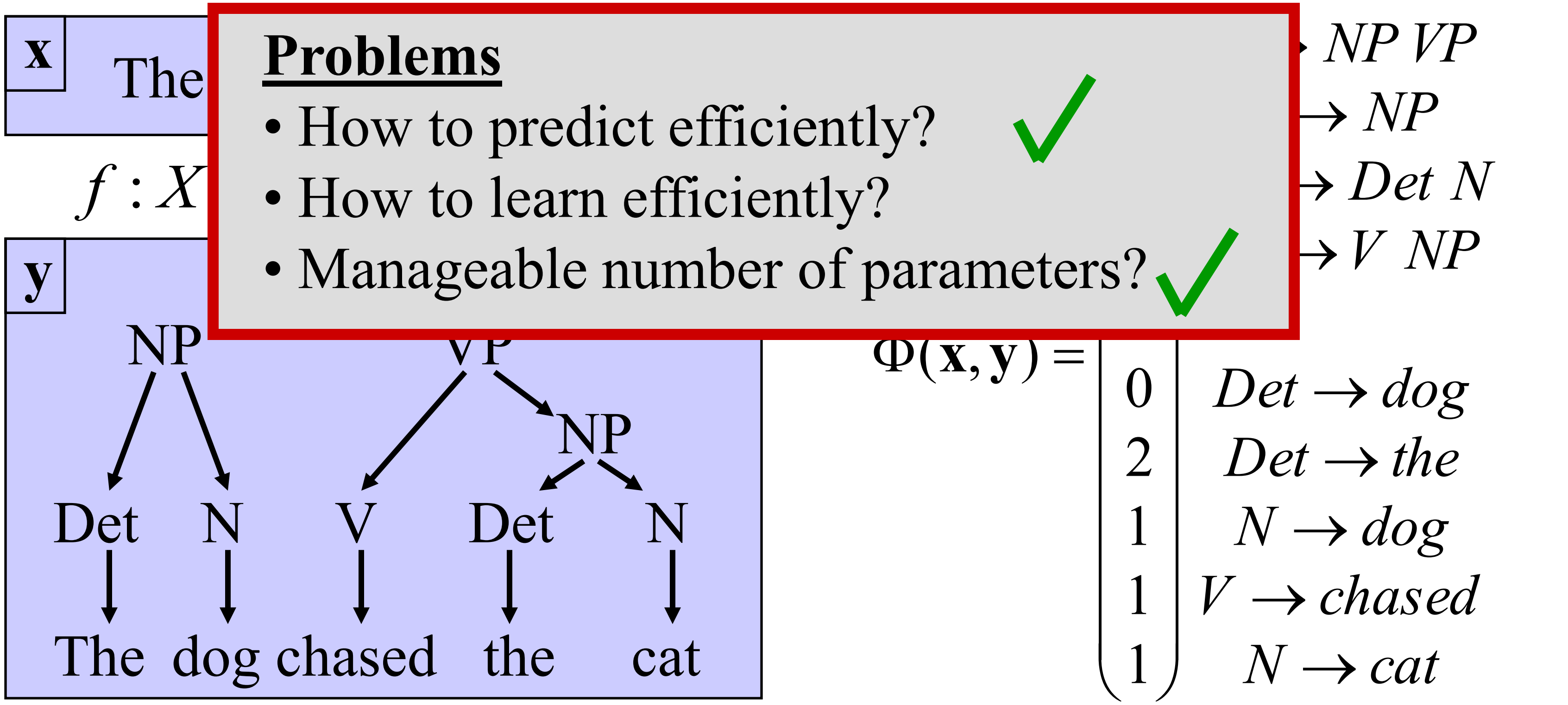


[redacted]

Joint Feature Map for Trees

CKY Parser

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



Problems

- How to predict efficiently? ✓
- How to learn efficiently?
- Manageable number of parameters? ✓

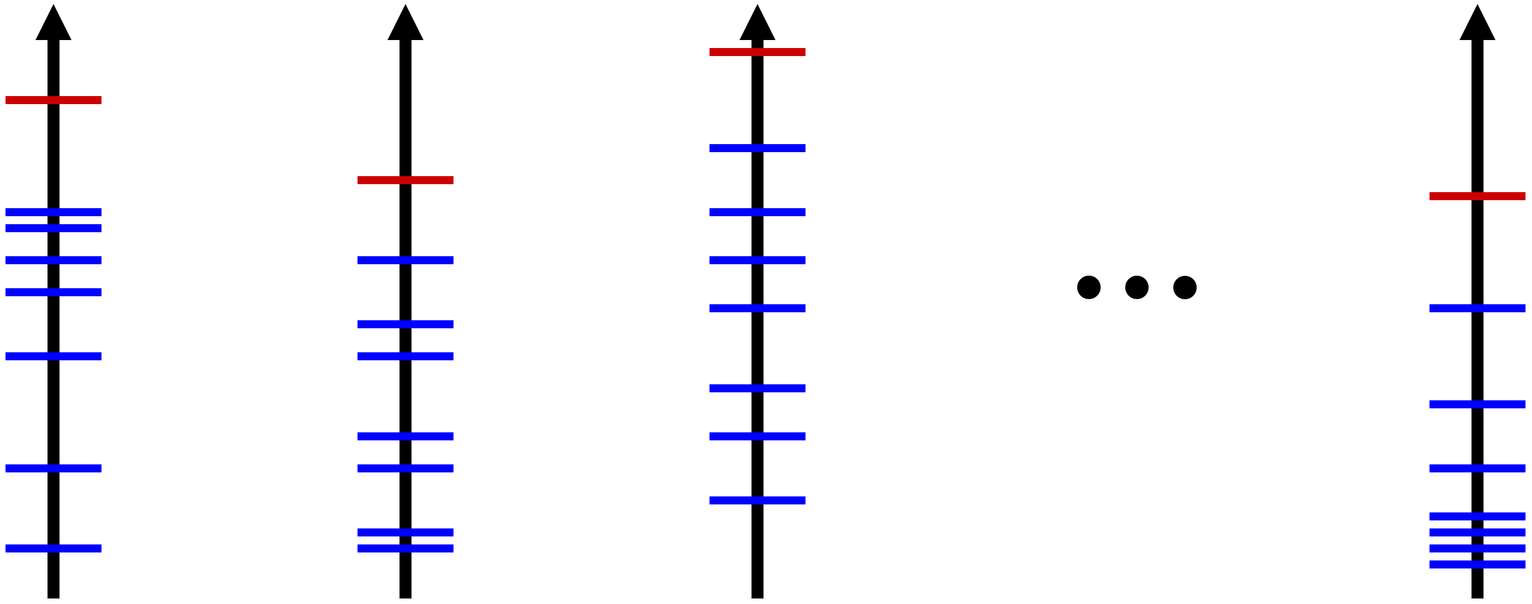
Structural Support Vector Machine

Hard-margin optimization problem:

[redacted]

- ,
- t y

[redacted] [redacted] [redacted] [redacted]



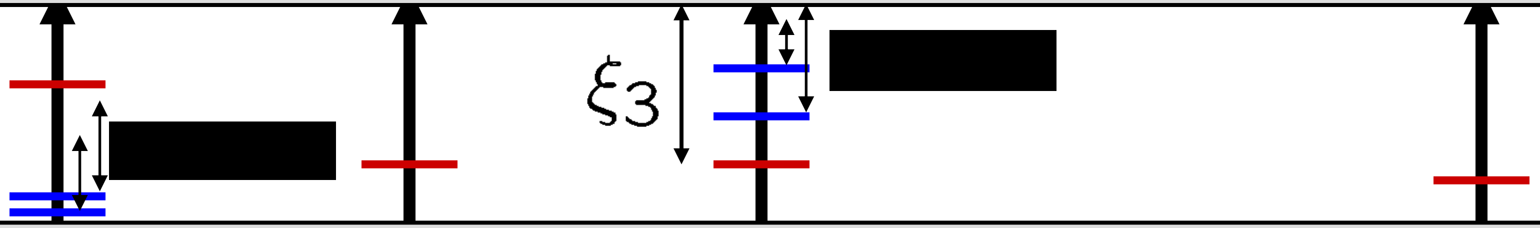
[redacted] [redacted] [redacted] [redacted]

Loss Functions: Soft-Margin Struct SVM

Soft-margin optimization problem:

[redacted]

d



[redacted]

[redacted]

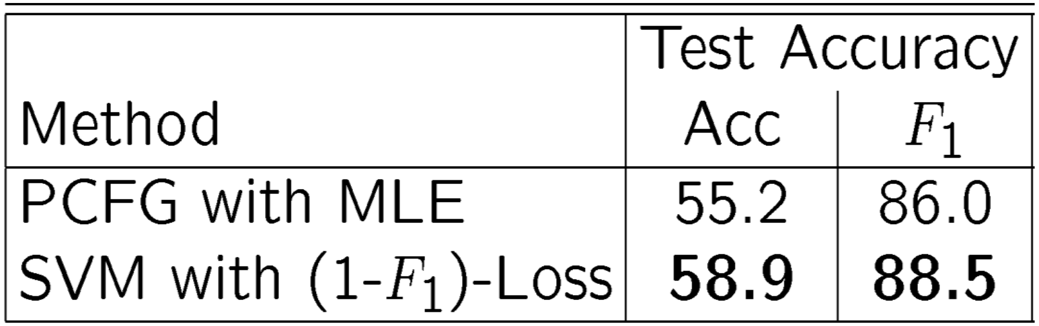
Lemma: The training loss is upper bounded by

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

[redacted] [redacted] [redacted] [redacted]

Experiment: Natural Language Parsing

- **Implemention**
  - Incorporated modified version of Mark Johnson's CKY parser
  - Learned weighted CFG with [redacted]
- **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]

Generic Structural SVM

- **Application Specific Design of Model**
  - Loss function [redacted]
  - 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:**

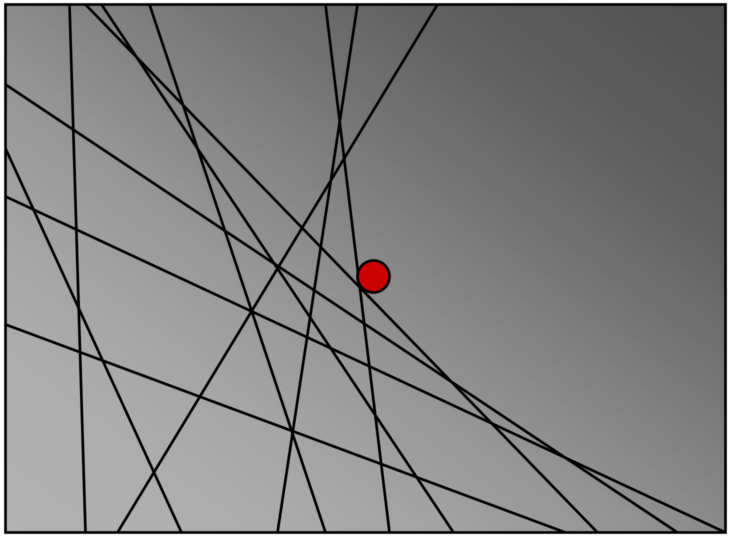
[redacted]

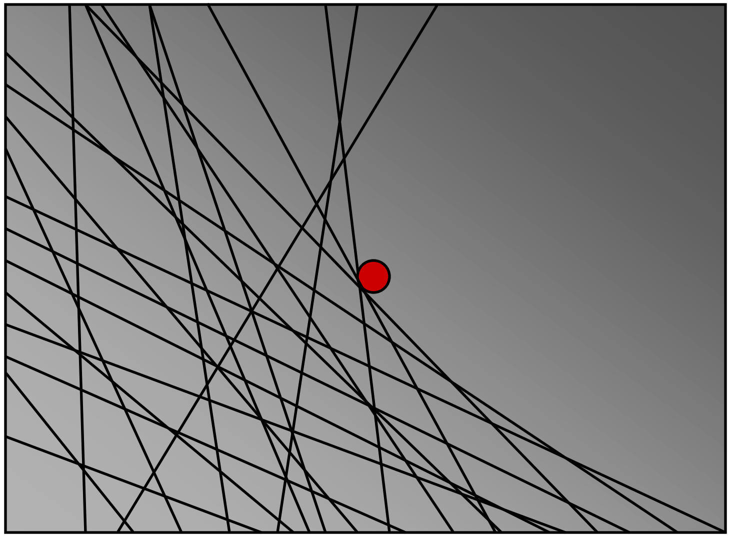
- **Applications:** Parsing, Sequence Alignment, Clustering, etc.

Reformulation of the Structural SVM QP

n-Slack Formulation: [TsoJoHoAl04]

[redacted]





Reformulation of the Structural SVM QP

n-Slack Formulation: [TsoJoHoAl04]

[redacted]

⇔

1-Slack Formulation: [JoFinYu08]

[redacted]

Cutting-Plane Algorithm for Structural SVM (1-Slack Formulation)

```
• Input: [redacted]
• [redacted]
• REPEAT
    – FOR [redacted]
        – Compute [redacted]
    – ENDFOR
    – IF [redacted]
        –
        – [redacted] optimize StructSVM over $S$
    – ENDIF
• UNTIL $S$ has not changed during iteration
```

Find most violated constraint

Violated by more than ε ?

Add constraint to working set

[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

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

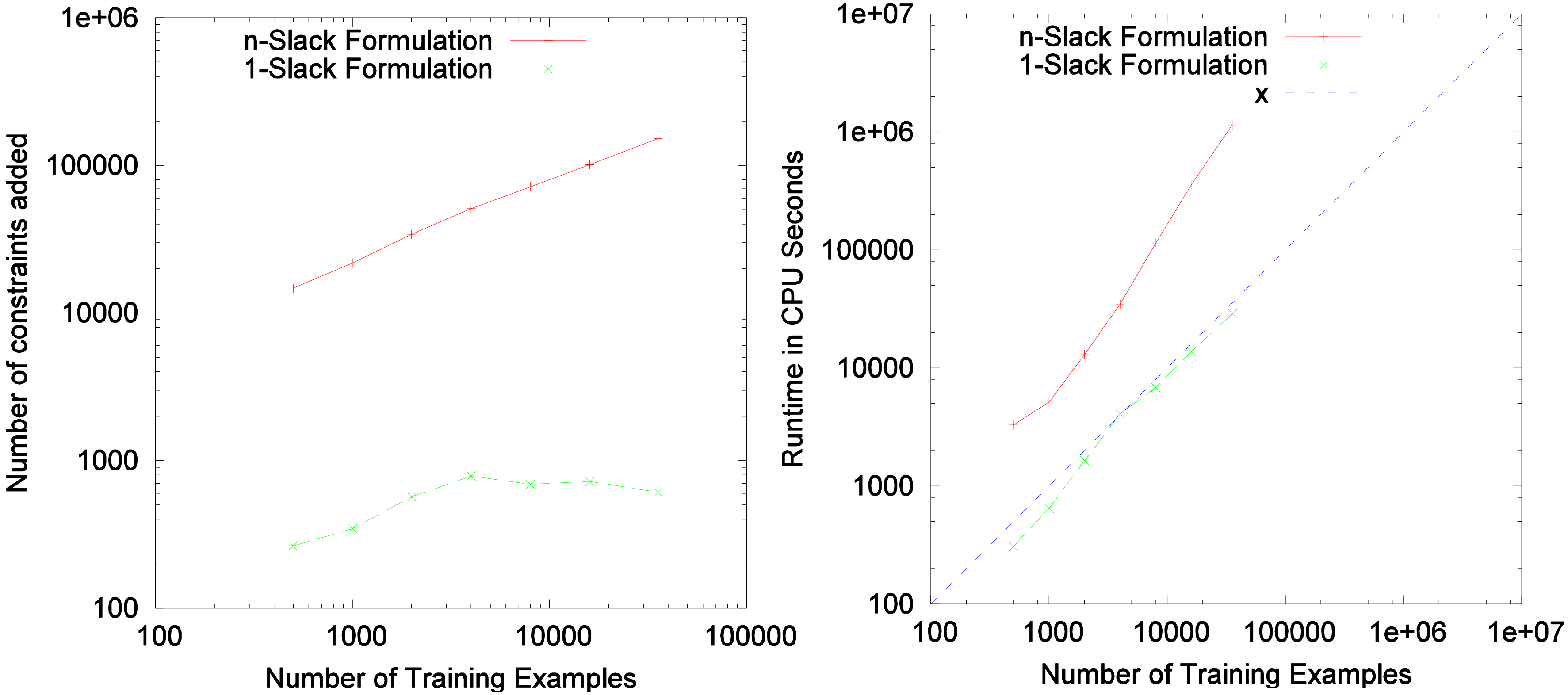
constraints to the working set S, so that the primal constraints are feasible up to a precision ϵ and the objective on S is optimal. The loss has to be bounded [redacted], and [redacted].

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

Empirical Comparison: Different Formulations

Experiment Setup:

- Part-of-speech tagging on Penn Treebank corpus
- ~36,000 examples, ~250,000 features in linear HMM model



[JoFinYu08]

Applying StructSVM to New Problem

- **General**
  - SVM-struct algorithm and implementation
    `http://svmlight.joachims.org`
  - Theory (e.g. training-time linear in n)
- **Application specific**
  - Loss function [redacted]
  - Representation $\Phi(x, y)$
  - Algorithms to compute

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

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

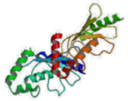
- **Properties**
  - General framework for discriminative learning
  - Direct modeling, not reduction to classification/regression
  - "Plug-and-play"

Overview

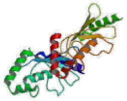
- **Task: Discriminative learning with complex outputs**
- **Related Work**
- **SVM algorithm for complex outputs**
  - Predict trees, sequences, equivalence relations, alignments
  - General non-linear loss functions
  - Generic formulation as convex quadratic program
- **Training algorithms**
  - n-slack vs. 1-slack formulation
  - Correctness and sparsity bound
- **Applications**
  - Sequence alignment for protein structure prediction [w/ Chun-Nam Yu]
  - Diversification of retrieval results in search engines [w/ Yisong Yue]
  - Supervised clustering [w/ Thomas Finley]
- **Conclusions**

Comparative Modeling of Protein Structure

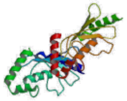
- **Goal: Predict structure from sequence**

  h("APPGEAYLQV") → 

- **Hypothesis:**
  - Amino Acid sequences for into structure with lowest energy
  - Problem: Huge search space (> 2^{100} states)
- **Approach: Comparative Modeling**
  - Similar protein sequences fold into similar shapes
    → use known shapes as templates
  - Task 1: Find a similar known protein for a new protein

    h("APPGEAYLQV", ) → yes/no

  → – Task 2: Map new protein into known structure

    h("APPGEAYLQV", ) → [A→3,P→4,P→7,…]

  - Task 3: Refine structure

[Jo03, JoElGa05,YuJoEl06]

Linear Score Sequence Alignment

Method: Find alignment *y* that maximizes linear score

$$y = argmax_{y \in Y}\{score(x=(s,t), y)\}$$

Example:

– Sequences:

```
s=(A B C D)
t=(B A C C)
```

	A	B	C	D	-
A	10	0	-5	-10	-5
B	0	10	5	-10	-5
C	-5	5	10	-10	-5
D	-10	-10	-10	10	-5
-	-5	-5	-5	-5	-5

– Alignment y_1:

A	B	C	D
B	A	C	C

→ $score(x=(s,t),y_1) = 0+0+10-10 = 0$

– Alignment y_2:

-	A	B	C	D
B	A	C	C	-

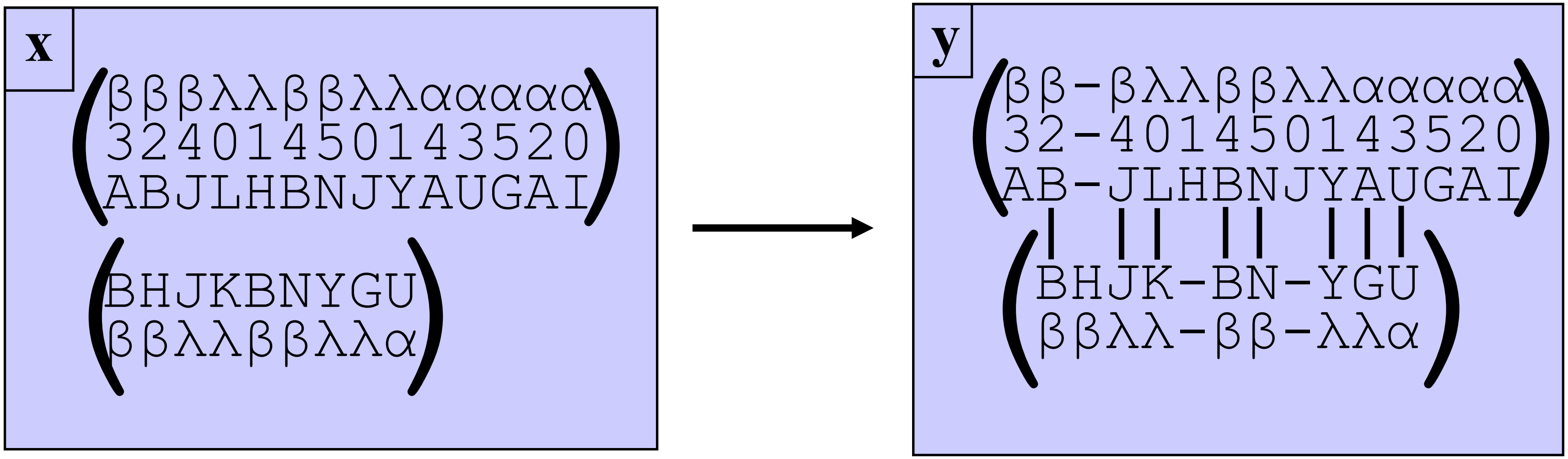
→ $score(x=(s,t),y_2) = -5+10+5+10-5 = 15$

Algorithm: Solve argmax via dynamic programming.

Predicting an Alignment

Protein Sequence to Structure Alignment (Threading)

- Given a pair *x=(s,t)* of new sequence *s* and known structure *t*, predict the alignment *y*.
- Elements of *s* and *t* are described by features, not just character identity.



```
x
(βββλλββλλααααα)
(32401450143520)
(ABJLHBNJYAUGAI)

(BHJKBNYGU)
(ββλλββλλα)

y
(ββ-βλλββλλααααα)
(32-401450143520)
(AB-JLHBNJYAUGAI)
  |  ||  ||  |||
( BHJK-BN-YGU   )
( ββλλ-ββ-λλα   )
```

[YuJoEl07]

Scoring Function for Vector Sequences

General form of linear scoring function:

$$\begin{aligned} score\,(\mathbf{x}{=}(\mathbf{s},\mathbf{t}),\mathbf{y}) &= \sum_i score(y_i^{\mathbf{s}}, y_i^{\mathbf{t}}) \\ &= \sum_i \mathbf{w}^T \phi(\mathbf{s},\mathbf{t},y_i) \\ &= \mathbf{w}^T \sum_i \phi(\mathbf{s},\mathbf{t},y_i) \\ &= \mathbf{w}^T \Phi(\mathbf{x},\mathbf{y}) \end{aligned}$$

→ match/gap score can be arbitrary linear function

→ argmax can still be computed efficiently via dynamic programming

Estimation:

– Generative estimation (e.g. log-odds, hidden Markov model)

– Discriminative estimation via structural SVM

[YuJoEl07]

Loss Function and Separation Oracle

- **Loss function:** $\Delta(y_i, y)$
  - Q loss: fraction of incorrect alignments
    - Correct alignment **y**=

–	A	B	C	D
B	A	C	C	–

    - Alternate alignment **y'**=

A	–	B	C	D
B	A	C	C	–

→ $\Delta_Q(y,y')=1/3$

  - Q4 loss: fraction of incorrect alignments outside window
    - Correct alignment **y**=

–	A	B	C	D
B	A	C	C	–

    - Alternate alignment **y'**=

A	–	B	C	D
B	A	C	C	–

→ $\Delta_{Q4}(y,y')=0/3$

- **Separation oracle:** $\hat{y} = argmax_{y \in Y}\{\Delta(y_i, y) + \vec{w}^T \Phi(x_i, y)\}$
  - Same dynamic programming algorithms as alignment

[YuJoEl07]

Experiment

- **Train set [Qiu & Elber]:**
  - 5119 structural alignments for training, 5169 structural alignments for validation of regularization parameter C
- **Test set:**
  - 29764 structural alignments from new deposits to PDB from June 2005 to June 2006.
  - All structural alignments produced by the program CE by superimposing the 3D coordinates of the proteins structures. All alignments have CE Z-score greater than 4.5.
- **Features (known for structure, SABLE predictions for sequence):**
  - Amino acid identity (A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y)
  - Secondary structure (α,β,λ)
  - Exposed surface area (0,1,2,3,4,5)

[YuJoEl07]

Experiment Results

Models:

- **Simple:** $\Phi(s,t,y_i)$ ⇔ (A|A; A|C; …;-|Y; α|α; α|β…; 0|0; 0|1;…)
- **Anova2:** $\Phi(s,t,y_i)$ ⇔ (Aα|Aα…; α0|α0…; A0|A0;…)
- **Tensor:** $\Phi(s,t,y_i)$ ⇔ (Aα0|Aα0; Aα0|Aα1; …)
- **Window:** $\Phi(s,t,y_i)$ ⇔ (AAA|AAA; …; ααααα|ααααα; …; 00000|00000;…)

Ability to train complex models?

Q-Score	# Features	Test
Simple	1020	39.89
Anova2	49634	44.98
Tensor	203280	42.81
Window	447016	46.30

Q-score when optimizing to Q-loss

Comparison against other methods?

Q4-score	Test
BLAST	28.44
SVM (Window)	70.71
SSALN [QiuElber]	67.30
TM-align [ZhaSko]	(85.32)

Q4-score when optimizing to Q4-loss

[YuJoEl07]

Overview

- **Task: Discriminative learning with complex outputs**
- **Related Work**
- **SVM algorithm for complex outputs**
  - Predict trees, sequences, equivalence relations, alignments
  - General non-linear loss functions
  - Generic formulation as convex quadratic program
- **Training algorithms**
  - n-slack vs. 1-slack formulation
  - Correctness and sparsity bound
- **Applications**
  - Sequence alignment for protein structure prediction [w/ Chun-Nam Yu]
  - ➡ Diversification of retrieval results in search engines [w/ Yisong Yue]
  - Supervised clustering [w/ Thomas Finley]
- **Conclusions**

Diversified Retrieval

- **Ambiguous queries:**
  - Example query: “SVM”
    - ML method
    - Service Master Company
    - Magazine
    - School of veterinary medicine
    - Sport Verein Meppen e.V.
    - SVM software
    - SVM books
  - “submodular” performance measure

    ➔ make sure each user gets at least one relevant result
- **Learning Queries:**
  - Find all information about a topic
  - Eliminate redundant information

Query: SVM

1. Kernel Machines
2. SVM book
3. SVM-light
4.
5.
6.
7.

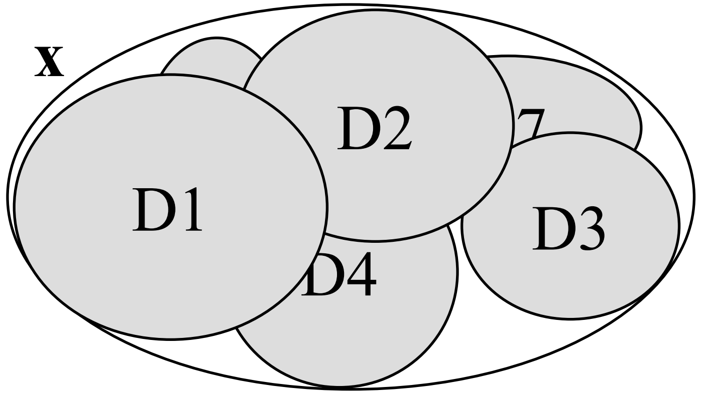
Query: SVM

1. Kernel Machines
2. Service Master Co
3. SV Meppen
4. UArizona Vet. Med.
5. SVM-light
6. Intro to SVM
7. …

[YueJo08]

Approach

- **Prediction Problem:**
  - Given set **x**, predict size *k* subset **y** that satisfies most users.
- **Approach: Topic Red. ¼ Word Red. [SwMaKi08]**



➔ **y** = { D1, D2, D3, D4 }

  - Weighted Max Coverage: $\mathbf{y} = \operatorname*{argmax}_{y\subset x,|y|=k} \left\{ \sum_{w\in\cup(y)} score(w) \right\}$
  - Greedy algorithm is 1-1/e approximation [Khuller et al 97]

➔ **Learn the benefit weights:** $score(w) = \mathbf{w}^T \phi(w,x)$

[YueJo08]

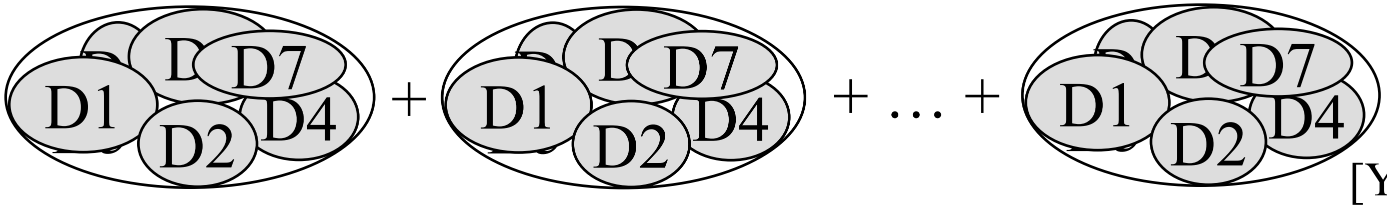
Features Describing Word Importance

- **How important is it to cover word w**
  - w occurs in at least X% of the documents in x
  - w occurs in at least X% of the titles of the documents in x
  - w is among the top 3 TFIDF words of X% of the documents in x
  - w is a verb

→ Each defines a feature in $\phi(w, x)$

- **How well a document d covers word w**
  - w occurs in d
  - w occurs at least k times in d
  - w occurs in the title of d
  - w is among the top k TFIDF words in d

→ Each defines a separate vocabulary and scoring function

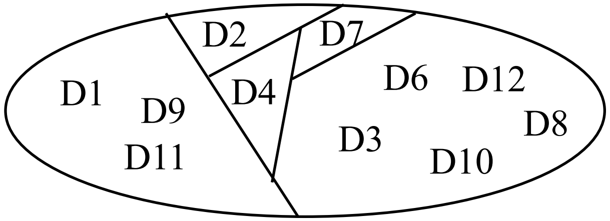


[YueJo08]

Loss Function and Separation Oracle

- **Loss function:** $\Delta(y_i, y)$
  - Popularity-weighted percentage of subtopics not covered in **y**

    →More costly to miss popular topics
  - Example:



[redacted]

- **Separation oracle:** $\hat{y} = argmax_{y \in Y}\{\Delta(y_i, y) + \vec{w}^T \Phi(x_i, y)\}$
  - Again a weighted max coverage problem

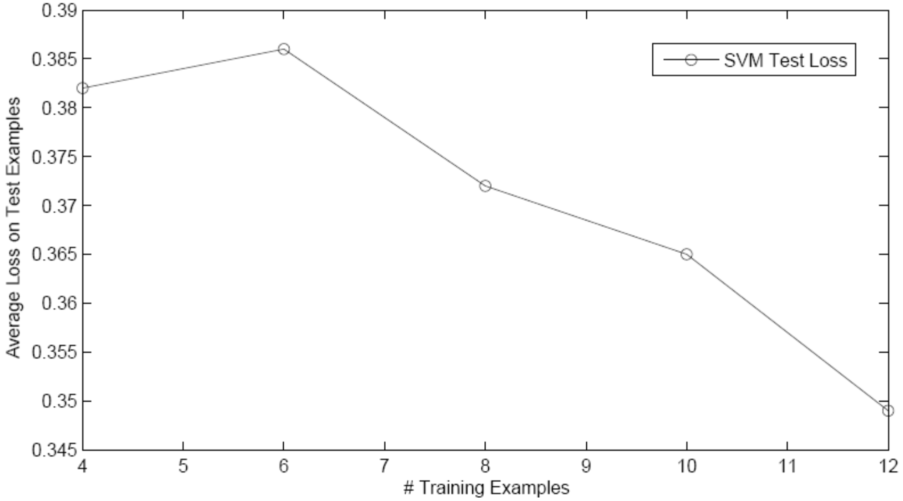
    → add artificial word for each subtopic with percentage weight
  - Greedy algorithm is 1-1/e approximation [Khuller et al 97]

[YueJo08]

Experiments

- **Data:**
  - TREC 6-8 Interactive Track
  - Relevant documents manually labeled by subtopic
  - 17 queries (~700 documents), 12/4/1 training/validation/test
  - Subset size k=5, two feature sets (div, div2)
- **Results:**

Method	Loss
Random	0.469
Okapi	0.472
Unweighted Model	0.471
Essential Pages	0.434
$\text{SVM}^{\Delta}_{div}$	0.349
$\text{SVM}^{\Delta}_{div2}$	0.382

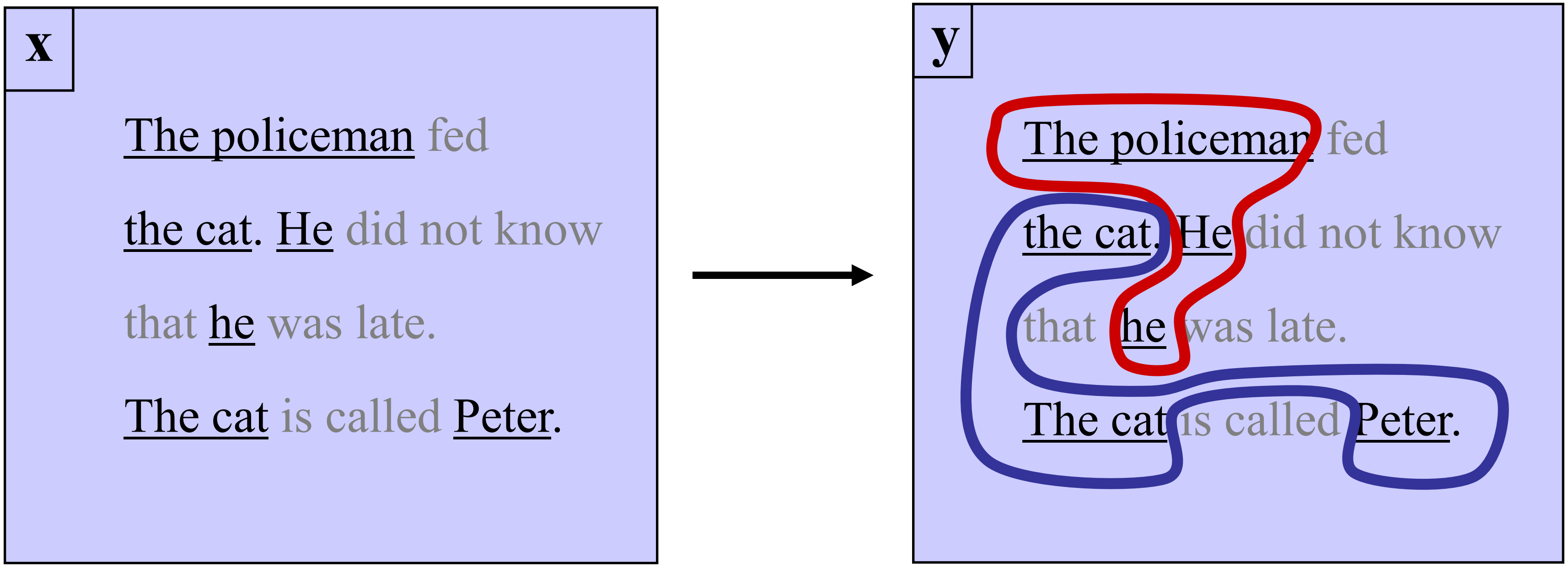


Overview

- **Task: Discriminative learning with complex outputs**
- **Related Work**
- **SVM algorithm for complex outputs**
  - Predict trees, sequences, equivalence relations, alignments
  - General non-linear loss functions
  - Generic formulation as convex quadratic program
- **Training algorithms**
  - n-slack vs. 1-slack formulation
  - Correctness and sparsity bound
- **Applications**
  - Sequence alignment for protein structure prediction [w/ Chun-Nam Yu]
  - Diversification of retrieval results in search engines [w/ Yisong Yue]
  - ➡ Supervised clustering [w/ Thomas Finley]
- **Conclusions**

Learning to Cluster

- **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.



x

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

y

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

Struct SVM for Supervised Clustering

- **Representation**
  - [redacted]
  - nd
  -
- **Los**
  - [redacted]
- **Predic**
  - [redacted]
  - NP orlica, 2003]
- **Find m**
  - $\hat{y}=$
  - NP orlica, 2003]

[redacted]

$\longrightarrow$

y						
1	1	1	1	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1
0	0	0	0	1	1	1

y						
1	1	1	1	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1
0	0	0	0	1	1	1

y'						
1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1
0	0	0	0	1	1	1

[FiJo05]

Summary and Conclusions

- **Learning to predict complex output**
  - Directly model machine learning application end-to-end
- **An SVM method for learning with complex outputs**
  - General method, algorithm, and theory
  - Plug in representation, loss function, and separation oracle
  - More details and further work:
    - Diversified retrieval [Yisong Yue, ICML08]
    - Sequence alignment [Chun-Nam Yu, RECOMB07, JCB08]
    - Supervised k-means clustering [Thomas Finley, forthcoming]
    - Approximate inference and separation oracle [Thomas Finley, ICML08]
    - Efficient kernelized structural SVMs [Chun-Nam Yu, KDD08]
- **Software: SVM$^{\text{struct}}$**
  - General API
  - Instances for sequence labeling, binary classification with non-linear loss, context-free grammars, diversified retrieval, sequence alignment, ranking
  - **`http://svmlight.joachims.org/`**