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 \( Y \)
  \[ h : X \rightarrow Y \]
  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.

The dog chased the cat

\[
\begin{array}{c}
x \quad \text{The dog chased the cat} \\
\end{array}
\]
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.

\[
\begin{align*}
x &= (s,t) \\
&= (ABJLHBNJYAU\text{GAI}) \\
t &= (BHJKB\text{N}Y\text{G}U)
\end{align*}
\]

\[
\begin{align*}
y &= (AB\text{---JLHBNJYAU\text{GAI}}) \\
&= (\text{BHJKB---N---YGU})
\end{align*}
\]
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)

\[ x \xrightarrow{\text{SVM}} y \]

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.

\[
\begin{align*}
\text{x} & : \begin{align*}
\text{The policeman} & \text{ fed } \\
\text{the cat. He did not know } & \\
\text{that he was late.} & \\
\text{The cat} & \text{ is called Peter.}
\end{align*} \\
\text{y} & : \begin{align*}
\text{The policeman} & \text{ fed } \\
\text{the cat. He did not know } & \\
\text{that he was late.} & \\
\text{The cat is called Peter.}
\end{align*}
\]
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?

- **Important applications for which conventional methods don't fit!**
  - Diversified retrieval [Carbonell & Goldstein 98] [Chen & Karger 06]
  - Directly optimize complex loss functions (e.g. F1, AvgPrec)

- **Direct modeling of problem instead of reduction!**
  - Noun-phrase co-reference: two step approach of pairwise classification and clustering as post-processing (e.g. [Ng & Cardie, 2002])

- **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

---

<table>
<thead>
<tr>
<th></th>
<th>Precision/Recall</th>
<th>Naïve Bayes</th>
<th>Linear SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break-Even Point</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuters</td>
<td>72.1</td>
<td>87.5</td>
<td></td>
</tr>
<tr>
<td>WebKB</td>
<td>82.0</td>
<td>90.3</td>
<td></td>
</tr>
<tr>
<td>Ohsumed</td>
<td>62.4</td>
<td>71.6</td>
<td></td>
</tr>
</tbody>
</table>
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 SVM

- Training Examples:
- Hypothesis Space: \( h(x) \)
- Training: Find hyperplane

Dual Opt. Problem:

Primal Opt. Problem:
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?

The dog chased the cat
Training: Find a hypothesis that solves the given problems efficiently.

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

The dog chased the cat
Joint Feature Map

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

$$h(\vec{x}) = \arg \max_{y \in Y} [\bar{w}^T \Phi(x, y)]$$

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

The dog chased the cat
Joint Feature Map for Trees

- **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 $h(\vec{x}) = \arg\max_{y \in Y} [\vec{w}^T \Phi(x, y)]$

**Problems**
- How to predict efficiently? 
- How to learn efficiently?
- Manageable number of parameters?
Structural Support Vector Machine

- Joint features describe the match between $x$ and $y$
- Learn weights so that $y$ is maximized for correct $y$

Hard-margin optimization problem:
Loss Functions: Soft-Margin Struct SVM

Soft-margin optimization problem:

Lemma: The training loss is upper bounded by

\[ Err\_S(h) = \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, h(x_i)) \leq \frac{1}{n} \sum_{i=1}^{n} \xi_i \]
Experiment: Natural Language Parsing

- **Implementation**
  - Incorporated modified version of Mark Johnson’s CKY parser
  - Learned weighted CFG with

- **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

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
</tr>
<tr>
<td>PCFG with MLE</td>
<td>55.2</td>
</tr>
<tr>
<td>SVM with $(1-F_1)$-Loss</td>
<td>58.9</td>
</tr>
</tbody>
</table>

- more complex features [TaKlCoKoMa04]
Generic Structural SVM

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

- **Prediction:**
  $$\hat{y} = \arg\max_{y \in Y} \{ \bar{w}^T \Phi(x,y) \}$$

- **Training:**

- **Applications:** Parsing, Sequence Alignment, Clustering, etc.
Reformulation of the Structural SVM QP

n-Slack Formulation: [TsoJoHoAl04]
Reformulation of the Structural SVM QP

n-Slack Formulation: [TsoJoHoAl04]

1-Slack Formulation: [JoFinYu08]
Cutting-Plane Algorithm for Structural SVM (1-Slack Formulation)

- **Input:**
- **REPEAT**
  - **FOR**
  - Compute
  - **ENDFOR**
  - **IF**
  - optimize StructSVM over $S$
  - **ENDIF**
- **UNTIL** $S$ 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-strategy formulation after adding at most

\[
\left\lfloor \log_2 \left( \frac{\Delta}{4R^2C} \right) \right\rfloor + \left\lfloor \frac{16R^2C}{\epsilon} \right\rfloor
\]

constraints to the working set \( S \), so that the primal constraints are feasible up to a precision \( \epsilon \) and the objective on \( S \) is optimal. The loss has to be bounded \( \Delta \), and \( R \).

[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
  – Representation \( \Phi(x, y) \)
  – Algorithms to compute
    \[
    \hat{y} = \arg \max_{y \in Y} \{w^T \Phi(x_i, y)\}
    \]
    \[
    \hat{y} = \arg \max_{y \in Y} \{\Delta(y_i, y) + w^T \Phi(x_i, y)\}
    \]

• 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
  \[ h(“APPGEAYLQV”) \rightarrow \]

• **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
    \(\rightarrow\) use known shapes as templates
  – Task 1: Find a similar known protein for a new protein
    \[ h(“APPGEAYLQV”, ) \rightarrow \text{yes/no} \]
  – Task 2: Map new protein into known structure
    \[ h(“APPGEAYLQV”, ) \rightarrow [A\rightarrow3,P\rightarrow4,P\rightarrow7,...] \]
  – Task 3: Refine structure

[Jo03, JoElGa05, YuJoEl06]
Linear Score Sequence Alignment

Method: Find alignment $y$ that maximizes linear score

$$y = \arg \max_{y \in \mathcal{Y}} \{ score(x=(s,t), y) \}$$

Example:

- Sequences:
  - $s = (A \ B \ C \ D)$
  - $t = (B \ A \ C \ C)$

- Alignment $y_1$:
  - $\begin{array}{cccc}
    A & B & C & D \\
    B & A & C & C \\
  \end{array}$
  $$\Rightarrow score(x=(s,t),y_1) = 0+0+10-10 = 0$$

- Alignment $y_2$:
  - $\begin{array}{cccc}
    - & A & B & C & D \\
    B & A & C & C & - \\
  \end{array}$
  $$\Rightarrow score(x=(s,t),y_2) = -5+10+5+10-5 = 15$$

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.

\[
\begin{align*}
  x &= (\beta\beta\lambda\lambda\beta\lambda\lambda\alpha\alpha\alpha\alpha \\
      &\quad 32401450143520 \\
      &\quad ABJLHBNNJYAUAGAI \\
      &\quad BHJKBNGUYGU \\
  y &= (\beta\beta-\beta\lambda\lambda\beta\lambda\lambda\alpha\alpha\alpha\alpha \\
      &\quad 32-401450143520 \\
      &\quad AB-\text{JLHBNNJYAUAGAI} \\
      &\quad BHJK-BN-YGU \\
      &\quad \beta\beta\lambda\lambda-\beta\beta-\lambda\lambda\alpha
\end{align*}
\]

[YuJoEi07]
Scoring Function for Vector Sequences

General form of linear scoring function:

$$\text{score} \left( x=(s, t), y \right) = \sum_i \text{score}(y_i^s, y_i^t)$$

$$= \sum_i w^T \phi(s, t, y_i)$$

$$= w^T \sum_i \phi(s, t, y_i)$$

$$= w^T \Phi(x, y)$$

→ 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 $y = \begin{array}{cccc} A & B & C & D \\ B & A & C & C & - \end{array}$  
    - Alternate alignment $y' = \begin{array}{cccc} A & - & B & C & D \\ B & A & C & C & - \end{array}$  
    - $\Delta_Q(y, y') = 1/3$
  - Alternate alignment $y' = \begin{array}{cccc} - & A & B & C & D \\ B & A & C & C & - \end{array}$  

- **Q4 loss:** fraction of incorrect alignments outside window
  - Correct alignment $y = \begin{array}{cccc} A & B & C & D \\ B & A & C & C & - \end{array}$  
    - $\Delta_Q(y, y') = 0/3$
  - Alternate alignment $y' = \begin{array}{cccc} A & - & B & C & D \\ B & A & C & C & - \end{array}$

- **Separation oracle:** $\hat{y} = \arg\max_{y \in Y} \{\Delta(y_i, y) + \bar{w}^T \Phi(x_i, y)\}$
  - 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):**
  - Secondary structure (α, β, λ)
  - Exposed surface area (0, 1, 2, 3, 4, 5)

[YuJoEl07]
Experiment Results

Models:
- **Simple**: $\Phi(s,t,y_i) \leftrightarrow (A|A; A|C; \ldots; |Y; \alpha|\alpha; \alpha|\beta; \ldots; 0|0; 0|1; \ldots)$
- **Anova2**: $\Phi(s,t,y_i) \leftrightarrow (A\alpha|A\alpha; \ldots; \alpha0|\alpha0; \ldots; A0|A0; \ldots)$
- **Tensor**: $\Phi(s,t,y_i) \leftrightarrow (A\alpha0|A\alpha0; A\alpha0|A\alpha1; \ldots)$
- **Window**: $\Phi(s,t,y_i) \leftrightarrow (AAA|AAA; \ldots; \alpha\alpha\alpha\alpha|\alpha\alpha\alpha\alpha; \ldots; 00000|00000; \ldots)$

### Ability to train complex models?

<table>
<thead>
<tr>
<th>Q-Score</th>
<th># Features</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>1020</td>
<td>39.89</td>
</tr>
<tr>
<td>Anova2</td>
<td>49634</td>
<td>44.98</td>
</tr>
<tr>
<td>Tensor</td>
<td>203280</td>
<td>42.81</td>
</tr>
<tr>
<td>Window</td>
<td>447016</td>
<td>46.30</td>
</tr>
</tbody>
</table>

Q-score when optimizing to Q-loss

### Comparison against other methods?

<table>
<thead>
<tr>
<th>Q4-score</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST</td>
<td>28.44</td>
</tr>
<tr>
<td>SVM (Window)</td>
<td>70.71</td>
</tr>
<tr>
<td>SSALN [QiuElber]</td>
<td>67.30</td>
</tr>
<tr>
<td>TM-align [ZhaSko]</td>
<td>(85.32)</td>
</tr>
</tbody>
</table>

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. Query: SVM
   1. Kernel Machines
   2. Service Master Co
   3. SV Meppen
   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: 
  \[ y = \arg\max_{y \subset x, |y| = k} \left\{ \sum_{w \in \cup(y)} \text{score}(w) \right\} \]

- Greedy algorithm is 1-1/e approximation [Khuller et al 97]

⇒ **Learn the benefit weights:** 
  \[ \text{score}(w) = w^T \phi(w, x) \]

[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
  $\rightarrow$ 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
  $\rightarrow$ 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 $y$
    $\rightarrow$ More costly to miss popular topics
  - Example:

  ![Diagram showing subtopics D1 to D12]

• **Separation oracle:** $\hat{y} = \arg\max_{y \in Y} \{\Delta(y_i, y) + \bar{w}^T \Phi(x_i, y)\}$
  - Again a weighted max coverage problem
    $\rightarrow$ add artificial word for each subtopic with percentage weight
  - Greedy algorithm is $1-1/e$ approximation [Khuller et al 97]
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:

<table>
<thead>
<tr>
<th>Method</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.469</td>
</tr>
<tr>
<td>Okapi</td>
<td>0.472</td>
</tr>
<tr>
<td>Unweighted Model</td>
<td>0.471</td>
</tr>
<tr>
<td>Essential Pages</td>
<td>0.434</td>
</tr>
<tr>
<td>$\text{SVM}^\Delta_{\text{div}}$</td>
<td>0.349</td>
</tr>
<tr>
<td>$\text{SVM}^\Delta_{\text{div2}}$</td>
<td>0.382</td>
</tr>
</tbody>
</table>

![Graph showing average test loss vs. number of training examples.]
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.

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.
Struct SVM for Supervised Clustering

- **Representation**
  - $y$ is reflexive ($y_{ii} = 1$), symmetric ($y_{ij} = y_{ji}$), and transitive (if $y_{ij} = 1$ and $y_{jk} = 1$, then $y_{ik} = 1$)

- **Joint feature map**

- **Loss Function**

- **Prediction**
  - NP hard, use linear relaxation instead [Demaine & Immorlica, 2003]

- **Find most violated constraint**
  - NP hard, use linear relaxation instead [Demaine & Immorlica, 2003]

- [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\textsuperscript{struct}
  – General API
  – Instances for sequence labeling, binary classification with non-linear loss, context-free grammars, diversified retrieval, sequence alignment, ranking
  – \url{http://svmlight.joachims.org/}