Large Margin Methods for Structured and Interdependent Output Variables

I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun
Journal of Machine Learning Research (JMLR), 6(Sep):1453-1484, 2005.

Presented by
Thorsten Joachims

Cornell University
Department of Computer Science

Goal of Paper

- **Learn function**

$$h : X \longrightarrow Y$$

where X is the input space and Y is some structured output space (e.g. sequences, trees, equivalence relations).
- **Paper proposes Support Vector Machine (SVM) method**
  - that applies to a large class of structured outputs Y
    - Sequences (i.e. Hidden Markov Models)
    - Trees (i.e. Weighted Context-Free Grammars)
    - Hierarchical classification
    - Sequence alignment (i.e. Edit-Distance cost function)
  - allows the use of fairly general loss functions
  - is a generalization of multi-class SVMs
  - has polynomial time training algorithm.

Examples of Complex Output Spaces

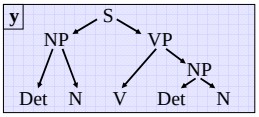
- **Part-of-Speech Tagging**
  - Given a sequence of words *x*, predict sequence of tags *y*.
  - Dependencies from tag-tag transitions in Markov model.

x: The bear chased the cat → y: Det N V Det N

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.

x: The dog chased the cat →



Structured Output Prediction as Multi-Class Classification

- **Learning Task: *P(X,Y) = P(X) P(Y|X)***
  - Input Space: *X* (i.e. feature vectors, word sequence, etc.)
  - Output Space: *Y* (i.e. class, tag sequence, parse tree, etc.)
  - Training Data: $S=((x_1,y_1), \ldots, (x_n,y_n)) \sim_{iid} P(X,Y)$
- **Approach: view as multi-class classification task**
  - Every complex output $y \in Y$ is one class
- **Goal: Find *h: X → Y* with low expected loss**
  - Loss function: $\Delta(y,y')$ (penalty for predicting *y'* if *y* correct)
  - Expected loss (i.e. risk, prediction error):

$$Err_P(h) = \sum_{x,y} \Delta(y, h(x))\ P(X=x, Y=y)$$

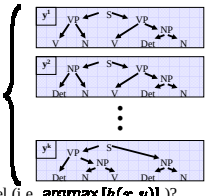
Natural Language Parsing as Multi-Class Classifications

- **Approach: view as multi-class classification task**
  - Every complex output $y \in Y$ is one class
  - Learn discriminant function

$$h(x) = \operatorname{argmax}_{y \in Y} [h(x,y)]$$

via empirical risk minimization.

x: The dog chased the cat →

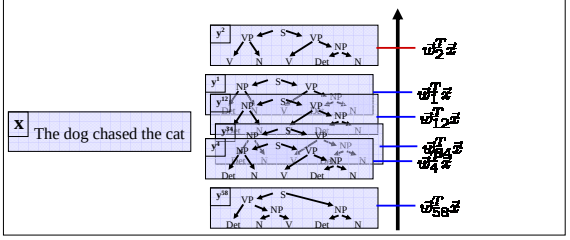


- **Challenges: exponentially many classes**
  - How to compactly represent model?
  - How to do efficient prediction with model (i.e. $\operatorname{argmax}_{y \in Y} [h(x,y)]$)?
  - How to effectively estimate model from data? (e.g. compute $h_w^* = \operatorname{argmin}_{h_w \in H_\Omega} \sum_{i=1}^{n} \Delta(y_i, h_w(x_i))$)

Multi-Class Linear Discriminant

- **Linear discriminant function of the form:** $h_w(x,y) = w_y^T x$
- **Learn one weight vector w_y for each class $y \in Y$**

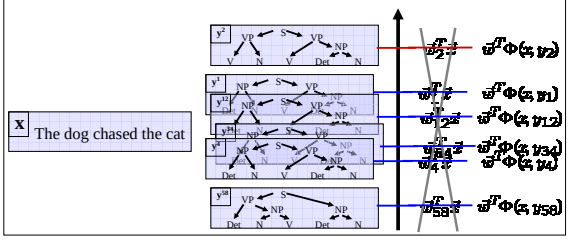
$$h(\vec{x}) = argmax_{y \in Y} \left[w_y^T x \right]$$



Joint Feature/Kernel Map

- **Feature vector $\Phi(x,y)$ that describes match between x and y**
- **Linear discriminant function of the form:** $h_w(x,y) = \vec{w}^T \Phi(x,y)$

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

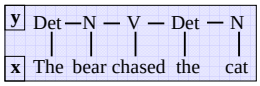


Joint Feature Map for Sequences

- **Linear Chain Model (HMM)**
  - Only local dependencies
  - Score for each adjacent label/label and word/label pair
  - Find highest scoring sequence

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

Viterbi

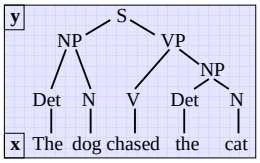


$$\Phi(\mathbf{x},\mathbf{y}) = \begin{pmatrix} 1 & N \rightarrow V \\ 0 & Det \rightarrow V \\ 2 & Det \rightarrow N \\ 1 & V \rightarrow Det \\ \vdots & \\ 0 & Det \rightarrow bear \\ 2 & Det \rightarrow the \\ 1 & N \rightarrow bear \\ 1 & V \rightarrow chased \\ 1 & N \rightarrow cat \end{pmatrix}$$

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

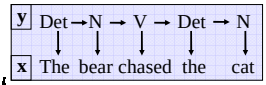
CKY Parser



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

Connection to Graphical Models

Hidden Markov Model:



– Assumptions

$$P(Y = (y^{(1)},..,y^{(l)})) = \prod_{i=1}^{l} P(Y_c = y^{(i)} | Y_p = y^{(i-1)})$$

$$P(X = (x^{(1)},..,x^{(l)}) | Y = (y^{(1)},..,y^{(l)})) = \prod_{i=1}^{l} P(X_c = x^{(i)} | Y_c = y^{(i)})$$

$\rightarrow$ Rule: $h(x) = \underset{y \in Y}{\text{argmax}} [P(X=x|Y=y)P(Y=y)]$

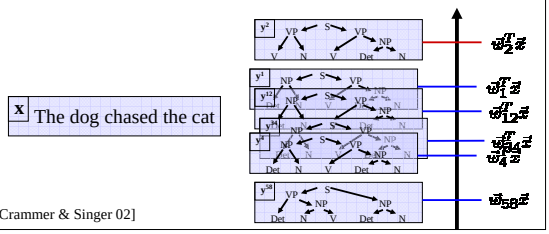
$$= \underset{(y^{(1)},..,y^{(l)}) \in Y}{\text{argmax}} \left[\prod_{i=1}^{l} P(Y_c = y^{(i)} | Y_p = y^{(i-1)}) P(X_c = x^{(i)} | Y_c = y^{(i)}) \right]$$

$$= \underset{(y^{(1)},..,y^{(l)}) \in Y}{\text{argmax}} \left[\vec{w}^T \Phi(x,y) \right]$$

with $w_{ab} = -log[P(Y_c=a|Y_p=b)]$ and $w_{cd} = -log[P(X_c=c|Y_c=d)]$
and $\Phi(x,y)$ histogram

Multi-Class SVM

- **Training Examples:** $(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n)$ $\vec{x} \in \Re^N$ $y \in \{1, ..., k\}$
- **Hypothesis Space:** $h(\vec{x}) = argmax_{i \in \{1,...,k\}} \left[\vec{w}_i^T \vec{x} \right]$



[Crammer & Singer 02]

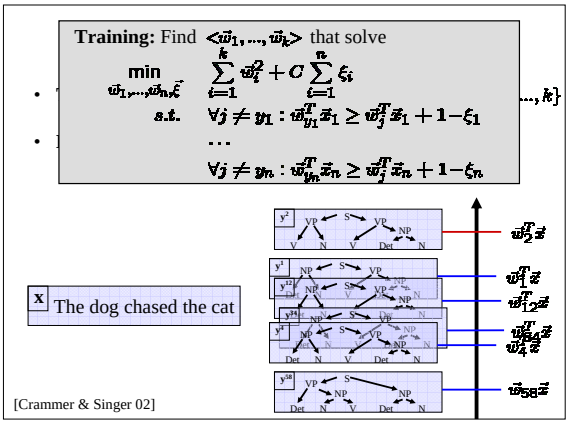
Training: Find $<\vec{w}_1,\ldots,\vec{w}_k>$ that solve

$$\min_{\vec{w}_1,\ldots,\vec{w}_k,\vec{\xi}} \sum_{i=1}^{k} \vec{w}_i^2 + C\sum_{i=1}^{n}\xi_i$$

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

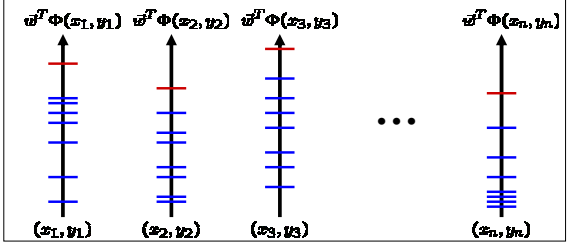
$$\ldots$$

$$\forall j \neq y_n : \vec{w}_{y_n}^T\vec{x}_n \geq \vec{w}_j^T\vec{x}_n + 1 - \xi_n$$



Structural Support Vector Machine

- **Joint features $\Phi(x,y)$ describe match between x and y**
- **Learn weights $\vec{w}$ so that $\vec{w}^T\Phi(x,y)$ is max for correct y**



Structural Support Vector Machine

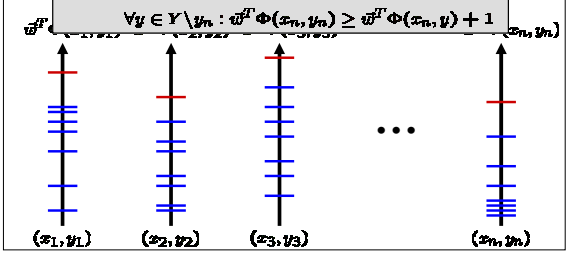
Hard-margin optimization problem:

$$\min_{\vec{w}} \frac{1}{2}\vec{w}^T\vec{w}$$

$$s.t.\quad \forall y \in Y\setminus y_1 : \vec{w}^T\Phi(x_1,y_1) \geq \vec{w}^T\Phi(x_1,y) + 1$$

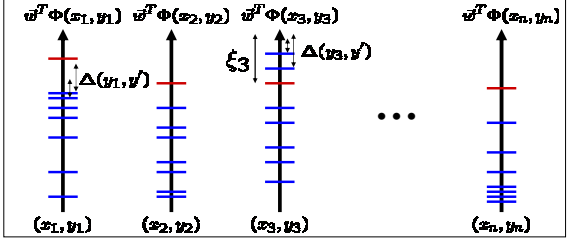
$$\ldots$$

$$\forall y \in Y\setminus y_n : \vec{w}^T\Phi(x_n,y_n) \geq \vec{w}^T\Phi(x_n,y) + 1$$



Soft-Margin Struct SVM (Margin Rescaling)

- **Loss function $\Delta(y_i,y)$ measures match between target and prediction.**



Soft-Margin Struct SVM (Margin Rescaling)

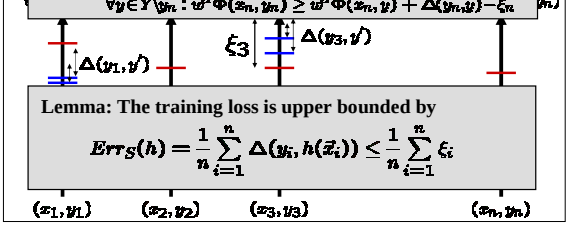
Soft-margin optimization problem:

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

$$s.t.\quad \forall y \in Y\setminus y_1 : \vec{w}^T\Phi(x_1,y_1) \geq \vec{w}^T\Phi(x_1,y) + \Delta(y_1,y) - \xi_1$$

$$\ldots$$

$$\forall y \in Y\setminus y_n : \vec{w}^T\Phi(x_n,y_n) \geq \vec{w}^T\Phi(x_n,y) + \Delta(y_n,y) - \xi_n$$

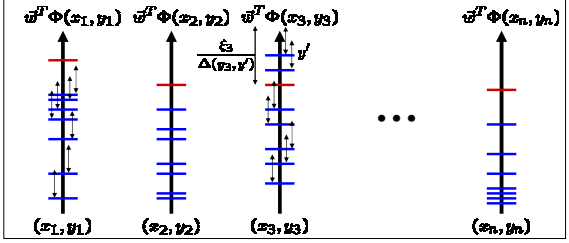


Lemma: The training loss is upper bounded by

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

Soft-Margin Struct SVM (Slack Rescaling)

- **Loss function $\Delta(y_i,y)$ measures match between target and prediction.**



Soft-Margin Struct SVM (Slack Rescaling)

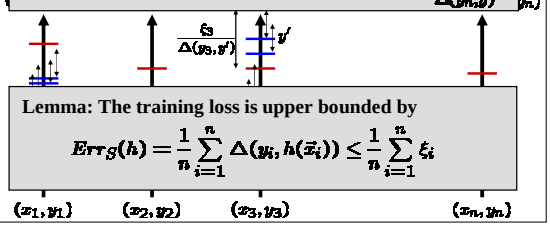
Soft-margin optimization problem:

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

$$\text{s.t.} \quad \forall y \in Y \setminus y_1 : \vec{w}^T\Phi(x_1,y_1) \geq \vec{w}^T\Phi(x_1,y) + 1 - \frac{\xi_1}{\Delta(y_1,y)}$$

$$\ldots$$

$$\forall y \in Y \setminus y_n : \vec{w}^T\Phi(x_n,y_n) \geq \vec{w}^T\Phi(x_n,y) + 1 - \frac{\xi_n}{\Delta(y_n,y)}$$



Lemma: The training loss is upper bounded by

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

Cutting-Plane Algorithm for Structural SVM

- **Input:** $(x_1,y_1),\ldots,(x_n,y_n),C,\epsilon$
- $S \leftarrow \emptyset, \vec{w} \leftarrow 0, \vec{\xi} \leftarrow 0$
- **REPEAT**
  - FOR $i = 1,\ldots,n$
    - compute $\hat{y} = argmax_{y \in Y}\{\Delta(y_i,y) + \vec{w}^T\Phi(x_i,y)\}$ (Find most violated constraint)
    - IF $(\Delta(y_i,\hat{y}) - \vec{w}^T[\Phi(x_i,y_i) - \Phi(x_i,\hat{y})] > \xi_i + \epsilon$ (Violated by more than ε ?)
      - $S \leftarrow S \cup \{\vec{w}^T[\Phi(x_i,y_i) - \Phi(x_i,\hat{y})] \geq \Delta(y_i,\hat{y}) - \xi_i\}$ (Add constraint to working set)
      - $[\vec{w},\vec{\xi}] \leftarrow$ optimize StructSVM over S
    - ENDIF
  - ENDFOR
- **UNTIL** S **has not changed during iteration**

[AltHo03] [Jo03] [TsoJoHoAl05]

Polynomial Sparsity Bound

- **Theorem:** The sparse-approximation algorithm finds a solution to the soft-margin optimization problem after adding at most

$$\max\left\{\frac{2nA}{\epsilon}, \frac{8nCAR^2}{\epsilon^2}\right\}$$

constraints to the working set S, so that the Kuhn-Tucker conditions are fulfilled up to a precision ϵ. The loss has to be bounded $0 \leq \Delta(y_i,y) \leq A$, and $\|\Phi(x,y)\| \leq R$.

[Jo 03] [Tsochantaridis et al. 04] [Tsochantaridis et al. 05]

Experiment: Natural Language Parsing

- **Implemention**
  - Implemented Sparse-Approximation Algorithm in SVM^{light}
  - 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

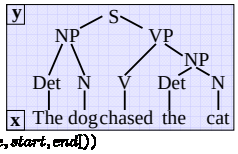
	Test Accuracy		Training Efficiency		
Method	Acc	F_1	CPU-h	Iter	Const
PCFG with MLE	55.2	86.0	0	N/A	N/A
SVM with (1-F_1)-Loss	**58.9**	**88.5**	3.4	12	8043

[Tsochantaridis et al. 05]

More Expressive Features

- **Linear composition:**

$$\Phi(x,y) = \sum_{i=1}^{l}\phi(x,y_i)$$



- **General form:**

$$\phi(x,y_i) = \phi_{kernel}(\phi(x,[rule,start,end]))$$

$$K(a,b) = \phi_{kernel}(a)^T\phi_{kernel}(a)$$

- **So far:**

$$\phi(x,y_i) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \ldots \\ 0 \end{pmatrix} \text{ if } rule(y_i) = \text{'S} \leftarrow \text{NP VP'}$$

- **Example:**

$$\phi(x,y_i) = \begin{pmatrix} 1 & \text{if } y_i = \text{NP} \wedge x_{end} = \text{'.'} \\ (start-end)^2 & \text{if } y_i = NP \\ 1 & \text{if } y_i = S \wedge \text{span contains } x_t = \text{'and'} \\ \ldots \end{pmatrix}$$

see [Taskar et al. 05]

Applying Structural SVM to New Problem

- **Application specific**
  - Loss function $\Delta(y_i,y)$
  - Representation $\Phi(x,y)$
  - Algorithms to compute

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

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

- **Implementation SVM-struct:** http://svmlight.joachims.org
  - Context-free grammars
  - Sequence alignment
  - Classification with multivariate loss (e.g. F1, ROC Area)
  - General API for other problems

Summary

- **Support Vector Machine approach to training**
  - Hidden Markov Models
  - Weighted Context-Free Grammars
  - Sequence Alignment cost functions
  - Etc.
- **Incorporate loss functions via**
  - Margin rescaling
  - Slack rescaling
- **General training algorithm based on cutting-plane method**
  - Efficient for all linear discriminant models where argmax efficient