Structured Output Prediction: Discriminative Learning

CS6780 – Advanced Machine Learning
Spring 2019

Thorsten Joachims
Cornell University

Reading:
Murphy 19.7, 19.6

Structured Output Prediction

- Supervised Learning from Examples
  - Find function from input space X to output space Y

$$h: X \to Y$$

    such that the prediction error is low.
- Typical
  - Output space is just a single number
    - Classification: -1,+1
    - Regression: some real number
- General
  - Predict outputs that are complex objects

Idea for Discriminative Training of HMM

Idea:

– $h_{bayes}(x) = argmax_{y \in \text{Y}} [P(Y = y|X = x)]$
  $= argmax_{y \in \text{Y}} [P(X = x|Y = y)P(Y = y)]$

– Model $P(Y = y|X = x)$ with $\vec{w} \cdot \phi(x, y)$ so that
  $\left(argmax_{y \in \text{Y}} [P(Y = y|X = x)]\right) = \left(argmax_{y \in \text{Y}} [\vec{w} \cdot \phi(x, y)]\right)$

Hypothesis Space:

h(x) = $argmax_{y \in \text{Y}} [\vec{w} \cdot \phi(x, y)]$ with $\vec{w} \in \Re^N$

Intuition:

– Tune $\vec{w}$ so that correct y has the highest value of $\vec{w} \cdot \phi(x, y)$

– $\phi(x, y)$ is a feature vector that describes the match between x and y

Training HMMs with Structural SVM

- HMM

$$P(x,y) = P(y_1)P(x_1|y_1)\prod_{i=2}^{l} P(x_i|y_i)P(y_i|y_{i-1})$$

$$\log P(x,y) = logP(y_1) + logP(x_1|y_1) + \sum_{i=2}^{l} logP(x_i|y_i) + logP(y_i|y_{i-1})$$

- Define $\phi(x,y)$ so that model is isomorphic to HMM
  - One feature for each possible start state
  - One feature for each possible transition
  - One feature for each possible output in each possible state
  - Feature values are counts

Joint Feature Map for Sequences

- Linear Chain HMM
  - Each transition and emission has a weight
  - Score of a sequence is the sum of its weights
  - Find highest scoring sequence h(x) = $argmax_{y \in Y} \, [\vec{w} \cdot \phi(x, y)]$

Viterbi

x The dog chased the cat

$$\downarrow$$

y

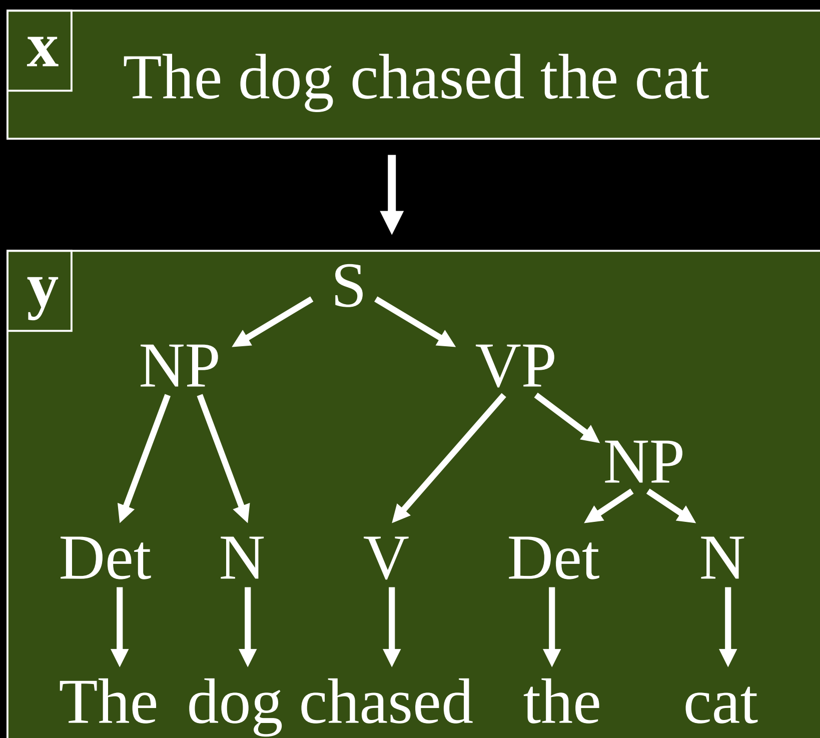
$$\begin{tikzcd}
\text{Det} \arrow[r] \arrow[d] & \text{N} \arrow[r] \arrow[d] & \text{V} \arrow[r] \arrow[d] & \text{Det} \arrow[r] \arrow[d] & \text{N} \arrow[d] \\
\text{The} & \text{dog} & \text{chased} & \text{the} & \text{cat}
\end{tikzcd}$$

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

Joint Feature Map for Trees

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

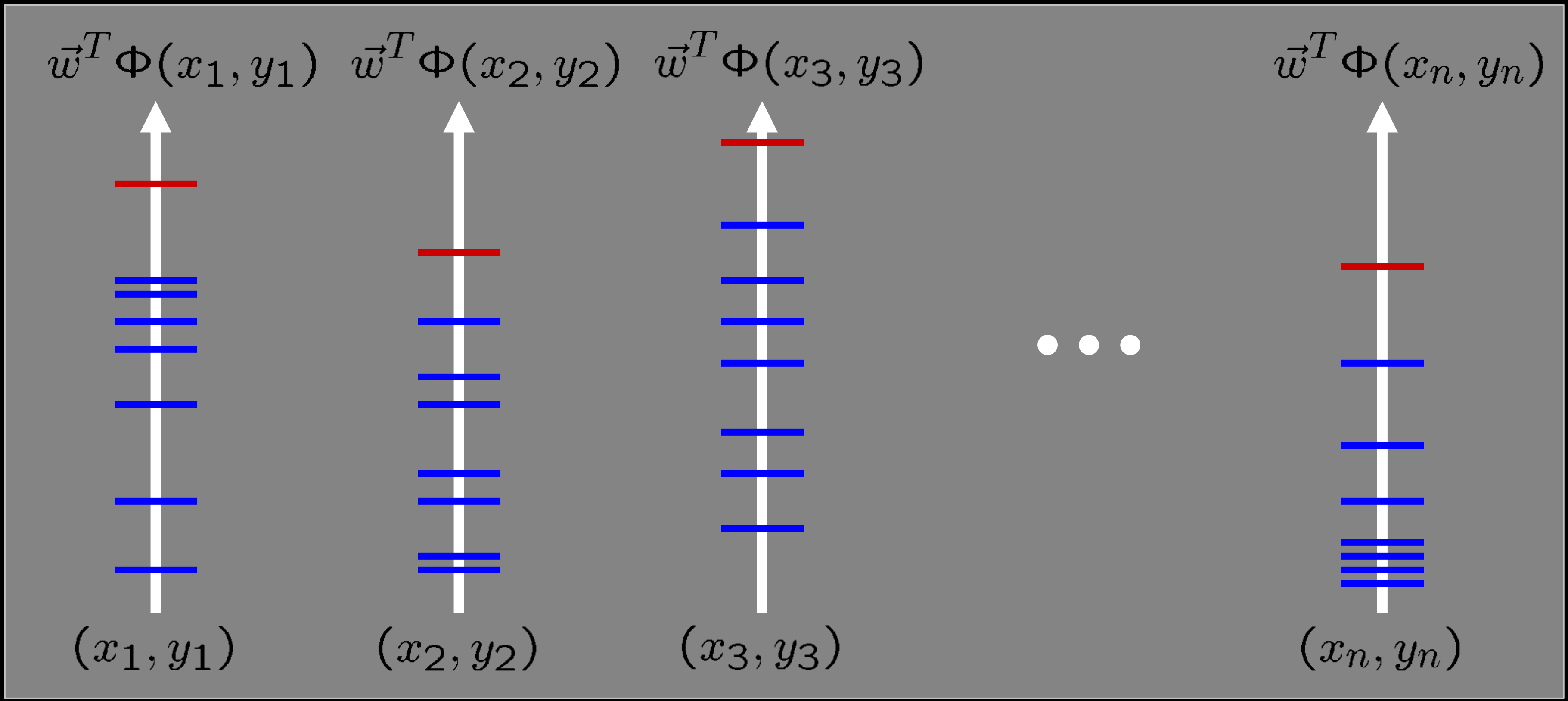
CKY Parser



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

Structural Support Vector Machine

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



Structural SVM Training Problem

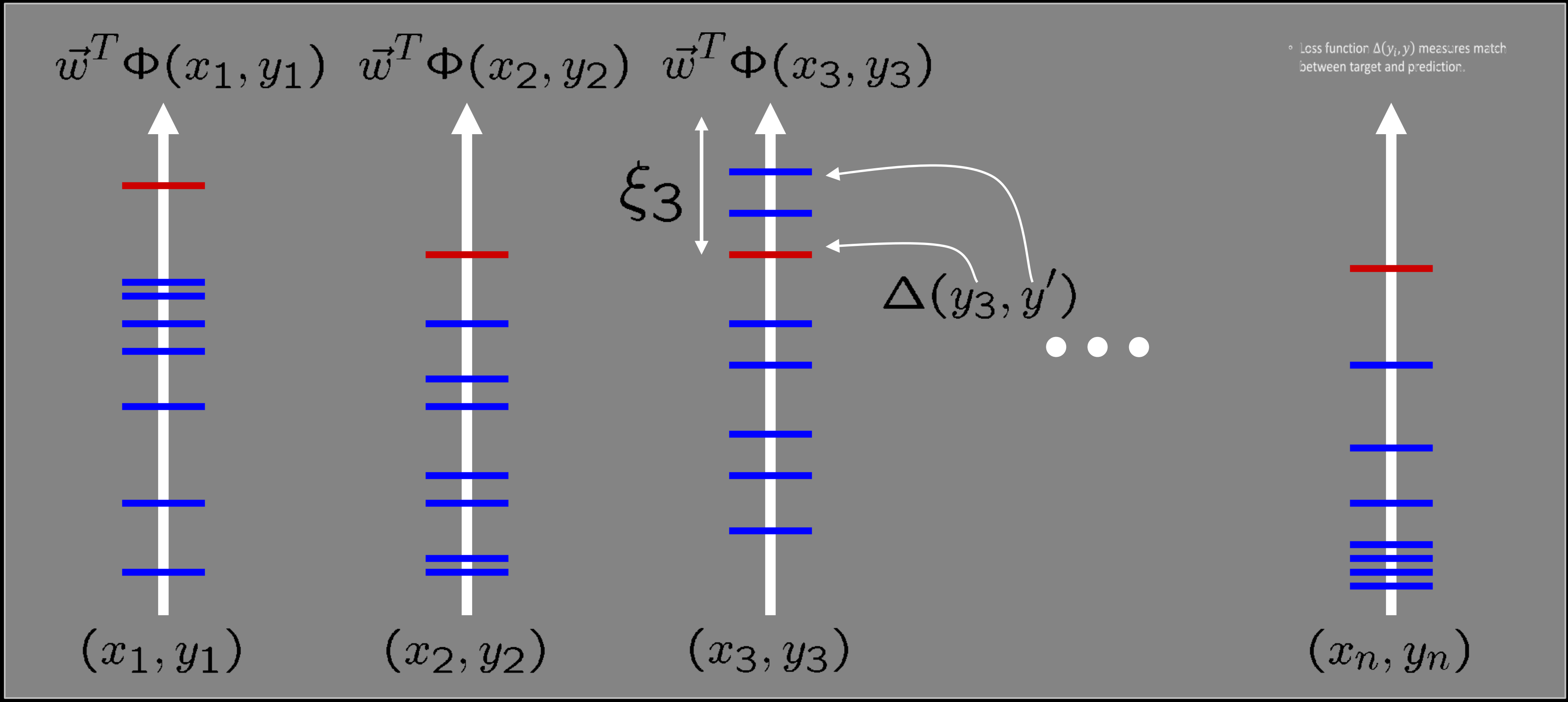
Hard-margin optimization problem:

$$
\begin{aligned}
\min_{\vec{w}} \quad & \frac{1}{2}\vec{w}^T\vec{w} \\
s.t. \quad & \forall y \in Y \setminus y_1 : \vec{w}^T\Phi(x_1, y_1) \geq \vec{w}^T\Phi(x_1, y) + 1 \\
& \dots \\
& \forall y \in Y \setminus y_n : \vec{w}^T\Phi(x_n, y_n) \geq \vec{w}^T\Phi(x_n, y) + 1
\end{aligned}
$$

- Training Set: $(x_1, y_1), \dots, (x_n, y_n)$
- Prediction Rule: $\mathrm{h}_{svm}(x) = argmax_{y \in \mathrm{Y}} \, [\vec{w} \cdot \phi(x, y)]$
- Optimization:
  - Correct label y_i must have higher value of $\vec{w} \cdot \phi(x, y)$ than any incorrect label y
  - Find weight vector with smallest norm

Soft-Margin Structural SVM

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



Soft-Margin Structural SVM

Soft-margin optimization problem:

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

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

$$\ldots$$

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

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

- 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.

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 $\hat{y} = argmax_{y \in Y} \{\Delta(y_i, y) + \vec{w}^T \Phi(x_i, y)\}$
    - IF $(\Delta(y_i, \hat{y}) - \vec{w}^T[\Phi(x_i, y_i) - \Phi(x_i, \hat{y})]) > \xi_i + \epsilon$
      - $S \leftarrow S \cup \{\vec{w}^T[\Phi(x_i, y_i) - \Phi(x_i, \hat{y})] \geq \Delta(y_i, \hat{y}) - \xi_i\}$
      - $[\vec{w}, \vec{\xi}] \leftarrow$ optimize StructSVM over S
    - ENDIF
  - ENDFOR
- UNTIL S has not changed during iteration

Find most violated constraint

Violated by more than ε ?

Add constraint to working set

Polynomial Sparsity Bound

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

$$n \frac{4CA^2R^2}{\epsilon^2}$$

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

More Expressive Features

- Linear composition: $\Phi(x,y) = \sum \phi(x, y_j)$
- So far: $\phi(x, y_i) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad if\ y_i =' S \to NP\ VP'$
- General: $\phi(x, y_i) = \phi_{kernel}\big(\phi(x, [rule, start, end])\big)$
- Example: $\phi(x, y_i) =$

$$\begin{pmatrix} 1 \\ (start - end)^2 \\ 1 \\ \vdots \end{pmatrix} \begin{matrix} if\ x_{start} = \text{"while and x}_{end}\text{="."} \\ \\ span\ contains\ \text{"}and\text{"} \end{matrix}$$

Applying StructSVM to New Problem

- Basic algorithm implemented in SVM-struct
  - http://svmlight.joachims.org
- Application specific
  - Loss function $\Delta(y_i, y)$
  - Representation $\Phi(x, y)$
  - Algorithms to compute
    - $\hat{y} = \underset{y \in \mathrm{Y}}{\operatorname{argmax}} \; [w \cdot \Phi(x, y)]$
    - $\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \; [\Delta(y_i, y) + w \cdot \Phi(x, y)]$

→ Generic structure covers OMM, MPD, Finite-State Transducers, MRF, etc.

Conditional Random Fields (CRF)

- Model:
  - $P(y|x,w) = \frac{\exp(w \cdot \Phi(x,y))}{\sum_{y'} \exp(w \cdot \Phi(x,y'))}$
  - $P(w) = N(w|0, \lambda I)$
- Conditional MAP training:

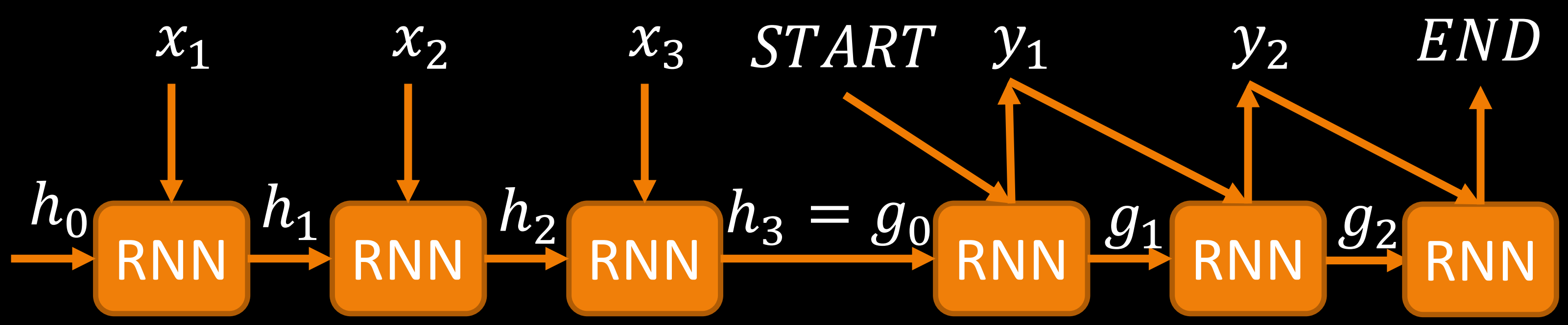
$$\widehat{w} = \underset{w}{\operatorname{argmax}}\left[-w \cdot w + \lambda \sum_i \log\left(P(y_i|x_i, w)\right)\right]$$

- Prediction for zero/one loss:

$$\hat{y} = \underset{y}{\operatorname{argmax}}[w \cdot \Phi(x, y)]$$

Encoder/Decoder Networks

- Encoder: Build fixed-size representation of input sequence x.
- Decoder: Generate output sequence y from encoder output.



$$h_t = h(W_h h_{t-1} + V_h x_t)$$

$$g_t = g(W_g g_{t-1} + V_g y_{t-1})$$

$$p = f(V_f g_t)$$