Structured Output Prediction: Discriminative Learning

CS6780 – Advanced Machine Learning
Spring 2015

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 $Y$

\[ h: X \rightarrow Y \]

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:

- \( h_{\text{bayes}}(x) = \arg \max_{y \in Y} [P(Y = y | X = x)] \)
  
  \[ = \arg \max_{y \in Y} [P(X = x | Y = y)P(Y = y)] \]
- Model \( P(Y = y | X = x) \) with \( \vec{w} \cdot \phi(x, y) \) so that
  \[ (\arg \max_{y \in Y} [P(Y = y | X = x)]) = (\arg \max_{y \in Y} [\vec{w} \cdot \phi(x, y)]) \]

Hypothesis Space:

\[ h(x) = \arg \max_{y \in Y} [\vec{w} \cdot \phi(x, y)] \text{ with } \vec{w} \in \mathbb{R}^N \]

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

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

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

- 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 $h(x) = \arg\max_{y \in Y} [\bar{w} \cdot \phi(x, y)]$

\[
\Phi(x, y) = \begin{pmatrix}
2 & Det \rightarrow N \\
0 & Det \rightarrow V \\
1 & N \rightarrow V \\
1 & V \rightarrow Det \\
\vdots
\end{pmatrix}
\]

- The dog chased the cat
- 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
  – Score of a tree is the sum of its weights
  – Find highest scoring tree \( h(x) = \arg\max_{y \in Y} [\vec{w} \cdot \phi(x, y)] \)

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

The dog chased the cat
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 $y$

\[
\begin{align*}
\vec{w}^T \Phi(x_1, y_1) & \quad \vec{w}^T \Phi(x_2, y_2) & \quad \vec{w}^T \Phi(x_3, y_3) & \quad \vec{w}^T \Phi(x_n, y_n)
\end{align*}
\]
Structural SVM Training Problem

Hard-margin optimization problem:

\[
\begin{align*}
\min_{\vec{w}} & \quad \frac{1}{2} \vec{w}^T \vec{w} \\
\text{s.t.} & \quad \forall y \in Y \setminus y_1 : \vec{w}^T \Phi(x_1, y) \geq \vec{w}^T \Phi(x_1, y_1) + 1 \\
& \quad \ldots \\
& \quad \forall y \in Y \setminus y_n : \vec{w}^T \Phi(x_n, y) \geq \vec{w}^T \Phi(x_n, y_1) + 1
\end{align*}
\]

- Training Set: \((x_1, y_1), \ldots, (x_n, y_n)\)
- Prediction Rule: \(h_{svm}(x) = \arg \max_{y \in Y} [\vec{w} \cdot \phi(x, y)]\)
- 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.

\[ \begin{align*}
\mathbf{w}^T \Phi(x_1, y_1) & \quad \mathbf{w}^T \Phi(x_2, y_2) & \quad \mathbf{w}^T \Phi(x_3, y_3) \\
(x_1, y_1) & \quad (x_2, y_2) & \quad (x_3, y_3) \\
\end{align*} \]
Soft-Margin Structural SVM

Soft-margin optimization problem:

\[
\min_{\vec{w}, \xi} \quad \frac{1}{2} \vec{w}^T \vec{w} + C \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad \forall y \in Y \setminus y_1 : \vec{w}^T \Phi(x_1, y_1) \geq \vec{w}^T \Phi(x_1, y) + \Delta(y_1, y) - \xi_1 \\
\vdots \\
\forall y \in Y \setminus y_n : \vec{w}^T \Phi(x_n, y_n) \geq \vec{w}^T \Phi(x_n, y) + \Delta(y_n, y) - \xi_n
\]

Lemma: The training loss is upper bounded by

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

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

- Training:

$$\min_{\bar{w}, \xi \geq 0} \frac{1}{2} \bar{w}^T \bar{w} + C \sum_{i=1}^{n} \xi_i$$

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

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

- Applications: Parsing, Sequence Alignment, Clustering, etc.
Cutting-Plane Algorithm for Structural SVM

- **Input:** \((x_1, y_1), \ldots, (x_n, y_n), C, \epsilon\)
- \(S \leftarrow \emptyset, \vec{w} \leftarrow 0, \xi \leftarrow 0\)
- **REPEAT**
  - **FOR** \(i = 1, \ldots, n\)
    - compute \(\hat{y} = \arg\max_y \{ \Delta(y_i, y) + \vec{w}^T \Phi(x_i, y) \} \)
    - **IF** \((\Delta(y_i, \hat{y}) - \vec{w}^T[\Phi(x_i, y_i) - \Phi(x_i, \hat{y})]) > \xi_i + \epsilon\)
      - \(S \leftarrow S \cup \{ \vec{w}^T[\Phi(x_i, y_i) - \Phi(x_i, \hat{y})] \geq \Delta(y_i, \hat{y}) - \xi_i \}\)
    - \([\vec{w}, \xi] \leftarrow \text{optimize StructSVM over } S\)
  - **ENDIF**
  - **ENDDFOR**
- **UNTIL** \(S\) has not changed during iteration

Find most violated constraint

Violated by more than \(\epsilon\)?

Add constraint to working set
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 \( \epsilon \). The loss has to be bounded \( 0 \leq \Delta(y_i, y) \leq A \), and \( \|\phi(x, y)\| \leq R \).
Experiment: Part-of-Speech Tagging

• Task
  – Given a sequence of words \( x \), predict sequence of tags \( y \).
    \[
    \begin{array}{c}
    \text{The dog chased the cat} \\
    \end{array}
    \rightarrow
    \begin{array}{c}
    \text{Det} \rightarrow \text{N} \rightarrow \text{V} \rightarrow \text{Det} \rightarrow \text{N} \\
    \end{array}
    \]
  – Dependencies from tag-tag transitions in Markov model.

• Model
  – Markov model with one state per tag and words as emissions
  – Each word described by \( \sim 250,000 \) dimensional feature vector (all word suffixes/prefixes, word length, capitalization ...)

• Experiment (by Dan Fleisher)
  – Train/test on 7966/1700 sentences from Penn Treebank

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brill (RBT)</td>
<td>95.78</td>
</tr>
<tr>
<td>HMM (ACOPOST)</td>
<td>95.63</td>
</tr>
<tr>
<td>kNN (MBT)</td>
<td>95.02</td>
</tr>
<tr>
<td>Tree Tagger</td>
<td>94.68</td>
</tr>
<tr>
<td>SVM Multiclass (SVM-light)</td>
<td>95.75</td>
</tr>
<tr>
<td>SVM-HMM (SVM-struct)</td>
<td>96.49</td>
</tr>
</tbody>
</table>
Experiment: Natural Language Parsing

- **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

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<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]

[TsoJoHoAl04]
More Expressive Features

• Linear composition: \( \Phi(x, y) = \sum \phi(x, y_j) \)

\[
\begin{pmatrix}
0 \\
\vdots \\
0 \\
0 \\
\end{pmatrix}
\]

• So far: \( \phi(x, y_i) = \begin{pmatrix} 0 \\
0 \\
1 \\
0 \\
\vdots \\
0 \end{pmatrix} \) if \( y_i = ' S \rightarrow NP \ VP' \)

• General: \( \phi(x, y_i) = \phi_{\text{kernel}}(\phi(x, [\text{rule, start, end]})) \)

• Example:

\[
\begin{pmatrix}
1 \\
(start - end)^2 \\
1 \\
\vdots \\
1 \\
\end{pmatrix}
\]

if \( x_{\text{start}} = " \text{while and x}_{\text{end}} =". " \)

span contains "and"
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
    • $\hat{y} = \arg\max_{y \in Y} [w \cdot \Phi(x, y)]$
    • $\hat{y} = \arg\max_{y \in Y} [\Delta(y_i, y) + w \cdot \Phi(x, y)]$

$\rightarrow$ 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:
  \[ \hat{w} = \text{argmax}_w [-w \cdot w + \lambda \sum_i \log(P(y_i|x_i, w))] \]

- Prediction for zero/one loss:
  \[ \hat{y} = \text{argmax}_y [w \cdot \Phi(x, y)] \]
Structured Prediction

- Discriminative ERM
  - Structural SVMs

- Discriminative MAP
  - Conditional Random Fields

- Generative
  - Hidden Markov Model

- Other Methods
  - Maximum Margin Markov Networks
  - Markov Random Fields
  - Bayesian Networks
  - Statistical Relational Learning