Structured Output Prediction: Discriminative Training

CS4780/5780 – Machine Learning
Fall 2019

Nika Haghtalab & Thorsten Joachims
Cornell University

Reading: Murphy 19.7
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:

- $h_{bayes}(x) = \arg\max_{y \in Y} [P(Y = y|X = x)]$
  
  $= \arg\max_{y \in Y} [P(X = x|Y = y)P(Y = y)]$

- 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:

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

<table>
<thead>
<tr>
<th>( \Phi(x, y) = )</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \overrightarrow{Det \rightarrow N} )</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{Det \rightarrow V} )</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>( \overrightarrow{N \rightarrow V} )</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{V \rightarrow Det} )</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{Det \rightarrow dog} )</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{Det \rightarrow the} )</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{N \rightarrow dog} )</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{V \rightarrow chased} )</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>( \overrightarrow{N \rightarrow cat} )</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

- **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 $h(x) = \text{argmax}_{y \in Y} [\overrightarrow{w} \cdot \phi(x, y)]$

$x$  The dog chased the cat

$y$

```
NP
  /\   /
 Det N V Det N
  \ /  \ /
 The dog chased the the cat
```

$\Phi(x, y) =$

```
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
```
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

Training Set: \( (x_1, y_1), \ldots, (x_n, y_n) \)

Prediction Rule: \( h_{\text{svm}}(x) = \arg \max_{y \in Y} [\vec{w} \cdot \phi(x, y)] \)

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.
Soft-Margin Structural SVM

Soft-margin optimization problem:

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

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

... 

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

Lemma: The training loss is upper bounded by

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

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

- Training:

  $$\min_{w, \xi \geq 0} \frac{1}{2} \hat{\omega}^T \hat{\omega} + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$
  
  s.t. $\forall y \in Y \setminus y_1 : \hat{\omega}^T \Phi(x_1, y_1) \geq \hat{\omega}^T \Phi(x_1, y) + \Delta(y_1, y) - \xi_1$
  
  ... 

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

- 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 \(\hat{y} = \arg\max_{y \in Y} \{ \Delta(y_i, y) + \vec{w}^T \Phi(x_i, y) \}\)
    - IF \((\Delta(y_i, \hat{y}) - \vec{w}^T[\Phi(x_i, y_i) - \Phi(x_i, \hat{y})]) > \xi_i + \epsilon\)
      - \(S \leftarrow S \cup \{ \vec{w}^T[\Phi(x_i, y_i) - \Phi(x_i, \hat{y})] \geq \Delta(y_i, \hat{y}) - \xi_i \}\)
      - \([\vec{w}, \xi] \leftarrow \text{optimize StructSVM over } S\)
  - **ENDIF**
  - ENDFOR
- **UNTIL** \(S\) has not changed during iteration
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 \( \epsilon \). The loss has to be bounded \( 0 \leq \Delta(y_i, y) \leq A \), and \( \|\phi(x, y)\| \leq R \).
Applying StructSVM to New Problem

• Basic algorithm stays the same (e.g. SVM-struct)
• Application specific
  – Loss function $\Delta(y_i, y)$
  – Representation $\Phi(x, y)$
  – Algorithms to compute
    • $\hat{y} = \arg\max_{y \in Y} [w \cdot \Phi(x, y)]$
    • $\hat{y} = \arg\max_{y \in Y} [\Delta(y_i, y) + w \cdot \Phi(x, y)]$

→ 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:
  \[\hat{w} = \arg\min_w [w \cdot w - \lambda \sum_i \log(P(y_i|x_i, w))]\]

• Prediction for zero/one loss:
  \[\hat{y} = \arg\max_y [w \cdot \Phi(x, y)]\]
Encoder/Decoder Networks

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

\[
\begin{align*}
    h_t &= h(W_h h_{t-1} + V_h x_t) \\
    g_t &= g(W_g g_{t-1} + V_g y_{t-1}) \\
    p_y &= f(V_f g_t)
\end{align*}
\]
Structured Prediction

• Discriminative ERM
  – Structural SVMs
  – Encoder/Decoder Nets

• Discriminative MAP
  – Conditional Random Fields

• Generative
  – Hidden Markov Model

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