Linear Models for Structure Prediction

Fernando Pereira

with

Koby Crammer, Ryan McDonald, Fei Sha

Department of Computer and Information Science
University of Pennsylvania

Thanks: Jeff Bilmes, Michael Collins, Andrew McCallum, John Lafferty, Pete White, Scott Winters, NSF EIA0205456, EIA 0205448, IIS 0428193, and DARPA/SRI contract NBCHD030010
Analyzing sequences

- **Text**
  - Information extraction
  - Syntactic structure
- Speech
  - Recognition
  - Language modeling
  - Boundary and segment prediction
- Biological sequences
Challenges

• Interacting decisions

• A wide range of sequence features

• Computing an answer is relatively costly
Analysis as labeling

- Labels give the role of corresponding inputs
- Part-of-speech tagging
- Shallow parsing
- (Some) information extraction
- Other segmentation/labeling tasks
  (speech, genomic sequences,...)

$$x = x_1 \cdots x_n \rightarrow \text{Sequence model} \rightarrow y = y_1 \cdots y_n$$
Previous approaches

- **Generative modeling**: probabilistic generators of sequence-structure pairs
- HMMs, probabilistic CFGs
- Hard to model non-independent features

- **Sequential classification**: decompose structure assignment into a sequence of structural decisions
- Cannot trade-off decisions at different locations: *label-bias* problem
Discriminative Whole Sequence Training

• Discriminative training of HMMs
  • Conditional maximum likelihood, MMI, ...
  • Still thinking generatively about model design

• Step out of the generative mindset
  • Scoring hypotheses instead of generating observations
HMMs in information extraction

[Seymore & McCallum 99, Freitag & McCallum 99]

- **Inputs** $x$: words
  \[ p(x, y) = \prod_i p(y_i|y_{i-1})p(x_i|y_i) \]

- **States** $y$: fields to extract
Problems with HMMs

- Applications need richer input representation

<table>
<thead>
<tr>
<th>Word features</th>
<th>Formatting features</th>
</tr>
</thead>
<tbody>
<tr>
<td>word identity</td>
<td>centered</td>
</tr>
<tr>
<td>capitalization</td>
<td>indentation</td>
</tr>
<tr>
<td>ends in “-tion”</td>
<td>white space ratio</td>
</tr>
<tr>
<td>word in word list</td>
<td>begins with number</td>
</tr>
<tr>
<td>word font</td>
<td>ends with “?”</td>
</tr>
</tbody>
</table>
Generating multiple features

- Relax conditional independence of features on labels \Rightarrow \textit{intractability}
Linear structure models

- Generalize linear classification
  \[ y^* = \arg \max_y \lambda \cdot F(y, x) \]

- Features based on local domains
  \[ F(y, x) = \sum_{C \in \mathcal{C}(x)} f_C(y, x) \]
  \[ f_C(y, x) = f_C(y_C, x) \]

- Efficient Viterbi decoding for tree-structured interactions
Why linear structure models

• Combine the best of generative and classification models:
  • Trade off labeling decisions at different positions
  • Allow overlapping features
• Modular
  • factored scoring
  • loss function
• From features to kernels
How it really started

- **Sequence** conditional random fields

\[
p_\lambda(y|x) = \frac{\exp \lambda \cdot F(y,x)}{Z_\lambda(x)}
\]

\[
Z_\lambda(x) = \sum_y \exp \lambda \cdot F(y,x)
\]

\[
F(y, x) = \sum_i f_i(y, x)
\]

\[
f_i(y, x) = f_i(y_{i-1}, y_i, x)
\]
Training CRFs

• Minimize data log-likelihood

\[ \mathcal{L}_\lambda = \sum_k \log p_\lambda(y_k | x_k) \]
\[ = \sum_k [\lambda \cdot F(y_k, x_k) - \log Z_\lambda(x_k)] \]

• ... by finding unique zero of gradient

\[ \nabla \mathcal{L}_\lambda = \sum_k \left[ F(y_k, x_k) - E_{p_\lambda(Y|x_k)} F(Y, x_k) \right] \]

• Dynamic programming for expectations (forward-backward algorithm)

• Standard convex optimization (L-BFGS)
Rockwell International Corp.'s Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co. to provide structural parts for Boeing's 747 jetliners.
Model details

• Second-order Markov \( y_i = c_{i-1}c_i \)

• Predicates \( p(\mathbf{x}, i) \)

• word unigram, bigram

• POS tag unigram, bigram, trigram

• Features
  \[ y_i = y \]
  \[ y_i = y \land y_{i-1} = y' \]
  \[ c_i = c \]
  \[ y_i = y \land p(\mathbf{x}, i) \]
  \[ c_i = c \land p(\mathbf{x}, i) \]
Evaluation

- **Precision $P$**: what proportion of predicted entities are correct
- **Recall $R$**: what proportion of correct entities are predicted
- **$F_1$ measure**: $\frac{2PR}{P + R}$
- **Alternative methods**:
  - **MEMM**: sequential maximum-entropy classifier, same features as CRF
  - **SVM**: voting among 12 forward and 12 backward SVM sequential classifiers
## Test results

- **F scores**

<table>
<thead>
<tr>
<th>Model</th>
<th>$F_1$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM combination</td>
<td>94.39</td>
</tr>
<tr>
<td>CRF</td>
<td>94.38</td>
</tr>
<tr>
<td>MEMM</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **Significance (McNemar’s)**

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF vs. SVM</td>
<td>0.469</td>
</tr>
<tr>
<td>CRF vs. MEMM</td>
<td>0.001</td>
</tr>
</tbody>
</table>
• Gene/protein mentions:

In the absence of MHC class II, purified soluble D10 TCR bound to Staphylococcus aureus enterotoxin C2 with an association rate of 1.

• Variation events: type, location, and state change

One ER showed a G to T point mutation in the second position of codon 12
## Gene/protein results

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F&lt;sub&gt;1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AbGene</strong></td>
<td>63</td>
<td>65</td>
<td>64</td>
</tr>
<tr>
<td><strong>CRF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>words + spelling</td>
<td>83</td>
<td>77.3</td>
<td>80.1</td>
</tr>
<tr>
<td>(non-)gene tokens + rare trigrams</td>
<td>86.4</td>
<td>78.7</td>
<td>82.4</td>
</tr>
</tbody>
</table>

- **Exact match**

- **AbGene**: Brill-style POS and gene tagger, post-processor
Technical challenges

- Very large number of features:
  - 820,000 at least once on training set
  - 3,800,000 predicates true at least once
- **Overfitting**: For CRF training, Gaussian prior on weights loses effectiveness near convergence

- Proposed solutions
  - Feature induction (McCallum 03)
  - Maximum margin (Taskar et al. 03)
  - Expensive search/optimization
  - Lack of modularity
Alternative: online training

- Process one training instance at a time
- Very simple
- Predictable runtime, small memory
- Adaptable to different loss functions

Basic idea:

\[
\mathbf{w} = 0 \\
\text{for } t = 1, \ldots, T : \\
\quad \text{for } i = 1, \ldots, N : \\
\quad \quad \text{classify } \mathbf{x}_i \text{ incurring loss } l \\
\quad \quad \text{update } \mathbf{w} \text{ to reduce } l
\]
Averaged perceptron
(Collins)

- Average the successive $w$ in generic perceptron algorithm:

\[
\begin{align*}
    w &= 0 \\
    \text{for } t = 1, \ldots, T : \\
    &\quad \text{for } i = 1, \ldots, N : \\
    y^* &= \arg \max_y w \cdot F(y, x_i) \\
    l &= \mathbb{1}[y^* \neq y_i] \\
    w &\leftarrow w + l(F(y_i, x_i) - F(y^*, x_i))
\end{align*}
\]
Discriminative LM
(Collins, Roark et al)

• Features:
  • individual $n$-grams and their backoffs
  • scaled ASR score
• Model form: weighted automaton
• Decoding: weighted intersection of model and lattice
• Training: perceptron, CRF
• Significant accuracy improvements over state-of-the-art methods
Online maximum margin

(MIRA)

• Project onto subspace where the correct structure scores “far enough” above all incorrect ones

\[ w = 0 \]
\[
\text{for } t = 1, \ldots, T : \\
\text{for } i = 1, \ldots, N : \\
\quad w \leftarrow \text{arg min}_{w'} \frac{1}{2} \| w' - w \|^2 \\
\quad \text{s.t. } \forall y : w' \cdot F(y_i, x_i) - w' \cdot F(y, x_i) \geq \text{loss}(y_i, y)
\]

• Exponentially many \( y \)s: select best \( n \) instead

• Use slack variables to handle non-separable situations
Dependency parsing

• Meaningful relationships between words
• Local domains over head-dependent pairs

• Search over all possible structures in cubic time (Eisner, Satta)
Parse features

... part-of-speech word ...

[Diagram showing part-of-speech features and words]

[University of Pennsylvania logo]
# Method comparisons

**NPchunking**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptron</td>
<td>93.5</td>
</tr>
<tr>
<td>Avg. Perceptron</td>
<td>94</td>
</tr>
<tr>
<td>MIRA</td>
<td>94.2</td>
</tr>
<tr>
<td>CRF</td>
<td>94.3</td>
</tr>
</tbody>
</table>

**Named entities**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptron</td>
<td>79.4</td>
</tr>
<tr>
<td>Avg. Perceptron</td>
<td>81.9</td>
</tr>
<tr>
<td>MIRA</td>
<td>82.7</td>
</tr>
<tr>
<td>CRF</td>
<td>83.1</td>
</tr>
</tbody>
</table>

**Dependency parsing**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptron</td>
<td>88.1</td>
</tr>
<tr>
<td>Avg. Perceptron</td>
<td>90.6</td>
</tr>
<tr>
<td>MIRA</td>
<td>90.9</td>
</tr>
<tr>
<td>SVM (seq. class.)</td>
<td>90.3</td>
</tr>
</tbody>
</table>
Conclusions

- We combined the best of generative and discriminative models in a simple but competitive way: *linear structure models*
- What we are doing now:
  - Parsing and its applications
    - non-projective parsing (HLT/EMNLP)
    - relation extraction
  - Discriminative acoustic models
  - Gene finder: great results, another talk