P5: Conditional Random Fields

COS 584

Spring 2021
Shallow Parsing with Conditional Random Fields

Fei Sha and Fernando Pereira
Department of Computer and Information Science
University of Pennsylvania
200 South 33rd Street, Philadelphia, PA 19104
(feisha|pereira)@cis.upenn.edu
Conditional Random Fields

- Generative models (HMMs) great for modeling and predicting entire sequences
  - But require lots of (strong) assumptions
- Discriminative models (MEMMs):
  - Great for adding arbitrary features (both local and global)
  - Cannot trade off decisions at different positions

CRFs provide a middle ground - combine the best of generative and discriminative
History of CRFs

- Lafferty, McCallum, Pereria (2001): introduced CRFs for sequence modeling
- Mitigates the label bias problem (in HMMs/MEMMs)
- Better empirical performance compared to HMMs/MEMMs
- Parameter estimation not straightforward
History of CRFs

• Very popular in the 2000s
• Wide variety of applications:
  • Information extraction
  • Summarization
  • Image labeling/segmentation
History of CRFs

- Very popular in the 2000s
- Wide variety of applications:
  - Information extraction
  - Summarization
  - Image labeling/segmentation

Software [edit]

This is a partial list of software that implement generic CRF tools.

- RNNSharp® CRFs based on recurrent neural networks (C#, .NET)
- CRF-ADF® Linear-chain CRFs with fast online ADF training (C#, .NET)
- CRFSharp® Linear-chain CRFs (C#, .NET)
- GCO® CRFs with submodular energy functions (C++, Matlab)
- DGM® General CRFs (C++)
- GRMM® General CRFs (Java)
- factorie® General CRFs (Scala)
- CRFall® General CRFs (Matlab)
- Sarawagi's CRF® Linear-chain CRFs (Java)
- HCRF library® Hidden-state CRFs (C++, Matlab)
- Accord.NET® Linear-chain CRF, HCRF and HMMs (C#, .NET)
- Wapiti® Fast linear-chain CRFs (C)[15]
- CRFSuite® Fast restricted linear-chain CRFs (C)
- CRF++® Linear-chain CRFs (C++)
- FlexCRFs® First-order and second-order Markov CRFs (C++)
- crf-chain1® First-order, linear-chain CRFs (Haskell)
- imageCRF® CRF for segmenting images and image volumes (C++)
- MALLET® Linear-chain for sequence tagging (Java)
CRFs for shallow parsing (Sha and Pereira)

- Predict non-recursive noun phrases
- Framed as a tagging task in BIO format
- Local features defined on \( X \) (word sequence) and \( Y \) (tag sequence)
- Maximize log likelihood:

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

Figure 1: NP chunks
CRFs for shallow parsing (Sha and Pereira)

- Maximize log likelihood:
  \[ L_\lambda = \sum_k \log p_\lambda(y_k|x_k) \]
  \[ = \sum_k [\lambda \cdot F(y_k, x_k) - \log Z_\lambda(x_k)] \]

- Use forward-backward to compute this efficiently!
Training and features

• Various optimization techniques: conjugate GD, quasi-newton, voted perceptron
• Nowadays - can use SGD with backpropagation
• Second-order markov assumption
• Constraints on certain feature bigrams (e.g. OI) by setting their weights to $-\infty$
## Results

### Table 2: NP chunking F scores

<table>
<thead>
<tr>
<th>Model</th>
<th>F score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM combination (Kudo and Matsumoto, 2001)</td>
<td>94.39%</td>
</tr>
<tr>
<td>CRF</td>
<td>94.38%</td>
</tr>
<tr>
<td>Generalized winnow (Zhang et al., 2002)</td>
<td>93.89%</td>
</tr>
<tr>
<td>Voted perceptron</td>
<td>94.09%</td>
</tr>
<tr>
<td>MEMM</td>
<td>93.70%</td>
</tr>
</tbody>
</table>

### Table 4: McNemar’s tests on labeling disagreements

<table>
<thead>
<tr>
<th>null 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>0.00109</td>
</tr>
<tr>
<td>CRF vs. voted perceptron</td>
<td>0.116</td>
</tr>
<tr>
<td>MEMM vs. voted perceptron</td>
<td>0.0734</td>
</tr>
</tbody>
</table>
CRFs in deep learning era

- Use CRFs on top of neural representations (instead of features and weights)
- Joint sequence prediction without the need for defining features!
- Recent architectures such as seq2seq w/ attention or Transformer may implicitly do the job
Discussion

• Q1: Sha and Pereira (2003) use a BIO labeling scheme where B indicates start of a chunk, I indicates continuation of the chunk and O indicates a word is outside any chunk. Could we add one more tag E for indicating the end of a chunk? What would be some advantages and disadvantages of doing so?

• Q2: The authors make use of words and POS tags to create features for shallow parsing with CRFs. Can you think of other inputs that might result in better features and help do this task better? Think especially about what a noun phrase fundamentally entails (and doesn't) and what information might help identify one.

• Can you think of any applications related to your research/area of study where you can use CRFs?