BMI/CS 776
Lecture 24
Discriminative sequence methods

Colin Dewey
2007.04.19
## Some sequence tasks

<table>
<thead>
<tr>
<th>x (observed)</th>
<th>y (hidden)</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiple sequences</td>
<td>motif locations</td>
</tr>
<tr>
<td>single DNA sequence</td>
<td>gene parse</td>
</tr>
<tr>
<td>single RNA sequence</td>
<td>secondary structure</td>
</tr>
<tr>
<td>multiple sequences</td>
<td>alignment</td>
</tr>
</tbody>
</table>
Generative models

• All models we’ve seen thus far model the joint probability $P(x,y|\theta)$

• Modeled as $P(x,y|\theta) = P(x|y,\theta)P(y|\theta)$

• Given $x$, generative models generally predict $y$ by finding the $\hat{y}$ such that $P(x, \hat{y}|\theta)$ is maximized

• Example (gene finding):
  • DNA sequence ($x$), gene parse ($y$), semi-HMM transition & emission probabilities ($\theta$)
Tradeoffs of generative models

• Advantages:
  • Parameters are often easily interpretable
  • Can be used to simulate (generate) data

• Disadvantages:
  • Models too much: don’t need to model x if all we want to do is predict y
  • Restricted in the features that can be used
Discriminative models

- Only model conditional probability of $y$ given $x$: $P(y|x, \theta)$
- Train to maximize conditional likelihood, not joint likelihood
- Often has better predictive power than generative models
SCFGs for RNA

• For RNA sequence \((x)\), parameters \((\theta)\), and parse \(\sigma\), a SCFG gives \(P(x, \sigma \mid \theta)\)

• To find the probability of a structure \((y)\) given the sequence \((x)\), we must sum over all parses that give that structure

\[
P(y \mid x, \theta) = \sum_{\sigma \in y} P(\sigma \mid x, \theta) = \frac{\sum_{\sigma \in y} P(\sigma, x \mid \theta)}{\sum_{\sigma \in \Omega(x)} P(\sigma, x \mid \theta)}
\]
SCFG Training

• Given training data:

\[ D = \{ (x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)}) \} \]

• Find \( \theta \) to maximize joint likelihood

\[ P(D|\theta) = \prod_{i=1}^{m} P(x^{(i)}, y^{(i)}|\theta) \]

• Easily computed for unambiguous grammars
  (just count transitions and emissions)
SCFG log-linear form

- We can transform the joint likelihood to log-linear form:

\[
P(x, \sigma|\theta) = \prod_{i=1}^{n} p_i F_i(x, \sigma) = \exp \left( \ln \left( \prod_{i=1}^{n} p_i F_i(x, \sigma) \right) \right) = \exp \left( \sum_{i=1}^{n} F_i(x, \sigma) \ln p_i \right) = \exp(w^T F(x, \sigma))
\]

where \( w_i = \ln p_i \)
Using the log linear form of SCFG joint likelihood, we can express the conditional likelihood in the form of a *conditional log-linear model*

\[
P(y|x, \theta) = \frac{\sum_{\sigma \in y} \exp(w^T F(x, \sigma))}{\sum_{\sigma \in \Omega(x)} \exp(w^T F(x, \sigma))} = \frac{1}{Z_w(x)} \sum_{\sigma \in y} \exp(w^T F(x, \sigma))
\]
Conditional log-linear models

\[ P(y|x, \theta) = \frac{1}{Z_w(x)} \sum_{\sigma \in y} \exp(w^T F(x, \sigma)) \]

- For general CLLMs, weights \( w \) and features \( F \) can be arbitrary
- SCFGs are CLLMs where
  - \( w \) is a vector of log probabilities
  - \( F \) features are counts of productions
CLLM parameter estimation

- For generative models, we find the parameters that maximize the joint likelihood:

\[ P(y|x, w) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)}, w) = \prod_{i=1}^{m} \frac{1}{Z_w(x^{(i)})} \exp(w^T F(y^{(i)}, x^{(i)})) \]

- For discriminative models, we can instead find the parameters that maximize the conditional likelihood:

\[ P(y|x, w) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)}, w) = \prod_{i=1}^{m} \frac{1}{Z_w(x^{(i)})} \exp(w^T F(y^{(i)}, x^{(i)})) \]
Improved Iterative Scaling (IIS)

- Gradient of conditional likelihood

\[ \nabla_w P(y|x, w) = \sum_{i=1}^{m} \left( F(x^{(i)}, y^{(i)}) - \mathbb{E}_{y' \sim P(y|x^{(i)}, w)} [F(x^{(i)}, y')] \right) \]

- Maximum reached when expected feature counts equal the observed feature counts!

- One algorithm for climbing to maximum is improved iterative scaling (IIS)

- Similar to the EM algorithm, except that we don’t have missing data
CONTRAfold

- Do et al., 2006
- Reformulate SCFG as CLLM for secondary structure prediction of single RNAs
- Add features not easily captured by SCFG, but that have thermodynamic analogs
- Use maximum expected accuracy parse for secondary structure
- Turns out you can do better than both SCFGs and thermodynamic models!
CONTRAfold features

• Allows for features and weighting not captured by SCFG

• However, features must still be easily calculated through inside-outside form of dynamic programming (e.g. pseudoknots still difficult)

\[
w^T F(x, \sigma) = \begin{bmatrix}
  w_{\text{hairpin length}[0]} \\
  w_{\text{hairpin length}[1]} \\
  w_{\text{hairpin length}[2]} \\
  \vdots
\end{bmatrix}^T \begin{bmatrix}
  \# \text{ of hairpins in } \sigma \text{ of length } 0 \\
  \# \text{ of hairpins in } \sigma \text{ of length } 1 \\
  \# \text{ of hairpins in } \sigma \text{ of length } 2 \\
  \vdots
\end{bmatrix}
\]
MEA parsing

- Define accuracy with respect to true structure $y$
  
  $\text{accuracy}_\gamma(\hat{y}, y) = \# \text{ correctly unpaired positions in } \hat{y} + \gamma \# \text{ correctly paired positions in } \hat{y}$

- Find structure with maximum expected accuracy
  
  $\hat{y}_{\text{mea}} = \arg \max \mathbb{E}_y [\text{accuracy}_\gamma(\hat{y}, y)]$

- Can be determined using Nussinov style algorithm after calculating conditional probabilities of $x_i$ pairing with $x_j$
Generative vs. Discriminative

- Performance of SCFG models converted to CLLM (without adding additional features)

Table 1. Comparison of generative and discriminative model structure prediction accuracy.

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Generative</th>
<th>Discriminative</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>0.0392</td>
<td>0.2713</td>
<td>+0.2321</td>
</tr>
<tr>
<td>G2</td>
<td>0.3640</td>
<td>0.5797</td>
<td>+0.2157</td>
</tr>
<tr>
<td>G3</td>
<td>0.4190</td>
<td>0.4159</td>
<td>−0.0031</td>
</tr>
<tr>
<td>G4</td>
<td>0.1361</td>
<td>0.1350</td>
<td>−0.0011</td>
</tr>
<tr>
<td>G5</td>
<td>0.0026</td>
<td>0.0031</td>
<td>+0.0005</td>
</tr>
<tr>
<td>G6</td>
<td>0.5446</td>
<td>0.5600</td>
<td>+0.0154</td>
</tr>
<tr>
<td>G7</td>
<td>0.5456</td>
<td>0.5582</td>
<td>+0.0126</td>
</tr>
<tr>
<td>G8</td>
<td>0.5464</td>
<td>0.5515</td>
<td>+0.0051</td>
</tr>
<tr>
<td>G6s</td>
<td>0.5501</td>
<td>0.5642</td>
<td>+0.0141</td>
</tr>
</tbody>
</table>

Each number in the table represents the area under the ROC curve of an MEA-based parser using the indicated model. As seen below, the discriminative model consistently outperforms its generative counterpart.

Do et al., 2006
CONTRAfold performance

- Parameter gamma controls sensitivity/specificity tradeoff for CONTRAfold
- CONTRAfold dominates all other single-sequence methods
Feature importance

- Removed features, analyzed decrease in area under ROC curve

**Table 5. Abrasion analysis of CONTRAfold model**

<table>
<thead>
<tr>
<th>Variant</th>
<th>ROC area</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTRAfold</td>
<td>0.6433</td>
<td>n/a</td>
</tr>
<tr>
<td>(without single base stacking)</td>
<td>0.6416</td>
<td>0.0017</td>
</tr>
<tr>
<td>(without helix lengths)</td>
<td>0.6370</td>
<td>0.0063</td>
</tr>
<tr>
<td>(without terminal mismatch penalties)</td>
<td>0.6362</td>
<td>0.0071</td>
</tr>
<tr>
<td>(without full internal loop table)</td>
<td>0.6336</td>
<td>0.0097</td>
</tr>
<tr>
<td>(without helix stacking)</td>
<td>0.6276</td>
<td>0.0157</td>
</tr>
<tr>
<td>(without outer)</td>
<td>0.6271</td>
<td>0.0162</td>
</tr>
<tr>
<td>(without internal loop asymmetry)</td>
<td>0.6134</td>
<td>0.0299</td>
</tr>
<tr>
<td>(without all of the above)</td>
<td>0.6003</td>
<td>0.0430</td>
</tr>
</tbody>
</table>

Do et al., 2006
Discriminative learning for gene prediction

- Hot off the press: Bernal et al., March 2007
- CRAIG: Discriminative gene model with semi-Markov structure
- Flexible features
- Online training algorithm
- Shows improvement over HMM models, particularly in specificity & whole gene accuracy
Score of parse: $S_w(x, s) = \sum_{j=1}^{Q} w \cdot f(s_j, \text{lab}(s_{j-1}), x)$

want to find:

$$\hat{s} = \arg \max_{s \in \text{GEN}(x)} S_w(x, s)$$

Bernal et al., 2006
CRAIG Viterbi

\[
M(i, y) = \max_{s \in \text{GEN}_{i,y}(x)} S_w(x, s)
\]

all segmentations of \(x_1...x_i\) ending in segment with label \(y\)

\[
M(i, y) = \begin{cases} 
\max_{y', 1 \leq l \leq \min \{i, B\}} M(i - l, y') + \\ w \cdot f(\langle i - l, l, y \rangle, y', x) & \text{if } i > 0 \\
0 & \text{if } i = 0 \\
-\infty & \text{otherwise}
\end{cases}
\]

\[
S_w(x, \hat{s}) = M(P + 1, \text{END})
\]

Bernal et al., 2006