Log-Linear Models, Logistic Regression and Conditional Random Fields

February 21, 2013
Generative, Conditional and Discriminative

- Given $\mathcal{D} = (x_t, y_t)_{t=1}^T$ sampled iid from unknown $P(x, y)$
- Generative Learning (maximum likelihood Gaussians)
  - Choose family of functions $p_\theta(x, y)$ parametrized by $\theta$
  - Find $\theta$ by maximizing likelihood: $\prod_{t=1}^T p_\theta(x_i, y_i)$
  - Given $x$, output $\hat{y} = \arg \max_y \frac{p_\theta(x, y)}{\sum_y p_\theta(x, y)}$
- Conditional Learning (logistic regression)
  - Choose family of functions $p_\theta(y|x)$ parametrized by $\theta$
  - Find $\theta$ by maximizing conditional likelihood: $\prod_{t=1}^T p_\theta(y_i|x_i)$
  - Given $x$, output $\hat{y} = \arg \max_y p_\theta(y|x)$
- Discriminative Learning (support vector machines)
  - Choose family of functions $y = f_\theta(x)$ parametrized by $\theta$
  - Find $\theta$ by minimizing classification error $\sum_{t=1}^T \ell(y_i, f_\theta(x_i))$
  - Given $x$, output $\hat{y} = f_\theta(x)$
Generative, Conditional and Discriminative

Generative

Conditional

Discriminative
Generative: Maximum Entropy

Maximum entropy (or generally) minimum relative entropy
\[ \mathcal{RE}(p \| h) = \sum_y p(y) \ln \frac{p(y)}{h(y)} \] subject to linear constraints

\[
\min_p \mathcal{RE}(p \| h) \text{ s.t. } \sum_y p(y)f(y) = 0, \sum_y p(y)g(y) \geq 0
\]

Solution distribution looks like an exponential family model

\[
p(y) = h(y) \exp \left( \theta^\top f(y) + \vartheta^\top g(y) \right) / Z(\theta, \vartheta)
\]

Maximize the dual (the negative log-partition) to get \( \theta, \vartheta \).

\[
\max_{\theta, \vartheta \geq 0} - \ln Z(\theta, \vartheta) = \max_{\theta, \vartheta \geq 0} - \ln \sum_y h(y) \exp \left( \theta^\top f(y) + \vartheta^\top g(y) \right)
\]
All maximum entropy models give an exponential family form:

$$p(y) = h(y) \exp(\theta^\top f(y) - a(\theta))$$

This is also a *log-linear model* over discrete $y \in \Omega$ where $|\Omega| = n$

$$p(y|\theta) = \frac{1}{Z(\theta)} h(y) \exp \left( \theta^\top f(y) \right)$$

- Parameters are vector $\theta \in \mathbb{R}^d$
- Features are $f : \Omega \mapsto \mathbb{R}^d$ mapping each $y$ to some vector
- Prior is $h : \Omega \mapsto \mathbb{R}^+$ a fixed non-negative measure
- Partition function ensures that $p(y|\theta)$ normalizes

$$Z(\theta) = \sum_y h(y) \exp(\theta^\top f(y))$$
Experiments

Generative: Exponential Family and Maximum Likelihood

We are given some iid data $y_1, \ldots, y_T$ where $y \in \{0, 1\}$. If we wanted to find the best parameters of an exponential family distribution known as the Bernoulli distribution:

$$p(y|\theta) = h(y) \exp(\theta^\top f(y) - a(\theta))$$

$$= \theta^y (1 - \theta)^{1-y}$$

This is unsupervised generative learning. We simply find the $\theta$ that maximizes the likelihood

$$L(\theta) = \prod_{t=1}^{T} p(y_t|\theta) = \theta^{\sum_t y_t} (1 - \theta)^{T - \sum_t y_t}$$

Taking log then derivatives and setting to zero gives $\theta = \frac{1}{T} \sum_t y_t$. 
Conditional: Logistic Regression

Given input-output iid data \((x_1, y_1), \ldots, (x_T, y_T)\) where \(y \in \{0, 1\}\). Binary logistic regression computes a probability for \(y = 1\) by

\[
p(y = 1|x, \vartheta) = \frac{1}{1 + \exp(-\vartheta^\top \phi(x))}.
\]

And the probability for \(p(y = 0|x, \theta) = 1 - p(y = 1|x, \theta)\). This is supervised conditional learning.

We find the \(\theta\) that maximizes the conditional likelihood

\[
L(\vartheta) = \prod_{t=1}^{T} p(y_t|x_t, \vartheta)
\]

We can maximize this by doing gradient ascent. Logistic regression is an example of a log-linear model.
Conditional: Log-linear Models

Like an exponential family, but allow $Z$, $h$ and $f$ also depend on $x$

$$p(y|x, \theta) = \frac{1}{Z(x, \theta)} h(x, y) \exp \left( \theta^\top f(x, y) \right)$$

- Parameters are just one long vector $\theta \in \mathbb{R}^d$
- Functions $f : \Omega_x \times \Omega_y \mapsto \mathbb{R}^d$ map $x, y$ to a vector
- Prior is $h : \Omega_x \times \Omega_y \mapsto \mathbb{R}^+$ a fixed non-negative measure
- Partition function ensures that $p(y|x, \theta)$ normalizes

To make a prediction, we simply output

$$\hat{y} = \arg \max_y p(y|x, \theta).$$

Let's mimic (multi-class) logistic regression with this form.
In multi-class logistic regression, we have $y \in \{1, \ldots, n\}$.

$$p(y|x, \theta) = \frac{1}{Z(x, \theta)} h(x, y) \exp \left( \theta^\top f(x, y) \right)$$

If $\phi(x) \in \mathbb{R}^k$, then $f(x, y) \in \mathbb{R}^{kn}$.

Choose the following for the feature function

$$f(x, y) = \left[ \delta[y = 1]\phi(x)^\top \delta[y = 2]\phi(x)^\top \ldots \delta[y = n]\phi(x)^\top \right]^\top.$$  

If $n = 2$ and $h(x, y) = 1$, get traditional binary logistic regression!
Rewrite binary logistic regression $p(y = 1|x, \vartheta) = \frac{1}{1 + \exp(-\vartheta^\top \phi(x))}$ as a log-linear model with $n = 2$, $h(x, y) = 1$ and $f(x, y)$ as before.

$$p(y|x, \theta) = \frac{h(x, y) \exp(\theta^\top f(x, y))}{Z(x, \theta)} = \frac{\exp(f(x, y)^\top \theta)}{\sum_{y=0}^{1} \exp(f(x, y)^\top \theta)}$$

$$p(y = 1|x, \theta) = \frac{\exp([0 \phi(x)^\top] \theta)}{\exp([\phi(x)^\top 0] \theta) + \exp([0 \phi(x)^\top] \theta)} = \frac{1}{1 + \exp([\phi(x)^\top 0] \theta - [0 \phi(x)^\top] \theta)}$$

Can you see how to write $\vartheta$ in terms of $\theta$?
Conditional Random Fields (CRFs)

- Conditional random fields generalize maximum entropy
- Trained on iid data \(\{(x_1, y_1), \ldots, (x_t, y_t)\}\)
- A CRF is just a log-linear model with big \(n\)

\[
p(y|x_j, \theta) = \frac{1}{Z(x_j, \theta)} h(x_j, y) \exp(\theta^\top f(x_j, y))
\]

- Maximum conditional log-likelihood objective function is

\[
J(\theta) = \sum_{j=1}^{t} \ln \frac{h(x_j, y_j)}{Z(x_j, \theta)} + \theta^\top f(x_j, y_j) \tag{1}
\]

- Regularized conditional maximum likelihood is

\[
J(\theta) = \sum_{j=1}^{t} \ln \frac{h(x_j, y_j)}{Z(x_j, \theta)} + \theta^\top f(x_j, y_j) - \frac{t\lambda}{2} \|\theta\|^2 \tag{2}
\]
To train a CRF, we maximize (regularized) conditional likelihood

Traditionally, maximum entropy, log-linear models and CRFs were trained using *majorization* (the EM algorithm is a majorization method)

The algorithms were called *improved iterative scaling (IIS)* or *generalized iterative scaling (GIS)*

- Maximum entropy [Jaynes '57]
- Conditional random fields [Lafferty, et al. '01]
- Log-linear models [Darroch & Ratcliff '72]
Majorization

If cost function $\theta^* = \arg \min_{\theta} C(\theta)$ has no closed form solution, Majorization uses with a surrogate $Q$ with closed form update to monotonically minimize the cost from an initial $\theta_0$

- Find bound $Q(\theta, \theta_i) \geq C(\theta)$ where $Q(\theta_i, \theta_i) = C(\theta_i)$
- Update $\theta_{i+1} = \arg \min_{\theta} Q(\theta, \theta_i)$
- Repeat until converged
Majorization

If cost function $\theta^* = \arg\min_\theta C(\theta)$ has no closed form solution, Majorization uses with a surrogate $Q$ with closed form update to monotonically minimize the cost from an initial $\theta_0$

- Find bound $Q(\theta, \theta_i) \geq C(\theta)$ where $Q(\theta_i, \theta_i) = C(\theta_i)$
- Update $\theta_{i+1} = \arg\min_\theta Q(\theta, \theta_i)$
- Repeat until converged
Majorization

If cost function $\theta^* = \arg \min_\theta C(\theta)$ has no closed form solution, Majorization uses with a surrogate $Q$ with closed form update to monotonically minimize the cost from an initial $\theta_0$

- Find bound $Q(\theta, \theta_i) \geq C(\theta)$ where $Q(\theta_i, \theta_i) = C(\theta_i)$
- Update $\theta_{i+1} = \arg \min_\theta Q(\theta, \theta_i)$
- Repeat until converged
IIS and GIS were preferred until [Wallach ’03, Andrew & Gao ’07]

<table>
<thead>
<tr>
<th>Method</th>
<th>Iterations</th>
<th>LL Evaluations</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIS</td>
<td>$\geq 150$</td>
<td>$\geq 150$</td>
<td>$\geq 188.65$</td>
</tr>
<tr>
<td>Conjugate gradient (FR)</td>
<td>19</td>
<td>99</td>
<td>124.67</td>
</tr>
<tr>
<td>Conjugate gradient (PRP)</td>
<td>27</td>
<td>140</td>
<td>176.55</td>
</tr>
<tr>
<td>L-BFGS</td>
<td>22</td>
<td>22</td>
<td>29.72</td>
</tr>
</tbody>
</table>

Gradient descent appears to be faster
But newer majorization methods are faster still
Gradient Ascent for CRFs

We have the following model

\[ p(y|x, \theta) = \frac{1}{Z(x, \theta)} h(x, y) \exp \left( \theta^\top f(x, y) \right) \]

We want to maximize the conditional (log) likelihood:

\[
\log L(\theta) = \sum_{t=1}^{T} \log p(y_t|x_t, \theta) \\
= \sum_{t=1}^{T} - \log Z(x_t, \theta) + \log(h(x_t, y_t)) + \theta^\top f(x_t, y_t) \\
= const - \sum_{t=1}^{T} \log Z(x_t, \theta) + \theta^\top \sum_{t=1}^{T} f(x_t, y_t)
\]

Same as minimizing the sum of log partition functions plus linear!
Gradient Ascent for CRFs

\[
\frac{\partial \log L}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \theta^\top \sum_{t=1}^{T} \mathbf{f}(x_t, y_t) - \sum_{t=1}^{T} \log Z(x_t, \theta) \right)
\]

\[
= \sum_{t=1}^{T} \mathbf{f}(x_t, y_t) - \sum_{t=1}^{T} \frac{1}{Z(x_t, \theta)} \sum_{y} h(x_t, y) \frac{\partial}{\partial \theta} \exp \left( \theta^\top \mathbf{f}(x_t, y) \right)
\]

\[
= \sum_{t=1}^{T} \mathbf{f}(x_t, y_t) - \sum_{t=1}^{T} \sum_{y} \frac{h(x_t, y)}{Z(x_t, \theta)} \exp \left( \theta^\top \mathbf{f}(x_t, y) \right) \mathbf{f}(x_t, y)
\]

\[
= \sum_{t=1}^{T} \mathbf{f}(x_t, y_t) - \sum_{t=1}^{T} \sum_{y} \mathbf{f}(x_t, y) p(y|x_t, \theta)
\]

The gradient is the difference between the feature vectors at the true labels minus the expected feature vectors under the current distribution. To update, \( \theta \leftarrow \theta + \eta \frac{\partial \log L}{\partial \theta} \).
Stochastic Gradient Ascent for CRFs

Given current $\theta$, update by taking a small step along the gradient

$$\theta \leftarrow \theta + \eta \frac{\partial \log L}{\partial \theta}.$$  

We can use the full derivative:

$$\frac{\partial \log L}{\partial \theta} = \sum_{t=1}^{T} f(x_t, y_t) - \sum_{t=1}^{T} \sum_{y} f(x_t, y)p(y|x_t, \theta)$$

Or do stochastic gradient with only a single random datapoint $t$:

$$\frac{\partial \log L}{\partial \theta} = f(x_t, y_t) - \sum_{y} f(x_t, y)p(y|x_t, \theta)$$
Better Majorization for CRFs

Recall log-linear model over discrete $y \in \Omega$ where $|\Omega| = n$

$$p(y|\theta) = \frac{1}{Z(\theta)} h(y) \exp \left( \theta^\top f(y) \right)$$

- Parameters are vector $\theta \in \mathbb{R}^d$
- Features are $f : \Omega \mapsto \mathbb{R}^d$ mapping each $y$ to some vector
- Prior is $h : \Omega \mapsto \mathbb{R}^+$ a fixed non-negative measure
- Partition function ensures that $p(y|\theta)$ normalizes

$$Z(\theta) = \sum_y h(y) \exp(\theta^\top f(y))$$

Problem: it’s ugly to minimize (unlike a quadratic function)
Better Majorization for CRFs

The bound \( \ln Z(\theta) \leq \ln z + \frac{1}{2}(\theta - \tilde{\theta})^\top \Sigma (\theta - \tilde{\theta}) + (\theta - \tilde{\theta})^\top \mu \)

is tight at \( \tilde{\theta} \) and holds for parameters given by

Input \( \tilde{\theta}, f(y), h(y) \ \forall y \in \Omega \)

Init \( z \to 0^+, \ \mu = 0, \ \Sigma = zI \)

For each \( y \in \Omega \) {
\[
\alpha = h(y) \exp(\tilde{\theta}^\top f(y)) \\
\mathbf{l} = f(y) - \mu \\
\Sigma + = \frac{\tanh(\frac{1}{2} \ln(\alpha/z))}{2 \ln(\alpha/z)} \mathbf{I}^\top \\
\mu + = \frac{\alpha}{z + \alpha} \mathbf{l} \\
z + = \alpha
\]
}

Output \( z, \mu, \Sigma \)
Better Majorization for CRFs

Bound Proof.

1) Start with bound $\log(e^\theta + e^{-\theta}) \leq c\theta^2$ [Jaakkola & Jordan '99]
2) Prove scalar bound via Fenchel dual using $\theta = \sqrt{\vartheta}$
3) Make bound multivariate $\log(e^{\theta^T \mathbf{1}} + e^{-\theta^T \mathbf{1}})$
4) Handle scaling of exponentials $\log(h_1 e^{\theta^T \mathbf{f}_1} + h_2 e^{-\theta^T \mathbf{f}_2})$
5) Add one term $\log(h_1 e^{\theta^T \mathbf{f}_1} + h_2 e^{-\theta^T \mathbf{f}_2} + h_3 e^{-\theta^T \mathbf{f}_3})$
6) Repeat extension for $n$ terms
Better Majorization for CRFs (Bound also Finds Gradient)

Init $z \rightarrow 0^+, \mu = 0, \Sigma = zI$

For each $y \in \Omega$ {
\[ \alpha = h(y) \exp(\tilde{\theta}^T f(y)) \]
\[ l = f(y) - \mu \]
\[ \Sigma + = \frac{\tanh(\frac{1}{2} \ln(\alpha/z))}{2 \ln(\alpha/z)} l^T \]
\[ \mu + = \frac{\alpha}{z+\alpha} l \]
\[ z + = \alpha \}

Output $z, \mu, \Sigma$

Recall gradient $\frac{\partial \log L}{\partial \theta} = \sum_{t=1}^{T} f(x_t, y_t) - \sum_{t=1}^{T} \sum_{y} f(x_t, y)p(y|x_t, \theta)$

The bound’s $\mu$ give part of gradient (can skip $\Sigma$ updates).

$\mu = \sum_{y} f(x_t, y)p(y|x_t, \theta)$
Better Majorization for CRFs

Input \(x_j, y_j\) and functions \(h_{x_j}, f_{x_j}\) for \(j = 1, \ldots, t\)
Input regularizer \(\lambda \in \mathbb{R}^+\)

Initialize \(\theta_0\) anywhere and set \(\tilde{\theta} = \theta_0\)
While not converged
   For \(j = 1\) to \(t\) compute bound for \(\mu_j, \Sigma_j\) from \(h_{x_j}, f_{x_j}, \tilde{\theta}\)
   Set \(\tilde{\theta} = \text{arg min}_\theta \sum_j \frac{1}{2}(\theta - \tilde{\theta})^\top(\Sigma_j + \lambda I)(\theta - \tilde{\theta})\)
   \[+ \sum_j \theta^\top(\mu_j - f_{x_j}(y_j) + \lambda \tilde{\theta})\]
Output \(\hat{\theta} = \tilde{\theta}\)

Theorem

If \(\|f(x_j, y)\| \leq r\) get \(J(\hat{\theta}) - J(\theta_0) \geq (1 - \epsilon) \max_\theta (J(\theta) - J(\theta_0))\)
within \(\left\lceil \ln \left(\frac{1}{\epsilon}\right) / \ln \left(1 + \frac{\lambda \log n}{2r^2 n}\right) \right\rceil\) steps
Convergence Proof

Proof.

Figure: Quadratic bounding sandwich. Compare upper and lower bound curvatures to bound maximum \# of iterations.
**Experiments - Multi-Class Classification & Linear Chains**

<table>
<thead>
<tr>
<th>Data-set</th>
<th>SRBCT</th>
<th>Tumors</th>
<th>Text</th>
<th>SecStr</th>
<th>CoNLL</th>
<th>PennTree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>$n = 4$</td>
<td>$n = 26$</td>
<td>$n = 2$</td>
<td>$n = 2$</td>
<td>$m = 9$</td>
<td>$m = 45$</td>
</tr>
<tr>
<td></td>
<td>$t = 83$</td>
<td>$t = 308$</td>
<td>$t = 1500$</td>
<td>$t = 83679$</td>
<td>$t = 1000$</td>
<td>$t = 1000$</td>
</tr>
<tr>
<td></td>
<td>$d = 9236$</td>
<td>$d = 390260$</td>
<td>$d = 23922$</td>
<td>$d = 632$</td>
<td>$d = 33615$</td>
<td>$d = 14175$</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 10^1$</td>
<td>$\lambda = 10^1$</td>
<td>$\lambda = 10^2$</td>
<td>$\lambda = 10^1$</td>
<td>$\lambda = 10^1$</td>
<td>$\lambda = 10^1$</td>
</tr>
<tr>
<td><strong>Algorithm</strong></td>
<td><strong>time</strong></td>
<td><strong>iter</strong></td>
<td><strong>time</strong></td>
<td><strong>iter</strong></td>
<td><strong>time</strong></td>
<td><strong>iter</strong></td>
</tr>
<tr>
<td>LBFGS</td>
<td>6.10</td>
<td>42</td>
<td>3246.83</td>
<td>8</td>
<td>15.54</td>
<td>7</td>
</tr>
<tr>
<td>Grad</td>
<td>7.27</td>
<td>43</td>
<td>18749.15</td>
<td>53</td>
<td>153.10</td>
<td>69</td>
</tr>
<tr>
<td>Congrad</td>
<td>40.61</td>
<td>100</td>
<td>14840.66</td>
<td>42</td>
<td>57.30</td>
<td>23</td>
</tr>
<tr>
<td>Bound</td>
<td><strong>3.67</strong></td>
<td><strong>8</strong></td>
<td><strong>1639.93</strong></td>
<td><strong>4</strong></td>
<td><strong>6.18</strong></td>
<td><strong>3</strong></td>
</tr>
</tbody>
</table>

**Table:** Time in seconds and iterations to match LBFGS solution for multi-class logistic regression (on SRBCT, Tumors, Text and SecStr data-sets where $n$ is the number of classes) and Markov CRFs (on CoNLL and PennTree data-sets, where $m$ is the number of classes). Here, $t$ is the number of samples, $d$ is the dimensionality of the feature vector and $\lambda$ is the cross-validated regularization setting.
Experiments - Linear Chains

<table>
<thead>
<tr>
<th>Model</th>
<th>Error</th>
<th>oov Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden Markov Model</td>
<td>5.69%</td>
<td>45.59%</td>
</tr>
<tr>
<td>Maximum Entropy Markov Model</td>
<td>6.37%</td>
<td>54.61%</td>
</tr>
<tr>
<td>Conditional Random Field</td>
<td>5.55%</td>
<td>48.05%</td>
</tr>
</tbody>
</table>

**Table:** Accuracy on Penn tree-bank data-set for parts-of-speech tagging with training on half of the 1.1 million word corpus. Note, the oov rate is the error rate on out-of-vocabulary words.

Parts of speech data-set where there are 45 labels per word, e.g.

```
P|R|P| V|B|D| D|T| N|N
| | | | | | | | | |
I saw the man with the telescope
```

\[ p(y|x, \theta) = \frac{1}{Z} \psi(y_1, y_2) \psi(y_2, y_3) \psi(y_3, y_4) \psi(y_4, y_5) \psi(y_5, y_6) \psi(y_6, y_7) \]

How big is \( y \)? Recall graphical models for large spaces...
Bounding Graphical Models with Large $n$

- Each iteration is $O(tn)$, but what if $n$ is large?
- Graphical model: an undirected graph $G$ representing a distribution $p(Y)$ where $Y = \{y_1, \ldots, y_n\}$ and $y_i \in \mathbb{Z}$
- $p(Y)$ factorizes as product of $\{\psi_1, \ldots, \psi_C\}$ functions over $\{Y_1, \ldots, Y_C\}$ subsets of variables over the maximal cliques of $G$

$$p(y_1, \ldots, y_n) = \frac{1}{Z} \prod_{c \in C} \psi_c(Y_c)$$

- E.g. $p(y_1, \ldots, y_6) = \psi(y_1, y_2)\psi(y_2, y_3)\psi(y_3, y_4, y_5)\psi(y_4, y_5, y_6)$
Experiments

Bounding Graphical Models with Large $n$

- Instead of enumerating over all $n$, exploit graphical model
- Build junction tree and run a Collect algorithm
- Useful for computing $Z(\theta)$, $\frac{\partial \log Z(\theta)}{\partial \theta}$ and $\Sigma$ efficiently
- Bound needs $\mathcal{O}(t \sum_c |Y_c|)$ rather than $\mathcal{O}(tn)$
- For an HMM, this is $\mathcal{O}(TM^2)$ instead of $\mathcal{O}(M^T)$
for $c = 1, \ldots, m$ 

$Y_{both} = Y_c \cap Y_{pa(c)}$; $Y_{solo} = Y_c \setminus Y_{pa(c)}$

for each $u \in Y_{both}$ 

initialize $z_{c|x} \leftarrow 0^+$, $\mu_{c|x} = 0$, $\Sigma_{c|x} = z_{c|x} I$

for each $v \in Y_{solo}$

$w = u \otimes v$; $\alpha_w = h_c(w) e^{\tilde{\theta}^T f_c(w)} \prod_{b \in ch(c)} z_{b|w}$

$I_w = f_c(w) - \mu_{c|u} + \sum_{b \in ch(c)} \mu_{b|w}$

$\Sigma_{c|u}^+ = \sum_{b \in ch(c)} \Sigma_{b|w}^+ \frac{\tanh\left(\frac{1}{2} \ln\left(\frac{\alpha_w}{z_{c|u}}\right)\right)}{2 \ln\left(\frac{\alpha_w}{z_{c|u}}\right)} I_w I_w^T$

$\mu_{c|u}^+ = \frac{\alpha_w}{z_{c|u} + \alpha_w} I_w$; $z_{c|u}^+ = \alpha_w$