Bayes Net Learning and Logistic Regression

EECS 349 Spring 2015
Learning in Bayes Nets – the upshot

- Where does the structure come from?
  - Write it down (BNs most useful in this case), or
  - Learn it automatically from data
  - (take 395/495 PGMs course to learn more)
Learning parameters in Bayes Nets

- Just statistical estimation for each CPT

<table>
<thead>
<tr>
<th>Training Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

$P_{ML}(A) = 0.714$

$P_{ML}(B \mid A=1) = 0.6$
Discriminative vs. Generative training

- Say our graph $G$ has variables $X$, $Y$
- Standard BN learning learns $P(X,Y)$
- But often, the only inferences we care about are of form $P(Y|X)$
  - $P(Disease|Symptoms=e)$
  - $P(StockMarketCrash|RecentPriceActivity=e)$
Discriminative vs. Generative training

- Learning $P(X, Y)$: **generative** training
  - Learned model can “generate” the full data $X, Y$
- Learning only $P(Y | X)$: **discriminative** training
  - Model can’t assign probs. to $X$ – only $Y$ given $X$
- Idea: Only model what we care about
  - Don’t “waste data” on params irrelevant to task
  - Side-step false independence assumptions in training (example to follow)
Generative Model Example

- Naïve Bayes model
  - $Y$ binary \{1=spam, 0=not spam\}
  - $X$ an $n$-vector: message has word (1) or not (0)
  - Re-write $P(Y \mid X)$ using Bayes Rule, apply Naïve Bayes assumption
  - $2n + 1$ parameters, for $n$ observed variables
But $P(Y \mid X)$ can be written more compactly

$$P(Y \mid X) = \frac{1}{1 + \exp(w_0 + w_1 x_1 + \ldots + w_n x_n)}$$

Total of $n + 1$ parameters $w_i$
One way to do conversion (vars binary):

\[
\exp(w_0) = \frac{P(Y = 0) \ P(X_1 = 0|Y=0) \ P(X_2 = 0|Y=0)}{P(Y = 1) \ P(X_1 = 0|Y=1) \ P(X_2 = 0|Y=1)} \ldots
\]

for \( i > 0 \):

\[
\exp(w_i) = \frac{P(X_i = 0|Y=1) \ P(X_i = 1|Y=0)}{P(X_i = 0|Y=0) \ P(X_i = 1|Y=1)}
\]
We reduced $2n + 1$ parameters to $n + 1$
- This must be better, right?

Not exactly. If we construct $P(Y \mid X)$ to be equivalent to Naïve Bayes (as before)
- then it’s…equivalent to Naïve Bayes

Idea: optimize the $n + 1$ parameters directly, using training data
Discriminative Training

- In our example:
  \[ P(Y | X) = \frac{1}{1 + \exp(w_0 + w_1 x_1 + \ldots + w_n x_n)} \]

- Goal: find \( w_i \) that maximize likelihood of training data \( Ys \) given training data \( Xs \)
  - Known as “logistic regression”
  - Solved with gradient ascent techniques
  - A convex optimization problem
Naïve Bayes vs. LR

- Naïve Bayes “trusts its assumptions” in training

- Logistic Regression doesn’t – recovers better when assumptions violated
**NB vs. LR: Example**

<table>
<thead>
<tr>
<th>Training Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPAM</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

- Naïve Bayes will classify the last example incorrectly, even after training on it!
- Whereas Logistic Regression is perfect with e.g.,
  \[ w_0 = 0.1 \quad w_{\text{lottery}} = w_{\text{winner}} = w_{\text{lunch}} = -0.2 \quad w_{\text{noon}} = 0.4 \]
Logistic Regression in practice

- Can be employed for any numeric variables $X_i$
  - or for other variable types, by converting to numeric (e.g. indicator) functions

- “Regularization” plays the role of priors in Naïve Bayes

- Optimization tractable, but (way) more expensive than counting (as in Naïve Bayes)
Discriminative Training

- Naïve Bayes vs. Logistic Regression one illustrative case

- Applicable more broadly, whenever queries $P(Y \mid X)$ known \textit{a priori}