Naïve Bayes and Logistic Regression

Naïve Bayes derivation
• Predict $\arg \max_y P(y \mid x) = \arg \max_y P(x \mid y) P(y) / P(x) = \arg \max_y P(x \mid y) P(y)$
• Naïve independence assumption
  \[ P(x \mid y) = \prod_j P(x_j \mid y) \]
• Predict the label $y$ maximizing
  \[ P(y) \prod_j P(x_j \mid y) \]

Naïve Bayes example using max likelihood estimates
• Data: $(x, y)$

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>1</td>
</tr>
</tbody>
</table>

- Predict on $x = (T, F)$ using max likelihood estimates from data
  \[ P(y=+1) = 4/7; \quad P(y=-1) = 3/7 \]
  \[ P(x_1 = T \mid y=+1) = 1/2; \quad P(x_2 = F \mid y=+1) = 1/4 \]
  \[ P(x_1 = T \mid y=-1) = 1/3; \quad P(x_2 = F \mid y=-1) = 2/3 \]
- For $`+1`$: $(4/7)(1/2)(1/4) = 1/14$
- For $`-1`$: $(3/7)(1/3)(2/3) = 2/21$
  Predict $`-1`$

Naïve Bayes example using max likelihood estimates
• Data: $(x, y)$

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>1</td>
</tr>
</tbody>
</table>

- Predict on $x = (T, F)$ using max likelihood estimates from data
  \[ P(y=+1) = 4/7; \quad P(y=-1) = 3/7 \]
  \[ P(x_1 = T \mid y=+1) = 1/2; \quad P(x_2 = F \mid y=+1) = 1/4 \]
  \[ P(x_1 = T \mid y=-1) = 1/3; \quad P(x_2 = F \mid y=-1) = 2/3 \]
- For $`+1`$: $(4/7)(1/2)(1/4) = 1/14$
- For $`-1`$: $(3/7)(1/3)(2/3) = 2/21$
  Predict $`-1`$, even on $+1$ example!
Naïve Bayes properties

• Straight from data, no searching
  – But need to estimate class conditional prob’s
• Successful applications:
  – Diagnosis,
  – Classifying text (Joachims, 1996) 89% accuracy for identifying source from 20 newsgroups (1000 documents each group, 2/3 train 1/3 test)
  – Newsweeder (Lang, 1995) interesting articles up from 16% to 59% after filtering

Naïve Bayes Issues

1. Conditional independence optimistic, but…
   Don’t have to get probabilities right, just the predictions
2. What if an attributeValue-label pair not in training set?
   • Use Laplace estimation.
3. Numeric Features: use Gaussian or other density (Poisson, exponential)
4. Attributes for text classification?
   • Bag of words model

Naïve Bayes for Text

(see Mitchell’s book)

• Let V be the vocabulary (all words/symbols in all training documents)
• For each class y, let Docs_y be the concatenation of all docs labeled y
• For each word w in V, let #w(docs_y) be # of times w occurs in Docs_y
• Set P(w | y) to:
  \( \frac{(\#w(docs_y) + 1)}{|V| + \sum_w \#w(docs_y)} \)
Naïve bayes for text (2)

- Predict on new document \( x \) with class \( y \) maximizing

\[
P(y) \prod_{w \in x} P(w|y)
\]

Note: repeated words multiplied in multiple times

discriminent (boolean case)

- For \( x \), the \( y \) maximizing: \( P(y) \prod_{j} P(x_j | y) \)
  Also maximizes: \( \log(P(y)) + \Sigma_{j} \log(P(x_j | y)) \)
- Let \( a_j = \log(P(x_j = 1 | y); b_j = \log(P(x_j = 0 | y)) \)
  \( \Sigma_{j} \log(P(x_j | y)) = \Sigma_{j} (a_j x_j + b_j (1-x_j)) \)
  \( = \Sigma_{j} x_j (a_j - b_j) + \Sigma_{j} b_j \)
  \( = w \cdot x + c_y \)
- So predict with the class maximizing a set of linear functions - a LTU for two class.

Exercise:

- Repeat slide 3 example using Laplacian probability estimates.
  Calculate the "vote" for each of the two classes for the new instance \( x=(T,F) \).
- Use Naïve Bayes in Weka for your iris2.arff

Logistic Regression

- Assume just two classes, 1 and 0
- Assume \( p(y=1|x) \) is some \( f(w \cdot x) \)
- What should \( f \) be?
  \( f(w \cdot x) = \exp(w \cdot x) / (1 + \exp(w \cdot x)) \)
  \( = 1 / (1 + \exp(-w \cdot x)) \)
Log. Regression (2)

- Logistic regression finds maximum likelihood estimator - find \( w \) such that \( p(w | S) \) maximized for sample \( S \)
- \( p(w | S) = p(S | w) p(w) / p(S) \) (Bayes’ rule)
- Assume uniform prior (\( p(w) \) constant) and \( p(S) \) same for all \( w \), so:
- Find \( w \) maximizing \( p(S | w) \)

Logistic regression 3

- Assume each \((x_i, y_i)\) drawn iid from some (fixed, unknown) distribution
- \( p(S|w)=\prod_i p((x_i, y_i) | w) = \prod_i p(x_i | w) p(y_i | x_i, w) \)
  \( = (\prod_i p(x_i)) \prod_i p(y_i | x_i, w) \)
  \( = \text{const}(x's) \prod_i p(y_i | x_i, w) \)

Logistic Regression 4

- Therefore, find \( w \) maximizing
  \( \prod_i p(y_i | x_i, w) \)
- Which is the \( w \) maximizing
  \[ J(w) = \sum_i \log(p(y_i | x_i, w)) \]
- Take derivatives (some algebra)
  \[ \frac{\partial J(w)}{\partial w_j} = \sum_i (y_i - p(y_i=1 | x_i, w)) x_{ij} \]

Batch Gradient Ascent Alg

1. Initially \( w \) is all 0’s
2. Compute gradient vector \( g \),
   - For each \((x_i, y_i)\) example
     \[ p_i = \frac{1}{1 + \exp(-w \cdot x_i)} \] (predicted prob. of \( y_i = 1 \))
     \[ \text{error}_i = y_i - p_i \]
   - For each feature \( j \)
     \[ g_j = g_j + \text{error}_i \cdot x_{ij} \]
3. Update \( w := w + \eta \cdot g \) \( (\eta \text{ is step size}) \)
4. Go to 2

Second order (Newton-Raphson) methods common
Logistic Regression Summary

• Logistic regression gives distribution on labels: \( p(y=1 \mid x, w) \)
• \( w \cdot x \) is equal to log odds: (exercise?)
  \[
  \log\left( \frac{p(y=1 \mid w,x)}{p(y=0 \mid w,x)} \right)
  \]
• Can threshold at \( w \cdot x = 0 \) to get predictions
• With asymmetric loss can use different thresholds

Questions:

• What are strengths / weaknesses of LDA, Naïve Bayes, logistic regression?
• When might one perform better than another?
• How can you test which learning algorithm is better?

Did Cross-validation here, do earlier?

Exercises

• Implement Perceptron algorithm and test on iris2 data.
• Run logistic regression in Weka on iris2 data
• Compare LDA, perceptron, and logistic regression results

Comparison

<table>
<thead>
<tr>
<th></th>
<th>LDA</th>
<th>Perceptron</th>
<th>Logistic regression</th>
<th>Naïve Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>P(x</td>
<td>y)</td>
<td>P(y</td>
<td>x)</td>
</tr>
<tr>
<td>Data</td>
<td>numeric</td>
<td>Numeric</td>
<td>Numeric</td>
<td>mixed</td>
</tr>
<tr>
<td>interpretable</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Some-what</td>
</tr>
<tr>
<td>Missing values</td>
<td>yes</td>
<td>No</td>
<td>No</td>
<td>yes</td>
</tr>
<tr>
<td>Outliers</td>
<td>Bad</td>
<td>Fatal</td>
<td>Ok</td>
<td>Fair/poor</td>
</tr>
</tbody>
</table>
Robustness

<table>
<thead>
<tr>
<th></th>
<th>LDA</th>
<th>Perceptron</th>
<th>Logistic regression</th>
<th>Naive Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone transform</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>maybe</td>
</tr>
<tr>
<td>irrelevant features</td>
<td>Bad</td>
<td>Bad</td>
<td>Bad</td>
<td>some</td>
</tr>
<tr>
<td>Compute time</td>
<td>good</td>
<td>good</td>
<td>good</td>
<td>good</td>
</tr>
</tbody>
</table>

Exercises (using iris2.arff)

- Duplicate an attribute 10 times, how does it affect algorithms?
- Add 10 random features (say 0,1), how does it affect algorithms?
- Cube an important feature, how does it affect hypothesis?

Bayes Nets Quick Intro

- Topic of much current research
- Models dependence/independence in probability distributions
- Graph based - aka "graphical models"
- Two kinds - directed and undirected

Basic idea:

- Model a joint distribution as a DAG with attributes (features) at the nodes
- Sample distribution from sources

```
  Party  Aptitude  study
     |       |
     v       ^
   Sleep    Pass Exam
```
At each node is a conditional distribution for the attribute depending only on incoming arcs.

In general, working with Bayes nets hard, (easy for trees, also approximate inference - loopy propagation)

Concise representation of some distributions

Very flexible - can fix or learn structure, can have unobserved nodes (latent variables)

Can capture conditional dependencies

Generalizes HMMs and other models

Bayes nets problems:

<table>
<thead>
<tr>
<th>Graph Known</th>
<th>Full Observability</th>
<th>Partial Observability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum likelihood</td>
<td>Expectation-Maximization</td>
<td></td>
</tr>
<tr>
<td>Local search over models</td>
<td>Hard</td>
<td></td>
</tr>
</tbody>
</table>

Bayes Nets Applications

Add probabilities to expert systems

Vista system (Horvitz) for space shuttle problem detection/remedies

Used in Microsoft products: Office ‘95 Answer Wizard, Office ‘97 Office Assistant, etc.

Computational Biology

Special cases applied to speech recognition, turbo coding, and many other places.