Logistic Regression (2 class)  
Alpaydin 10.7 logistic Discrim.

- Hypothesis is a "soft" hyperplane:
  of form $w \cdot x$ (use "add a dimension" trick)
- Assume $p(y=1|x)$ is some known $f(w \cdot x)$
- Experiment:
  - At the start, pick $w$ from some prior
  - For each example:
    - pick $x$'s from some $P(x)$
    - pick $y=1$ with probability $f(w \cdot x)$ ($y=0$ otherwise)
- Learning Goal: learn $w$
  - Model: generate $x$ first then pick $y$

Logistic Regression

- What should $f(w \cdot x)$ be?
- Want confusion at $w \cdot x=0$, more certainty away from boundary
- One $f(w \cdot x) = \exp(w \cdot x) / (1+ \exp(w \cdot x))$
  $= 1 / (1+ \exp(-w \cdot x))$

Logistic regression 3

- Assume each $(x_i,y_i)$ drawn iid from some (fixed, unknown) distribution
- $p(X|w)=\prod_j p((x,y) | x_i, w)$
  $=\prod_j p(x_i| w)p(y_i| x_i, w)$
  $=\text{const}(x_i) \prod_j p(y_i| x_i, w)$

Logistic Regression 4

- Therefore, find $w$ maximizing
  $\prod_j p(y_j| x_j, w)$
  - Which is the the $w$ maximizing $J(w) = \sum_j \log(p(y_j| x_j, w))$
  - Take derivatives (some algebra)
  $\frac{\partial J(w)}{\partial w_j} = \sum_j (y_j-p(y=1|x_j, w)) x_{ij}$
  $\rightarrow$ prediction error
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
   \[ g_j = \text{error}_i \cdot x_{ij} \]
3. Update \( w := w + \eta \cdot g \) \( (\eta \) is step size) \]
4. Go to 2

Second order (Newton-Raphson) methods common

Logistic Regression Summary
- Logistic regression gives distribution on labels: \( p(y=1| x, w) \)
- \( w \cdot x \) is equal to log odds: (exercise)
  \[ \log \left( \frac{p(y=1| w, x)}{p(y=0| w, x)} \right) \]
- 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?

Exercises
- Run logistic regression in Weka on iris 2 data
- Compare Naïve Bayes and logistic regression results

<table>
<thead>
<tr>
<th>Models</th>
<th>LDA</th>
<th>Perceptron</th>
<th>Logistic regression</th>
<th>Naïve Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(x</td>
<td>y)</td>
<td>?</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>

<table>
<thead>
<tr>
<th>Models</th>
<th>LDA</th>
<th>Perceptron</th>
<th>Logistic regression</th>
<th>Naïve 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?