Nearest Neighbor Classification and Regression

Greg Grudic
(Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Notes:

- Downloadable Machine Learning Software
  - Many algorithms studied in this class are implemented in JAVA in the WEKA environment:
    - http://www.cs.waikato.ac.nz/ml/weka/
  - Support Vector Machine code (in C)
    - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- Homework 1: Implement the Nearest Neighbor algorithm in matlab
  - Due Feb 23
  - Details next class

Learning Classification Models

- Collect Training data
- Build Model: happy = f(feature space)
- Make a prediction

<table>
<thead>
<tr>
<th>Dimension 1</th>
<th>Dimension 2</th>
<th></th>
<th></th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>0.95011</td>
<td>0.58279</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Example 2</td>
<td>0.23114</td>
<td>0.4235</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>Example 3</td>
<td>0.8913</td>
<td>0.43291</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Example 4</td>
<td>0.018504</td>
<td>0.76037</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Multi-Class Classification Learning Data...

<table>
<thead>
<tr>
<th>Dimension 1</th>
<th>Dimension 2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>...</td>
<td></td>
<td>$y$</td>
</tr>
<tr>
<td>Example 1</td>
<td>0.85013</td>
<td>0.58279</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Example 2</td>
<td>0.23114</td>
<td>0.4235</td>
<td>...</td>
<td>5</td>
</tr>
<tr>
<td>Example 3</td>
<td>0.8913</td>
<td>0.43291</td>
<td>...</td>
<td>6</td>
</tr>
<tr>
<td>Example 4</td>
<td>0.018504</td>
<td>0.76037</td>
<td>...</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Learning Regression Models

- Collect Training data
- Build Model: stock value = f(feature space)
- Make a prediction

Regression Learning Data...

<table>
<thead>
<tr>
<th>Dimension 1</th>
<th>Dimension 2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>...</td>
<td></td>
<td>$y$</td>
</tr>
<tr>
<td>Example 1</td>
<td>0.85013</td>
<td>0.58279</td>
<td>...</td>
<td>0.22</td>
</tr>
<tr>
<td>Example 2</td>
<td>0.23114</td>
<td>0.4235</td>
<td>...</td>
<td>-17.34</td>
</tr>
<tr>
<td>Example 3</td>
<td>0.8913</td>
<td>0.43291</td>
<td>...</td>
<td>50.1</td>
</tr>
<tr>
<td>Example 4</td>
<td>0.018504</td>
<td>0.76037</td>
<td>...</td>
<td>6.2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The Learning Data

- Symbolic Representation of $N$ learning examples of $d$ dimensional inputs

$$
\begin{pmatrix}
    x_{11} & \cdots & x_{1d} & y_1 \\
    \vdots & \ddots & \vdots & \vdots \\
    x_{N1} & \cdots & x_{Nd} & y_N
\end{pmatrix}
$$
Graphical Representation of 2 Dimensional Classification Training Data

Nearest Neighbor Algorithm

• Given training data \((x_1, y_1), \ldots, (x_N, y_N)\)

• Define a distance metric between points in inputs space. Common measures are:
  - Euclidean (squared) \(D(x, x) = \sum_{j=1}^{d} (x_j - x_{ij})^2\)
  - Weighted Euclidean \(w_j \geq 0\)

\[D(x, x) = \sum_{j=1}^{d} w_j (x_j - x_{ij})^2\]

K-Nearest Neighbor Model

• Given test point \(x\)

• Find the K nearest training inputs \(x_1, \ldots, x_N\) to \(x\) given the distance metric \(D(x, x)\)

• Denote these points as \((x_1, y_1), \ldots, (x_K, y_K)\)

K-Nearest Neighbor Model

• Regression:

\[\hat{y} = \frac{1}{K} \sum_{i=1}^{K} y_i\]

• Classification:

\[\hat{y} = \text{most common class in set } \{y_1, \ldots, y_K\}\]
Picking K

• Goal of Supervised Learning
  – Accurate prediction on future data!!!
• Use N fold cross validation
  – Pick K to minimize the cross validation error
• For each of N training example
  – Find its K nearest neighbors
  – Make a prediction based on these K neighbors (classification and regression)
  – Calculate Error in Prediction (difference between predicted output and actual output)
  – Output average error over all examples
• Use the K that gives lowest average error over the N training examples

Measuring Model Accuracy: Regression

• Assume a set of data \( \{(x_i, y_i)\}_{i=1}^{K} \)
• Regression accuracy of model M
  – Two commonly used metrics
    • Mean Square Error:
      \[ error_{M[i]} = \frac{1}{K} \sum_{i=1}^{K} (y_i - M(x_i))^2 = \frac{1}{K} \sum_{i=1}^{K} (y_i - \hat{y}_i)^2 \]
    • Relative Error:
      \[ error_{M[i]} = \frac{\sum_{i=1}^{K} (y_i - M(x_i))^2}{\sum_{i=1}^{K} (y_i - \bar{y})^2} \]

Measuring Model Accuracy: Classification

• Assume a set of data \( \{(x_i, y_i)\}_{i=1}^{K} \)
• Classification accuracy of model M
  \[ error_{M[i]} = \frac{1}{K} \sum_{i=1}^{K} c(x_i, y_i, M(x_i)) \]
  Where \( c(x_i, y_i, M(x_i)) = \begin{cases} 0 & \text{if } y_i = M(x_i) \\ 1 & \text{otherwise} \end{cases} \)

K-Nearest Neighbor Model: Weighted by Distance

• Regression:
  \[ \hat{y} = \frac{\sum_{k=1}^{K} D(x, x_k) y_k}{\sum_{k=1}^{K} D(x, x_k)} \]
• Classification:
  \[ \hat{y} = \text{most common class in weighted set} \]
  \[ \left\{ \frac{1}{D(x, x_1)} y_1, \ldots, \frac{1}{D(x, x_K)} y_K \right\} \]
Picking $w_1, \ldots, w_d$

- Use N fold cross validation
  - Pick values that minimize the cross validation error
  - This is can be computationally expensive…
- Dimensionality reduction.

Nearest Neighbor Properties – Class Decision Boundaries: The Voronoi Diagram

Each line segment is equidistant between points in opposite classes. The more points, the more complex the boundaries.

K-Nearest Neighbor Algorithm Characteristics

- Universal Approximator
  - Can model any many to one mapping arbitrarily well
- Curse of Dimensionality: Can be easily fooled in high dimensional spaces
  - Dimensionality reduction techniques are often used
- Model can be slow to evaluate for large training sets
  - kd-trees can help
  - Selectively storing data points also helps

kd-trees
More Recent Optimized NN Searches

- Cover Trees
  - http://hunch.net/~jl/projects/cover_tree/cover_tree.html
- Fast for large d…