

Nearest Neighbor Classification and Regression

Greg Grudic

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

1

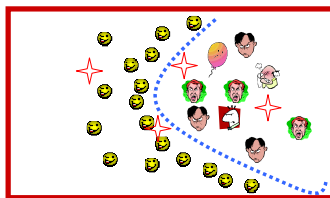
Notes:

- 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

2

Learning Classification Models

- Collect Training data
- Build Model: $\text{happy} = f(\text{feature space})$
- Make a prediction



High Dimensional Feature Space

3

Binary Classification Learning Data...

	Dimension 1 x_1	Dimension 2 x_2	...	y
<i>Example 1</i>	0.95013	0.58279	...	1
<i>Example 2</i>	0.23114	0.4235	...	-1
<i>Example 3</i>	0.8913	0.43291	...	1
<i>Example 4</i>	0.018504	0.76037	...	-1
...

4

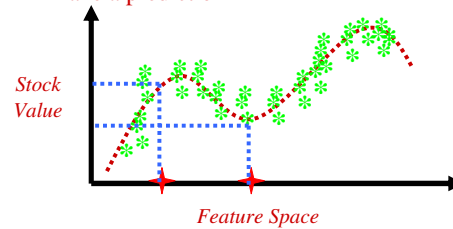
Multi-Class Classification Learning Data...

	Dimension 1 x_1	Dimension 2 x_2	...	y
<i>Example 1</i>	0.95013	0.58279	...	1
<i>Example 2</i>	0.23114	0.4235	...	5
<i>Example 3</i>	0.8913	0.43291	...	6
<i>Example 4</i>	0.018504	0.76037	...	6
...

5

Learning Regression Models

- Collect Training data
- Build Model: stock value = $f(\text{feature space})$
- Make a prediction



6

Regression Learning Data...

	Dimension 1 x_1	Dimension 2 x_2	...	y
<i>Example 1</i>	0.95013	0.58279	...	0.22
<i>Example 2</i>	0.23114	0.4235	...	-17.34
<i>Example 3</i>	0.8913	0.43291	...	50.1
<i>Example 4</i>	0.018504	0.76037	...	6.2
...

7

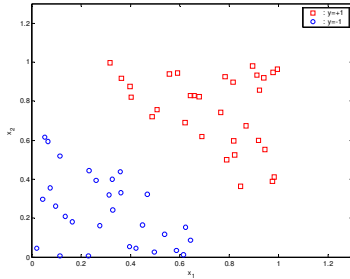
The Learning Data

- Symbolic Representation of N learning examples of d dimensional inputs

$$\begin{pmatrix} x_{11} & \cdots & x_{1d} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{N1} & \cdots & x_{Nd} & y_N \end{pmatrix}$$

8

Graphical Representation of 2 Dimensional Classification Training Data



9

Nearest Neighbor Algorithm

- Given training data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$
- Define a distance metric between points in inputs space. Common measures are:

- Euclidean (squared) $D(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^d (x_j - x_{i,j})^2$

- Weighted Euclidean $w_j \geq 0$

$$D(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^d w_j (x_j - x_{i,j})^2$$

10

K-Nearest Neighbor Model

- Given test point \mathbf{x}
- Find the K nearest training inputs $\mathbf{x}_1, \dots, \mathbf{x}_N$ to \mathbf{x} given the distance metric $D(\mathbf{x}, \mathbf{x}_i)$

- Denote these points as $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_K, y_K)$

11

K-Nearest Neighbor Model

- Regression:

$$\hat{y} = \frac{1}{K} \sum_{k=1}^K y_k$$

- Classification:

$$\hat{y} = \text{most common class in set } \{y_1, \dots, y_K\}$$

12

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 out and actual out)
 - Output average error over all examples
- Use the K that gives lowest average error over the N training examples

13

Measuring Model Accuracy: Regression

- Assume a set of data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_K, y_K)$
- Regression accuracy of model M
 - Two commonly used metrics

- Mean Square Error

$$error_{M(\mathbf{x})} = \frac{1}{K} \sum_{i=1}^K (y_i - M(\mathbf{x}_i))^2 = \frac{1}{K} \sum_{i=1}^K (y_i - \hat{y}_i)^2$$

- Relative Error

$$error_{M(\mathbf{x})} = \frac{\sum_{i=1}^K (y_i - M(\mathbf{x}_i))^2}{\sum_{i=1}^K (y_i - \bar{y})^2}$$

14

Measuring Model Accuracy: Classification

- Assume a set of data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_K, y_K)$
- Classification accuracy of model M

$$error_{M(\mathbf{x})} = \frac{1}{K} \sum_{i=1}^K c(\mathbf{x}_i, y_i, M(\mathbf{x}_i))$$

$$\text{Where } c(\mathbf{x}_i, y_i, M(\mathbf{x}_i)) = \begin{cases} 0 & \text{if } y_i = M(\mathbf{x}_i) \\ 1 & \text{otherwise} \end{cases}$$

15

K-Nearest Neighbor Model: Weighted by Distance

- Regression:

$$\hat{y} = \frac{\sum_{k=1}^K D(\mathbf{x}, \mathbf{x}_k) y_k}{\sum_{k=1}^K D(\mathbf{x}, \mathbf{x}_k)}$$

- Classification:

$\hat{y} =$ most common class in weighted set

$$\left\{ \frac{1}{D(\mathbf{x}, \mathbf{x}_1)} y_1, \dots, \frac{1}{D(\mathbf{x}, \mathbf{x}_K)} y_K \right\}$$

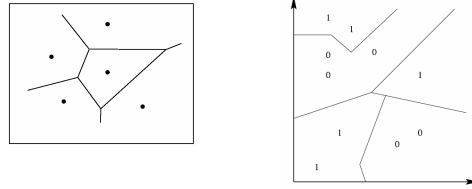
16

Picking w_1, \dots, w_d

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

17

Nearest Neighbor Properties – Class Decision Boundaries: The Voronoi Diagram



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

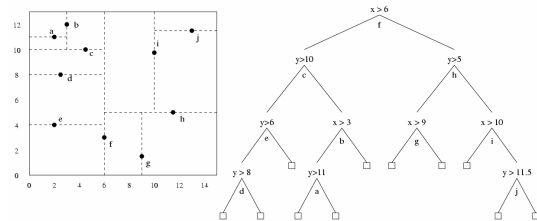
18

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

19

kd-trees



20

More Recent Optimized NN Searches

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