Learning from Data: Nearest Neighbour Methods

Amos Storkey, School of Informatics

October 10, 2005

http://www.anc.ed.ac.uk/~amos/lfd/

Classification

- Training data with attributes $\mathbf{x}$ and class label t.
- $\mathbf{x}$ could represent the presence or absence of a set of words in a web page, and t could be whether Tim Jericho is interested in that particular web page.
- Nearest neighbour classification: Things which are similar in $\mathbf{x}$-space should have the same class label with a high probability.
- This is a smoothness assumption.
- Not going to build an explicit model of the data in this case.
- Discriminative approach.

Similarity

- How are two data points similar?
- Define a dissimilarity function between data points. Usually this involves defining some metric or distance measure such as the Euclidean distance:

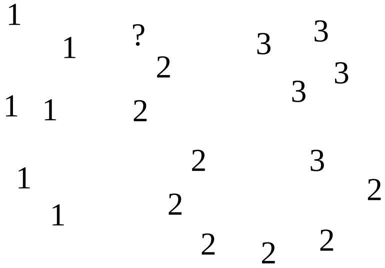
$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})$$

- Possible to be more general. For example one attribute may be more important than another attribute, and should be weighted differently in the distance calculation.

Nearest Neighbour

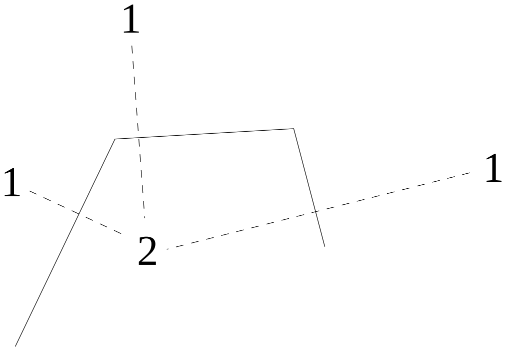
- Have training data $(\mathbf{x}_i, \mathbf{t}_i)$, $i = 1, 2, \ldots, n$.
- Have some test point $\mathbf{x}$ we wish to classify.
- Calculate the dissimilarity between the test point $\mathbf{x}$ and the training points.
- Find which training point k which is 'closest' to the test point. In other words find the minimum dissimilarity of those you calculated.
- Set the classification t for the test point to be identical to that of the nearest training point k.
- In the case of dissimilarity ties, pick the classification which is most common amongst those nearest neighbours.

Example



- Three classes. Training set. Test point '?'.
- Nearest training point is classified as '2'.

Decision boundary



- Where the classification label given by the algorithm flips from one class to another
- Figure: the decision boundary for the nearest neighbour method is piecewise linear.

Problems

- Sensitive to outliers.
- Store all the data.
- Cost of calculating distances.
- Invariance to linear transformation.
- No measure of certainty.

K Nearest Neighbours (KNN)

- Have training data $(\mathbf{x}_i, \mathbf{t}_i)$, $i = 1, 2, \ldots, n$.
- Have some test point $\mathbf{x}$ we wish to classify.
- Calculate the dissimilarity between the test point $\mathbf{x}$ and the training points.
- Find the K training points $k_1, k_2, \ldots k_K$ which is 'closest' to the test point.
- Set the classification t for the test point to be the most common of the K nearest neighbours.
- Solves the problem of outliers

Choosing K

- K is dependent on the 'smoothness' of the classification model we have in mind.
- Large K - everything classified the same.
- Small K - individual points (including outliers) can have significant effects.
- Varying K - varying smoothness of classification.
- Set using generalisation performance. Set aside a validation data set, and test performance on that dataset for different values of K.

Examples

- Comparison with class-conditional models.
- Handwritten character example - see notes.

Summary

- Distance between data points.
- Nearest neighbour calculation.
- Nearest neighbour classification.
- Decision boundaries.
- Outliers.
- K Nearest Neighbours.
- Setting K using generalisation performance.