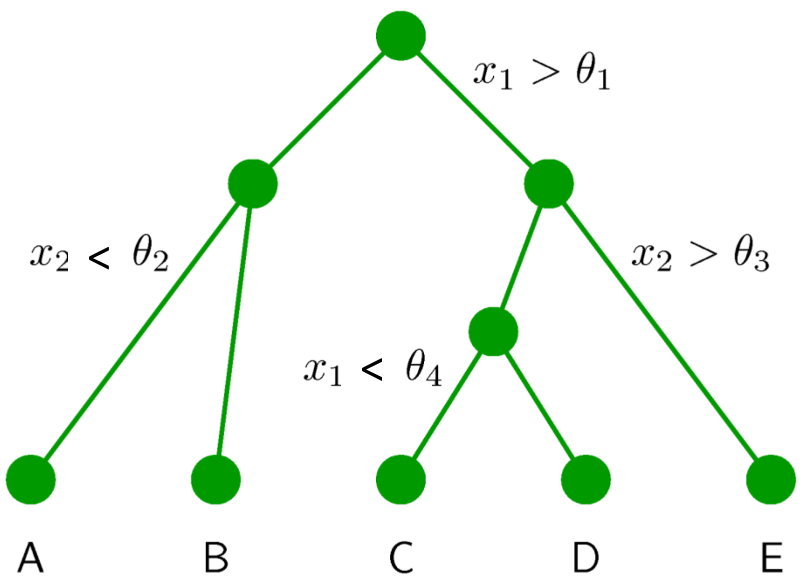
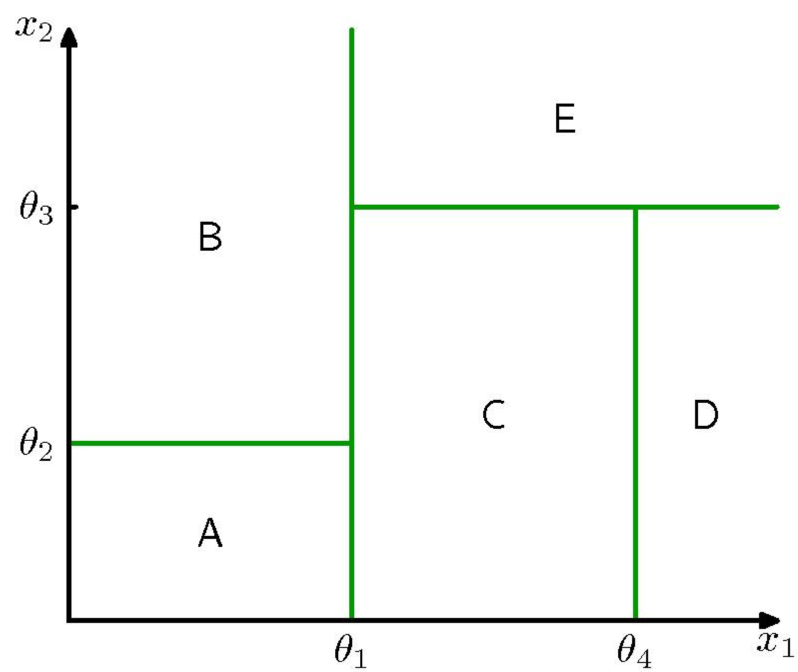
CS485/685 Machine Learning
Lecture 3: Jan 12, 2016

Nearest Neighbour
[RN] Sec. 18.8.1, [HTF] Sec. 2.3.2,
[D] Chapt. 2, [B] Sec. 2.5.2,
[M] Sec. 1.4.2

Decision tree with continuous attributes

- Tree partitions the input space





Decision tree with continuous attributes

- How do we come up with good partitions?

- Common approach: thresholding
  - Single attribute: $x_j > \theta_j$
  - Multi-attribute: $f(x_1, \ldots, x_M) > \theta_j$
    - Where f can be linear or non-linear

Single Attribute Thresholding

- Idea:
  - Discretize continuous attribute into finite set of intervals.
  - Pick thresholds midway between pairs of consecutive values
- Example:

Full Tree

- In the limit, single attribute thresholding leads to a full tree with one example per leaf
  - Partition input space into bins or hypercubes
  - Future examples classified according to bins' labels
    - Close to "nearest neighbour" classification
- Picture:

Nearest Neighbour Classification

- Instead of building tree, find nearest neighbour

$$x^* = argmin_{x'} \, d(x, x')$$

Label: $y_x \leftarrow y_{x^*}$

- Distance measures: $d(x, x')$

L_1: $d(x, x') = \sum_j^M |x_j - x_j'|$

L_2: $d(x, x') = \sum_j^M |x_j - x_j'|^2$

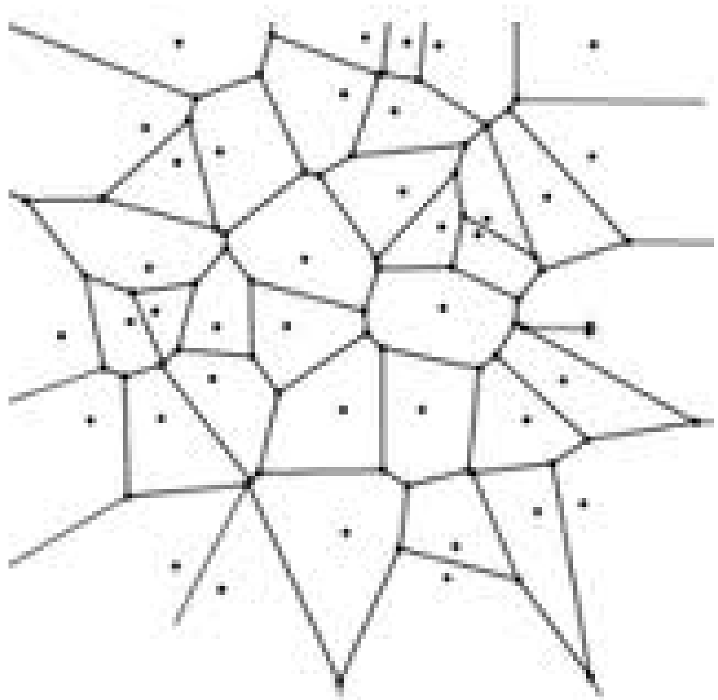
…

L_p: $d(x, x') = \sum_j^M |x_j - x_j'|^p$

Weighted dimensions: $d(x, x') = \sum_j^M c_j |x_j - x_j'|^p$

Voronoi diagram

- Partition implied by nearest neighbour
  - Assuming Euclidean distance

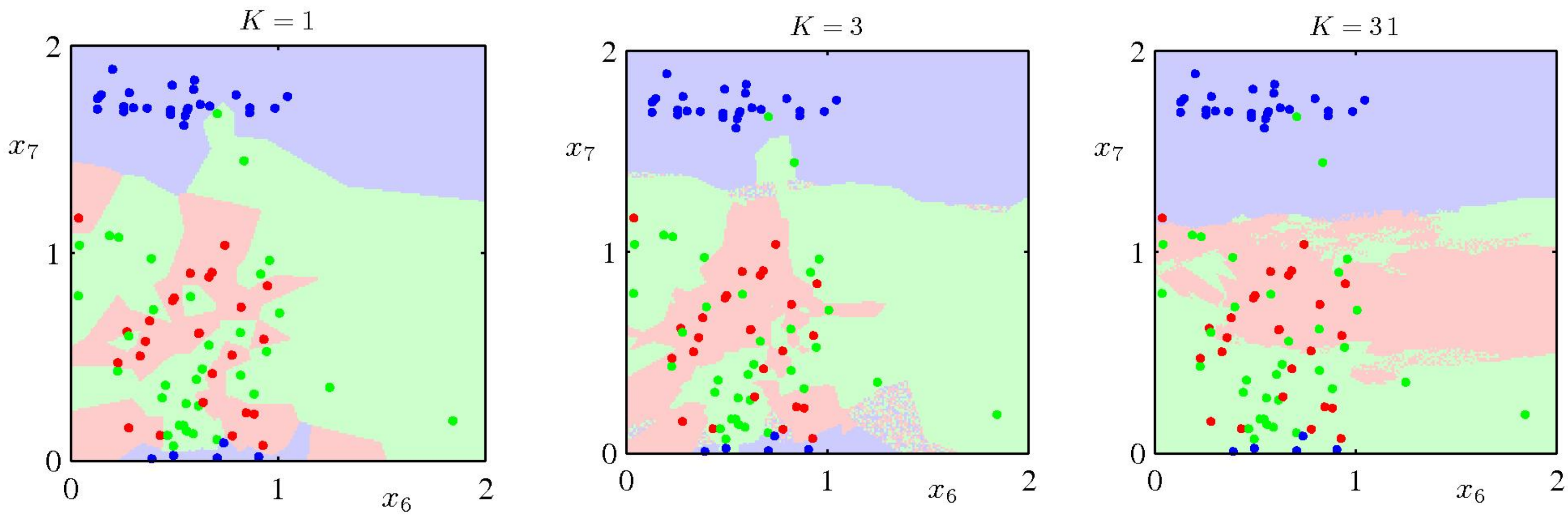


K-nearest neighbour

- Nearest neighbour often instable (overfitting)
- Idea: assign most frequent label among k-nearest neighbours
  - Let $knn(x)$ be the k-nearest neighbours of x according to distance d
  - Label: $y_x \leftarrow mode(\{y_{x'} | x' \in knn(x)\})$

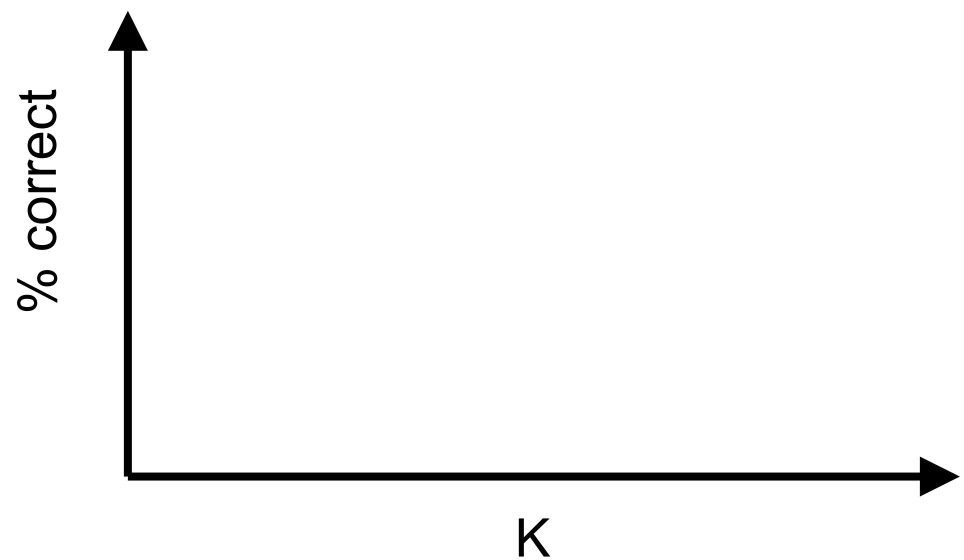
Effect of K

- K controls the degree of smoothing.
- Which partition do you prefer? Why?



Choosing K

- Best K depends on
  - Problem
  - Amount of training data
- Choose K by k-fold cross validation



Complexity

- Nearest neighbour computation:
  - Training: no computation (simply store examples)
  - Testing: return label of nearest example
- Complexity with respect to
  - N: size of training set
  - M: number of attributes

	Training	**Testing**
Decision tree		
Nearest neighbour		