Classification

Nearest Neighbor

Instance based classifiers

Set of Stored Cases

Atr1	………	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- **Store the training samples**
- **Use training samples to predict the class label of test samples**

Unseen Case

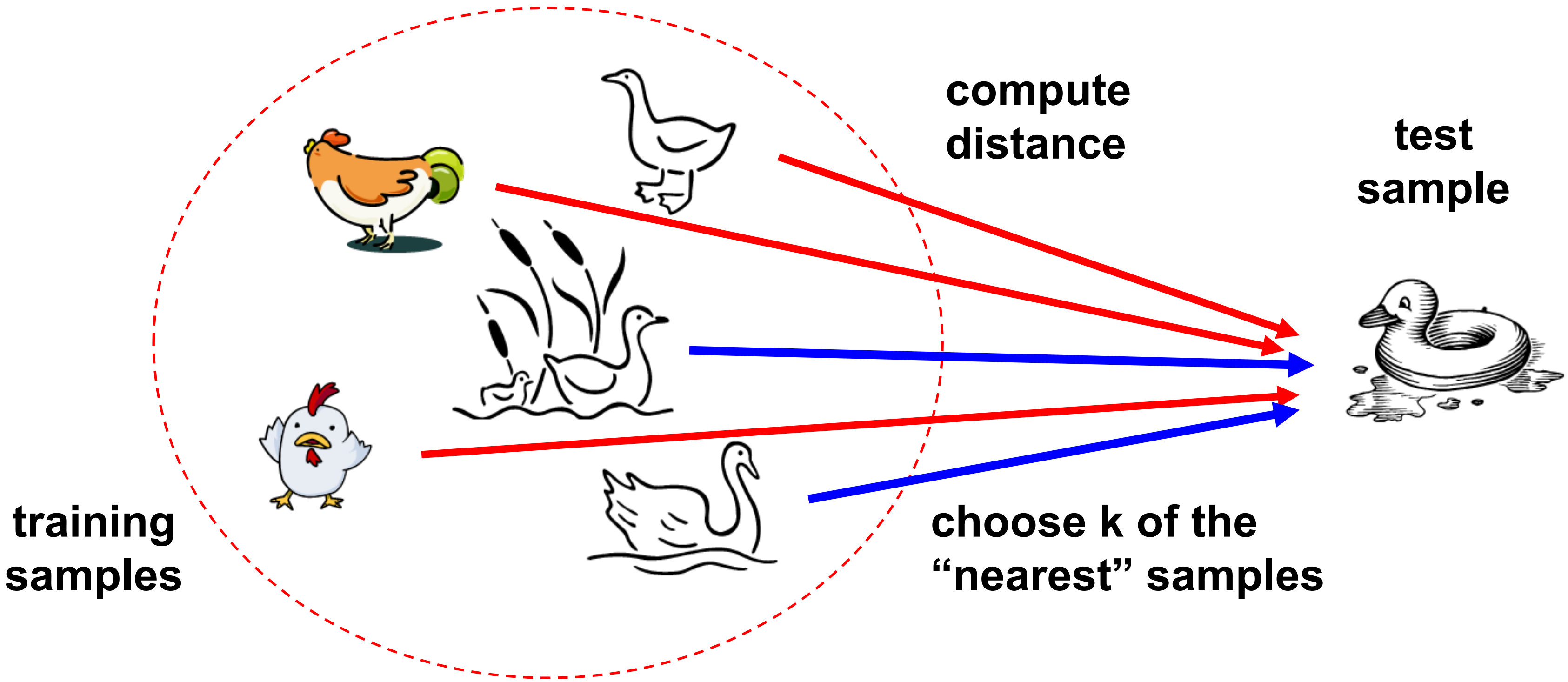
Atr1	………	AtrN

Instance based classifiers

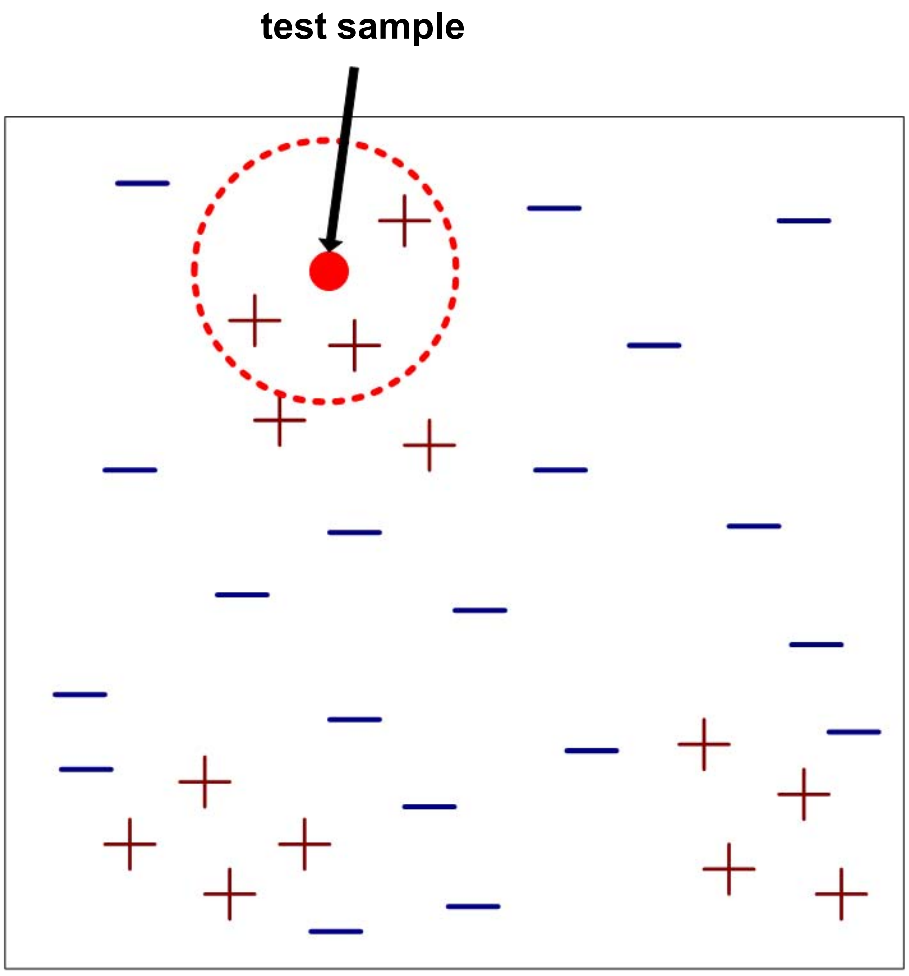
- Examples:
  - Rote learner
    - memorize entire training data
    - perform classification only if attributes of test sample match one of the training samples exactly
  - Nearest neighbor
    - use *k* “closest” samples (nearest neighbors) to perform classification

Nearest neighbor classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



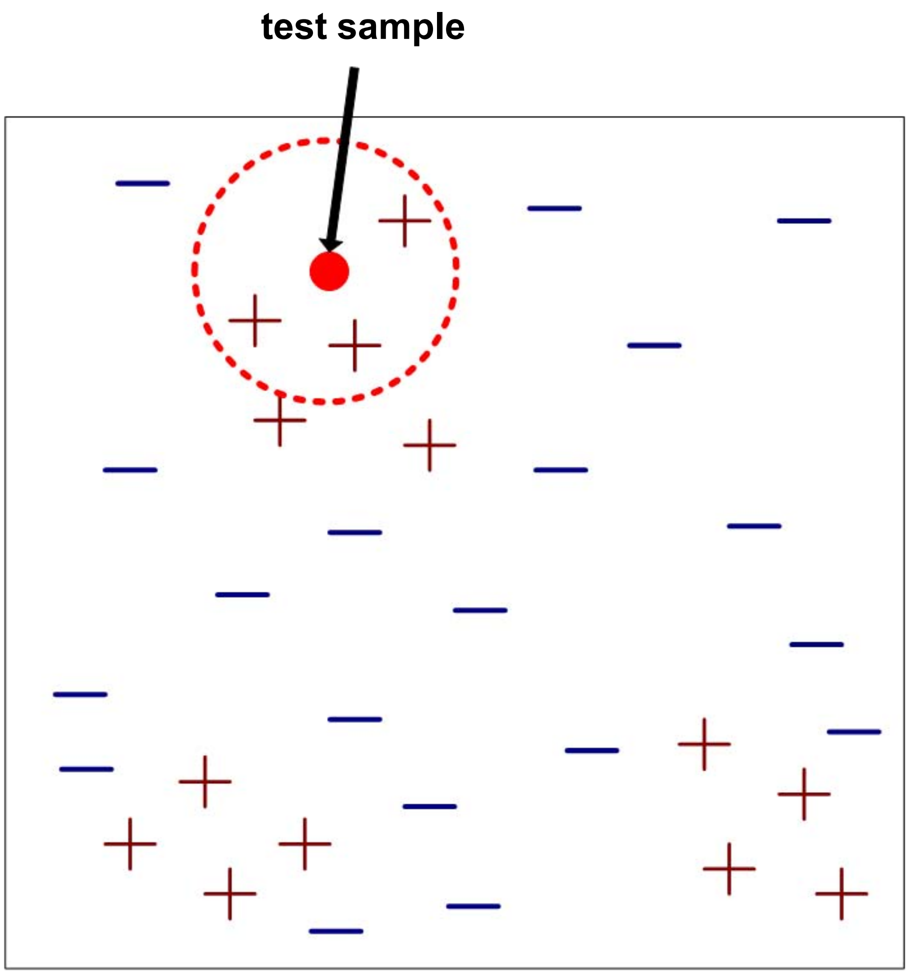
Nearest neighbor classifiers



Requires three inputs:

1. The set of stored samples
2. Distance metric to compute distance between samples
3. The value of *k*, the number of nearest neighbors to retrieve

Nearest neighbor classifiers

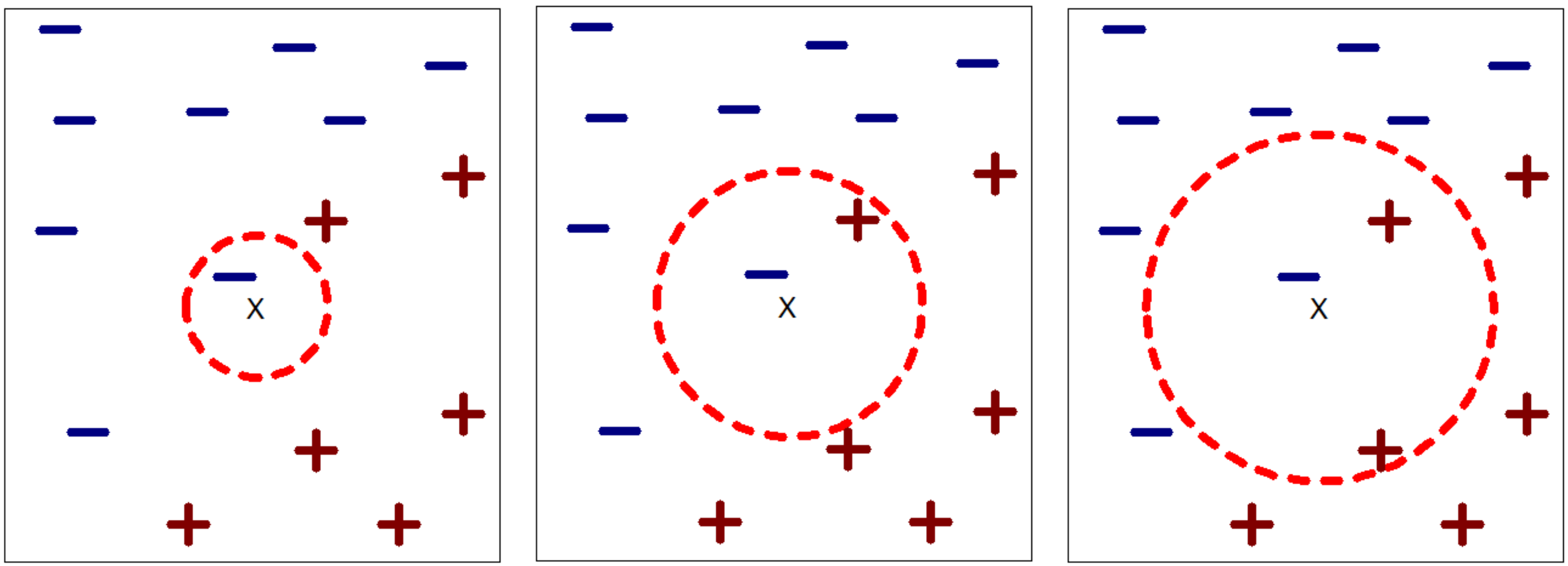


To classify test sample:

1. Compute distances to samples in training set
2. Identify *k* nearest neighbors
3. Use class labels of nearest neighbors to determine class label of test sample (e.g. by taking majority vote)

Definition of nearest neighbors

k-nearest neighbors of test sample x are training samples that have the *k* smallest distances to x



1-nearest neighbor **2-nearest neighbor** **3-nearest neighbor**

Distances for nearest neighbors

- Options for computing distance between two samples:
  - Euclidean distance

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

  - Cosine similarity

$$d(\mathbf{x},\mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$$

  - Hamming distance
  - String edit distance
  - Kernel distance
  - Many others

Distances for nearest neighbors

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5 m to 1.8 m
    - weight of a person may vary from 90 lb to 300 lb
    - income of a person may vary from $10K to $1M

Distances for nearest neighors

- Euclidean measure: high dimensional data subject to curse of dimensionality
  - range of distances compressed

1 0 1 0 1 0 1 0 1 0 1 0		vs.	**0 0 0 0 0 0 0 0 0 0 0 0**
0 1 0 1 0 1 0 1 0 1 0 1			**0 0 0 0 0 0 0 0 0 0 0 1**
d = 3.46			**d = 1.00**

  - effects of noise more pronounced
  - one solution: normalize the vectors to unit length

Distances for nearest neighbors

- Cosine similarity measure: high dimensional data subject often very sparse
  - example: word vectors for documents

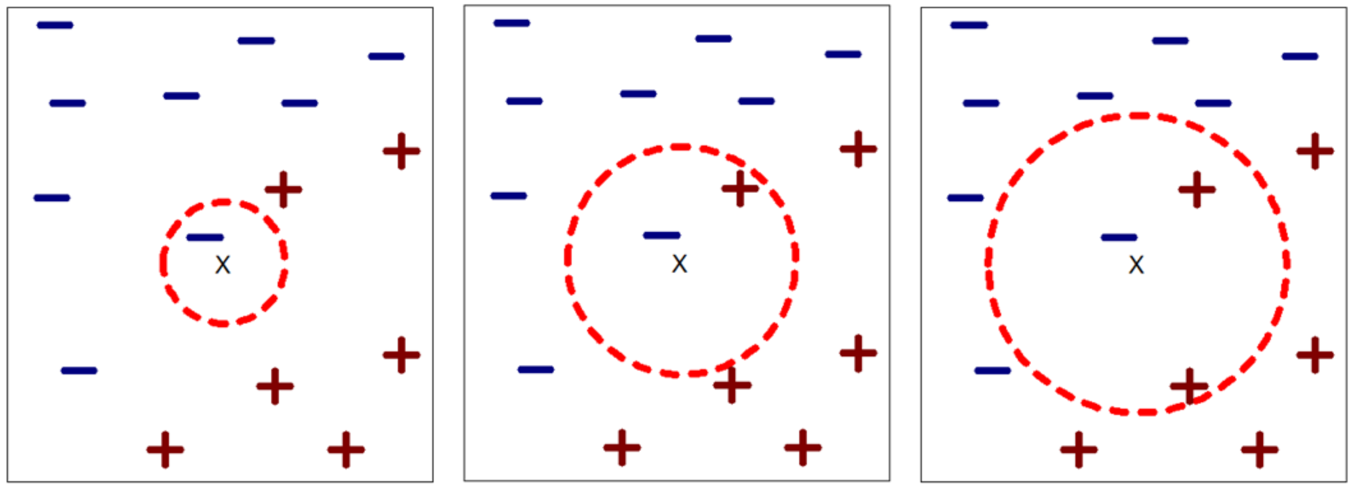
LA Times section	Average cosine similarity within section
Entertainment	0.032
Financial	0.030
Foreign	0.030
Metro	0.021
National	0.027
Sports	0.036
Average across all sections	0.014

  - nearest neighbor rarely of same class
  - one solution: use larger values for *k*

Predicting class from nearest neighbors

- Options for predicting test class from nearest neighbor list
  - Take majority vote of class labels among the *k*-nearest neighbors
  - Weight the votes according to distance
    - example: weight factor $w = 1 / d^2$

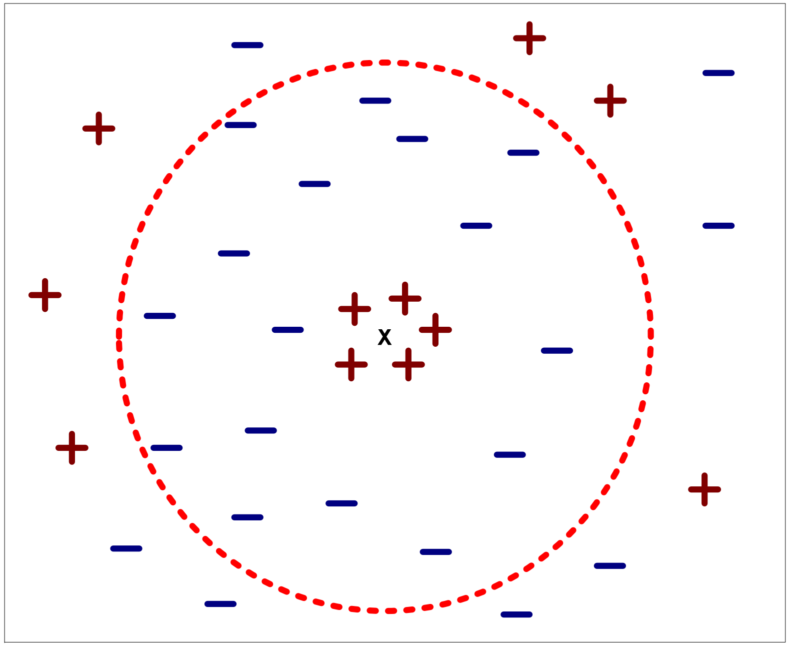
Predicting class from nearest neighbors



nearest neighbors	1	2	3
majority vote	–	?	+
distance-weighted vote	–	–	– or +

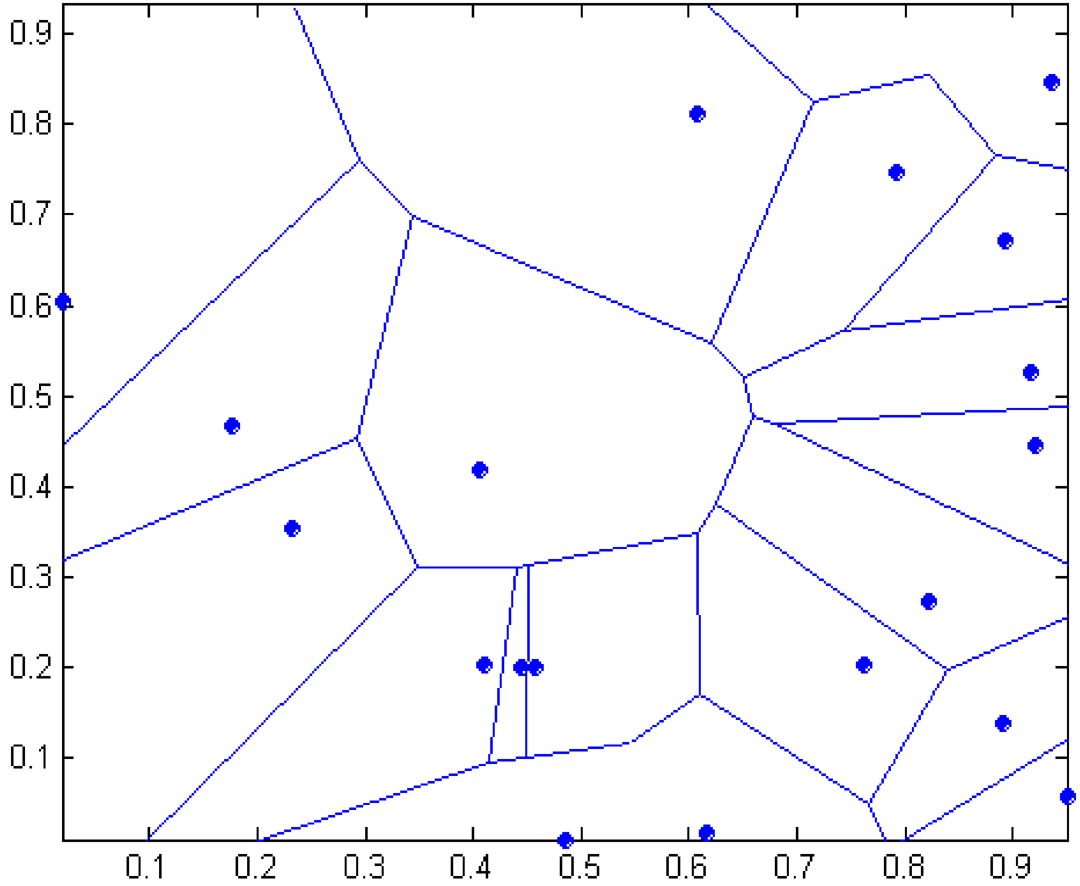
Predicting class from nearest neighbors

- Choosing the value of *k*:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes



1-nearest neighbor

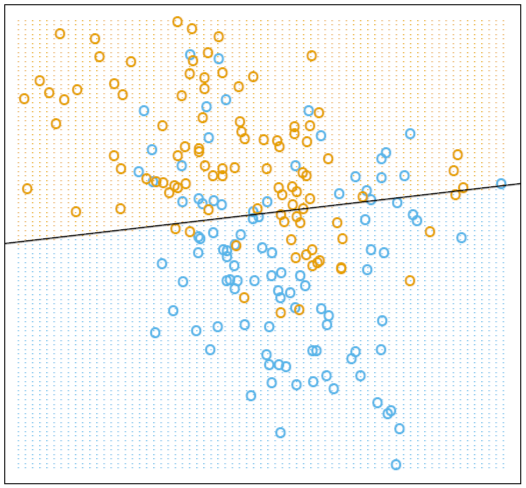
Voronoi diagram



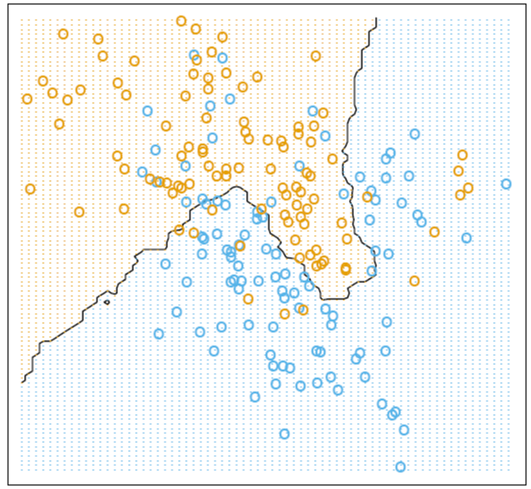
Nearest neighbor classification

- *k*-Nearest neighbor classifier is a lazy learner.
  - Does not build model explicitly.
  - Unlike eager learners such as decision tree induction and rule-based systems.
  - Classifying unknown samples is relatively expensive.
- *k*-Nearest neighbor classifier is a local model, vs. global models of linear classifiers.
- *k*-Nearest neighbor classifier is a non-parametric model, vs. parametric models of linear classifiers.

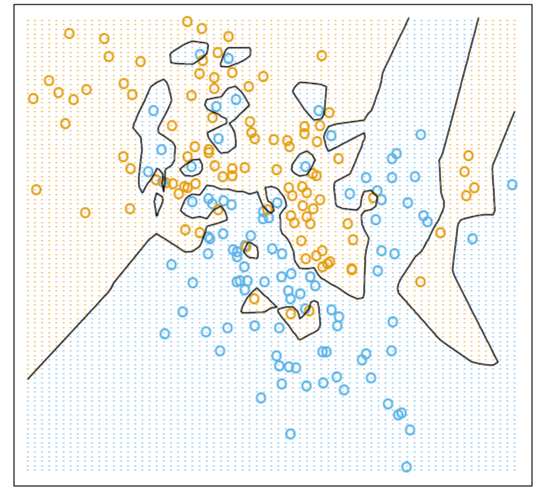
Decision boundaries in global vs. local models



logistic regression



15-nearest neighbor



1-nearest neighbor

- global
- stable
- can be inaccurate

- local
- unstable
- accurate

stable: model decision boundary not sensitive to addition or removal of samples from training set

What ultimately matters: ***GENERALIZATION***

Example: PEBLS

- PEBLS: Parallel Examplar-Based Learning System (Cost & Salzberg)
  - Works with both continuous and nominal features
    - For nominal features, distance between two nominal values is computed using modified value difference metric (MVDM)
  - Each sample is assigned a weight factor
  - Number of nearest neighbor, $k = 1$

Example: PEBLS

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Distance between nominal attribute values:

d(Single,Married)

= | 2/4 – 0/4 | + | 2/4 – 4/4 | = 1

d(Single,Divorced)

= | 2/4 – 1/2 | + | 2/4 – 1/2 | = 0

d(Married,Divorced)

= | 0/4 – 1/2 | + | 4/4 – 1/2 | = 1

d(Refund=Yes,Refund=No)

= | 0/3 – 3/7 | + | 3/3 – 4/7 | = 6/7

Class	Marital Status		
	Single	Married	Divorced
Yes	2	0	1
No	2	4	1

Class	Refund	
	Yes	No
Yes	0	3
No	3	4

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|$$

Example: PEBLS

Tid	Refund	Marital Status	Taxable Income	Cheat
X	Yes	Single	125K	**No**
Y	No	Married	100K	**No**

Distance between record X and record Y:

$$\Delta(X,Y) = w_X w_Y \sum_{i=1}^{d} d(X_i, Y_i)^2$$

where:

$$w_X = \frac{\text{Number of times X is used for prediction}}{\text{Number of times X predicts correctly}}$$

$w_X \cong 1$ if X makes accurate prediction most of the time

$w_X > 1$ if X is not reliable for making predictions

Nearest neighbor regression

- Steps used for nearest neighbor classification are easily adapted to make predictions on continuous outcomes.