Machine Learning

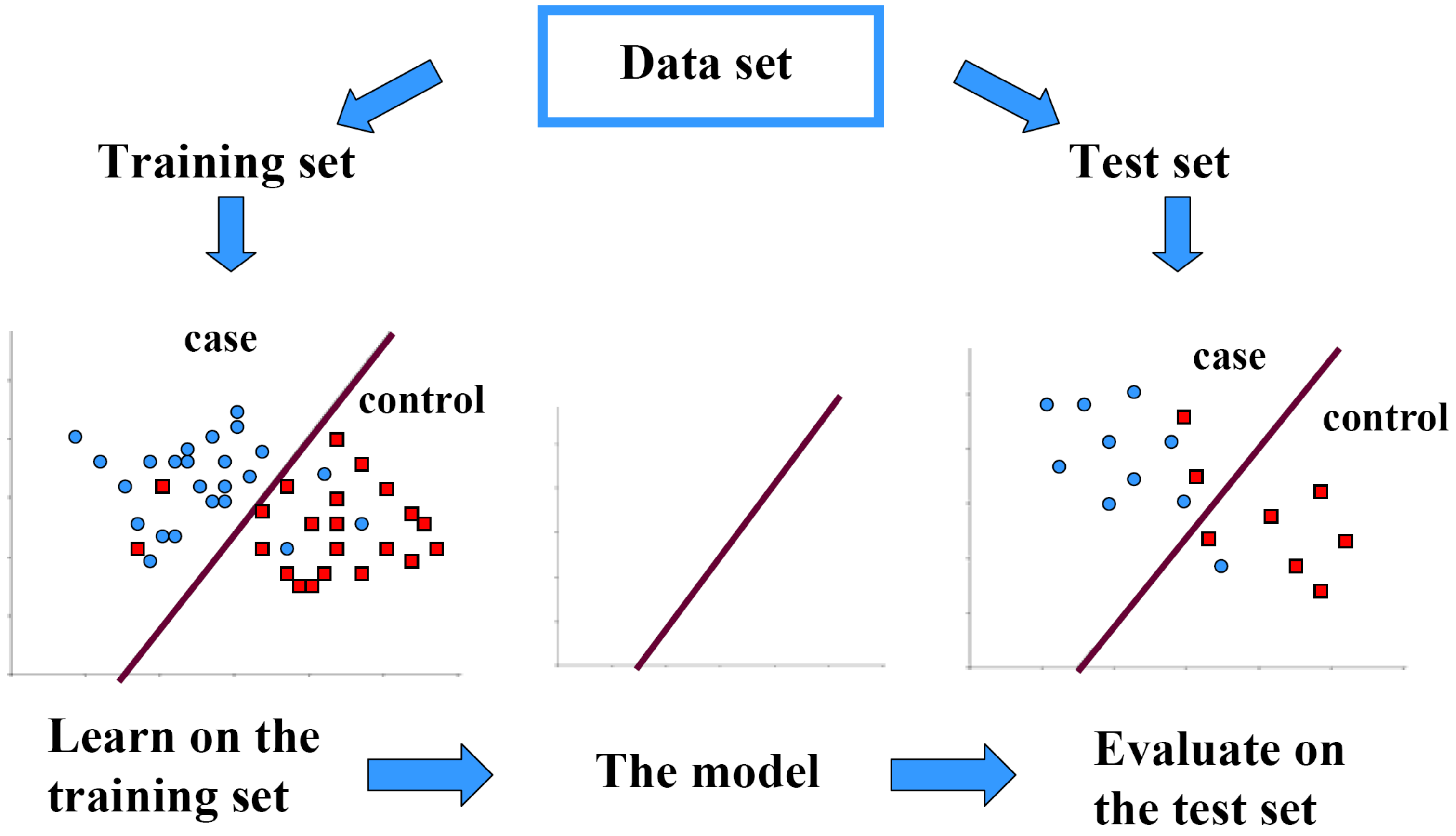
CSE 6363 (Fall 2019)

Lecture 11 Validation, KNN, Clustering

Dajiang Zhu, Ph.D.

Department of Computer Science and Engineering

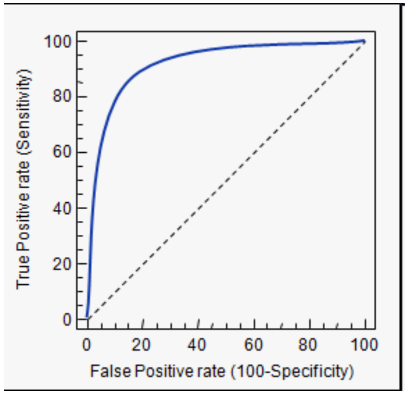
Evaluation for Classification



Evaluation Metrics

Confusion matrix: Records the percentages of examples in the testing set that fall into each group

		Actual: Case	Actual: Control
Prediction	Case	**TP** 0.3	**FP** 0.1
	Control	**FN** 0.2	**TN** 0.4



Misclassification error:

$$E = FP + FN$$

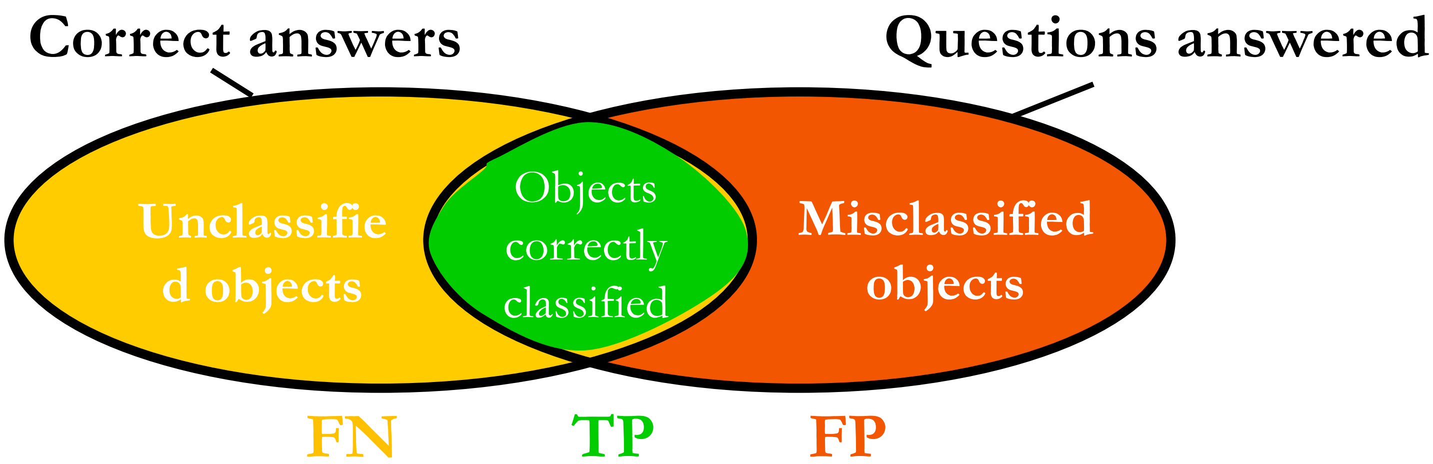
Sensitivity:

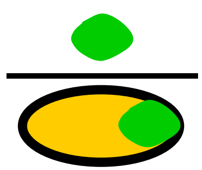
$$SN = \frac{TP}{TP + FN}$$

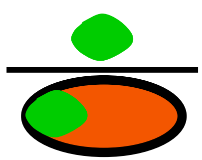
Specificity:

$$SP = \frac{TN}{TN + FP}$$

Precision-Recall



Recall=  = fraction of all objects correctly classified

Precision=  = fraction of all questions correctly answered

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Evaluation

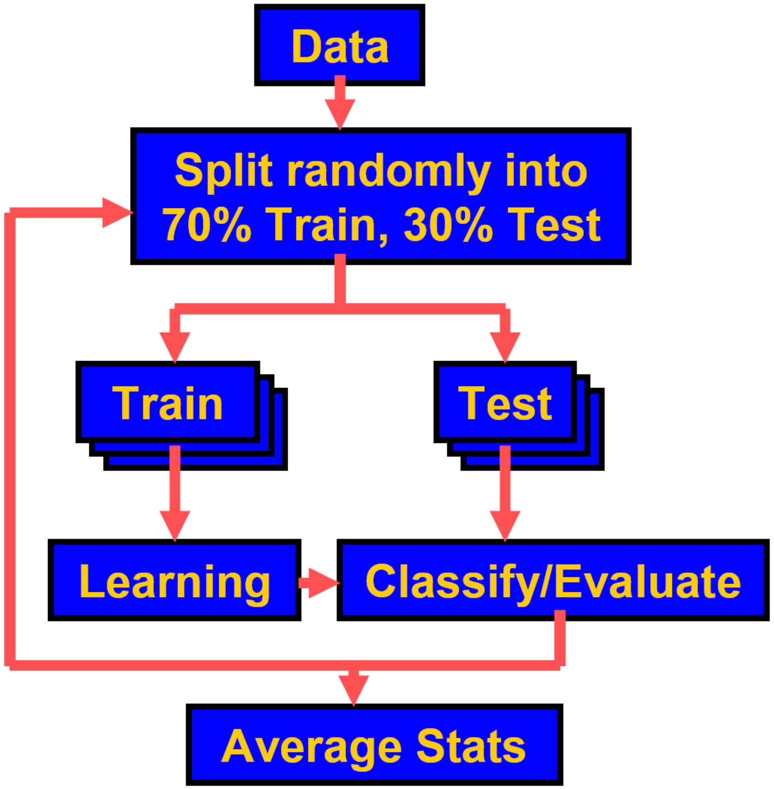
- Problem: if the sample size is relatively small one split may be lucky or unlucky hence biasing the statistics

- Solution: use multiple train/test splits and average their results

- Random resampling validation techniques:
  - random sub-sampling
  - k-fold cross-validation
  - bootstrap-based validation

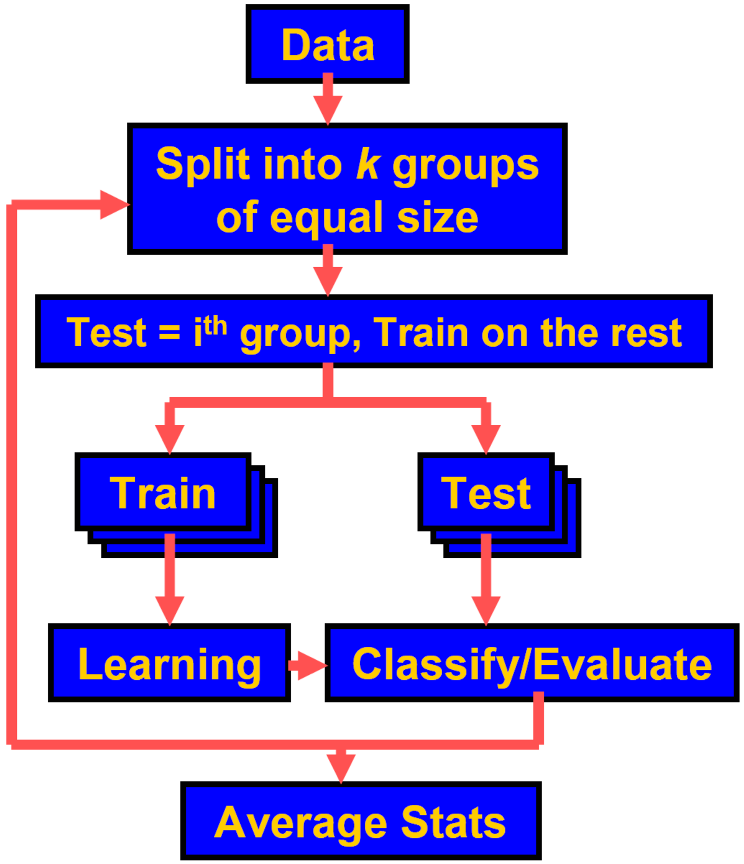
Random Sub-sampling

- Split the data into train and test set with some split ratio (typically 70:30)
- Repeat this k times for different random splits
- Average the results of statistics



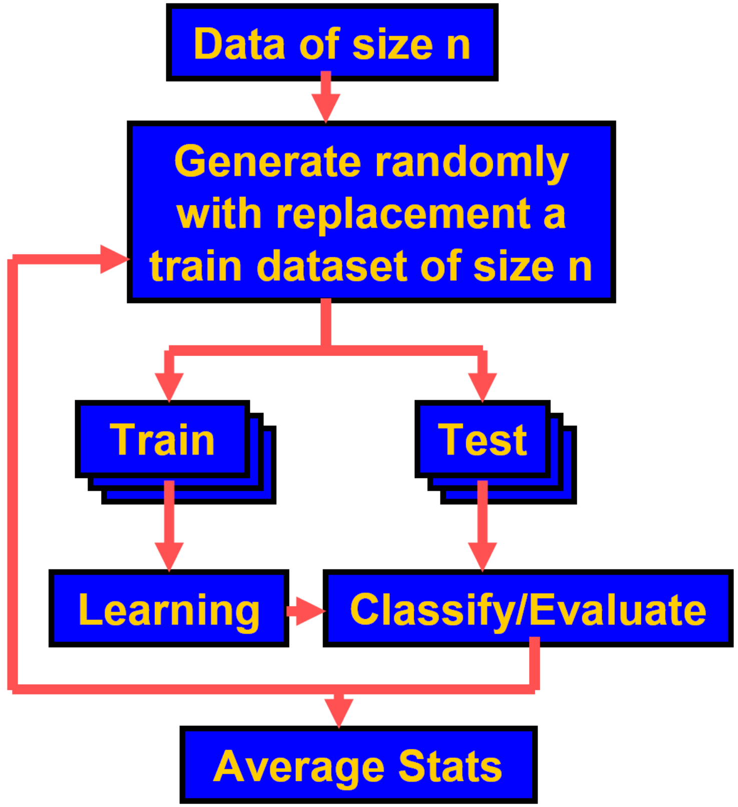
K-fold Cross-validation

- Split the data into k equal size groups
- Use each group once as a test set, and the remaining groups as the training set
- Repeat this k times for k groups
- Average the results of statistics



Bootstrap-based Validation

- Bootstrap technique - used primarily to estimate the sampling distribution of an estimator
- Generate randomly with replacement a training dataset of size n that equals the original data size
- Some examples are repeated in the training set, some are missing
- Build a test set from examples not used in the training set.



Parametric Models

A **parametric** model implements a very restricted family of functions $\mathbf{f}(\mathbf{x};\ \mathbf{w})$, leaving only a few parameters $\mathbf{w}$ to be learned. It thus expresses a strong presupposition (= prior) about the structure of the data.

Example: parametric density estimation

Assume density is isotropic Gaussian: $f(\mathbf{x}_k;\ \mathbf{w}) = N(\mathbf{x}_k;\ \boldsymbol{\mu}, \sigma^2\mathbf{I})$

=> need only determine optimal mean $\boldsymbol{\mu}^*$ and variance σ^{2*}

ML quickly gives

$$\boldsymbol{\mu}^* = \frac{1}{n}\sum_k \mathbf{x}_k\,, \qquad \sigma^{2*} = \frac{1}{n}\sum_k \|\mathbf{x}_k - \boldsymbol{\mu}\|^2.$$

Non-Parametric Models

Non-parametric models make only weak, general prior assumptions about the data, such as smoothness. $\mathbf{f(x; w)}$ is constructed directly over the memorized training data $\mathbf{X}$; the construction involves no or few parameters $\mathbf{w}$ to be learned.

Example: k-nearest neighbor methods

The model's output $\mathbf{f(x; w)}$ for some new datum $\mathbf{x}$ is calculated by combining (in some fixed way) the memorized responses for the k nearest neighbors of $\mathbf{x}$ in the training data. Example:

(regression) interpolate between nearest neighbor responses

(classification) take majority vote of nearest neighbor classes

K Nearest Neighbor Classifier

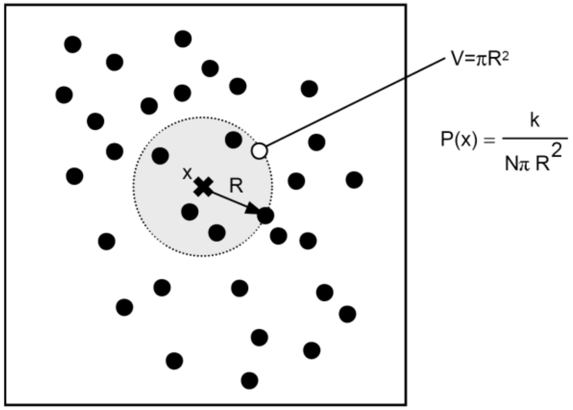
- **The kNN classifier is based on non-parametric density estimation techniques**
  - Let us assume we seek to estimate the density function P(x) from a dataset of examples
  - P(x) can be approximated by the expression

$$P(x) \cong \frac{k}{NV} \quad \text{where} \begin{cases} V \text{ is the volume surrounding } x \\ N \text{ is the total number of examples} \\ k \text{ is the number of examples inside } V \end{cases}$$

  - The volume V is determined by the D-dim distance $R_k^D(x)$ between x and its k nearest neighbor

$$P(x) \cong \frac{k}{NV} = \frac{k}{N \cdot c_D \cdot R_k^D(x)}$$

    - Where c_D is the volume of the unit sphere in D dimensions



K Nearest Neighbor Classifier

- **We use the previous result to estimate the posterior probability**
  - The unconditional density is, again, estimated with

$$P(x \mid \omega_i) = \frac{k_i}{N_i V}$$

  - And the priors can be estimated by

$$P(\omega_i) = \frac{N_i}{N}$$

  - The posterior probability then becomes

$$P(\omega_i \mid x) = \frac{P(x \mid \omega_i) P(\omega_i)}{P(x)} = \frac{\frac{k_i}{N_i V} \cdot \frac{N_i}{N}}{\frac{k}{NV}} = \frac{k_i}{k}$$

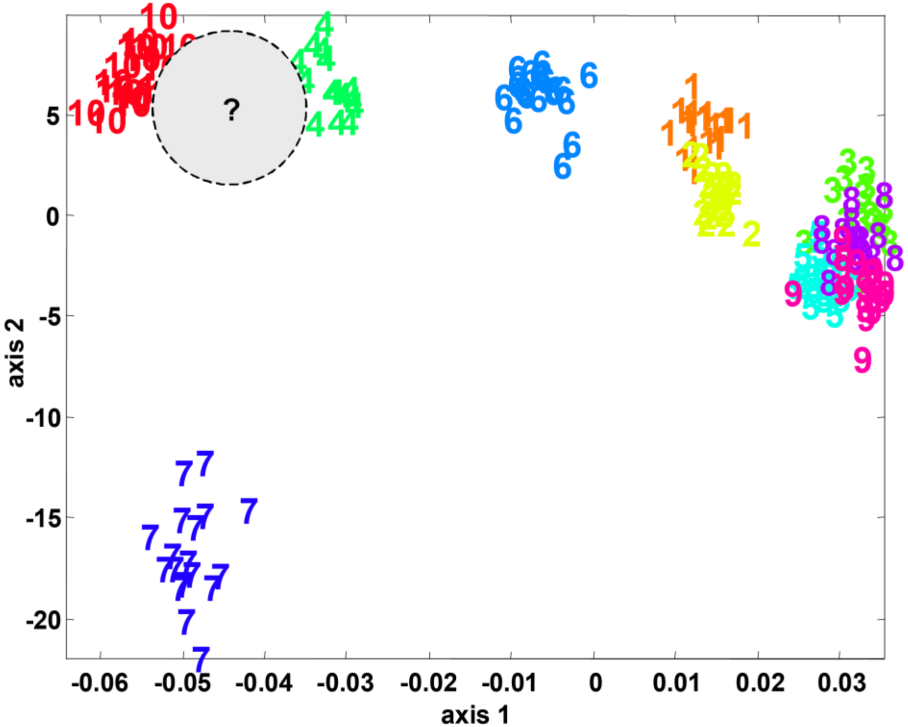
  - Yielding discriminant functions

$$\boxed{g_i(x) = \frac{k_i}{k}}$$

    - This is known as the k Nearest Neighbor classifier

K Nearest Neighbor Classifier

- **The kNN classifier is a very intuitive method**
  - Examples are classified based on their similarity with training data
    - For a given unlabeled example $x_u \in \Re^D$, find the k "closest" labeled examples in the training data set and assign x_u to the class that appears most frequently within the k-subset
- **The kNN only requires**
  - An integer k
  - A set of labeled examples
  - A measure of "closeness"

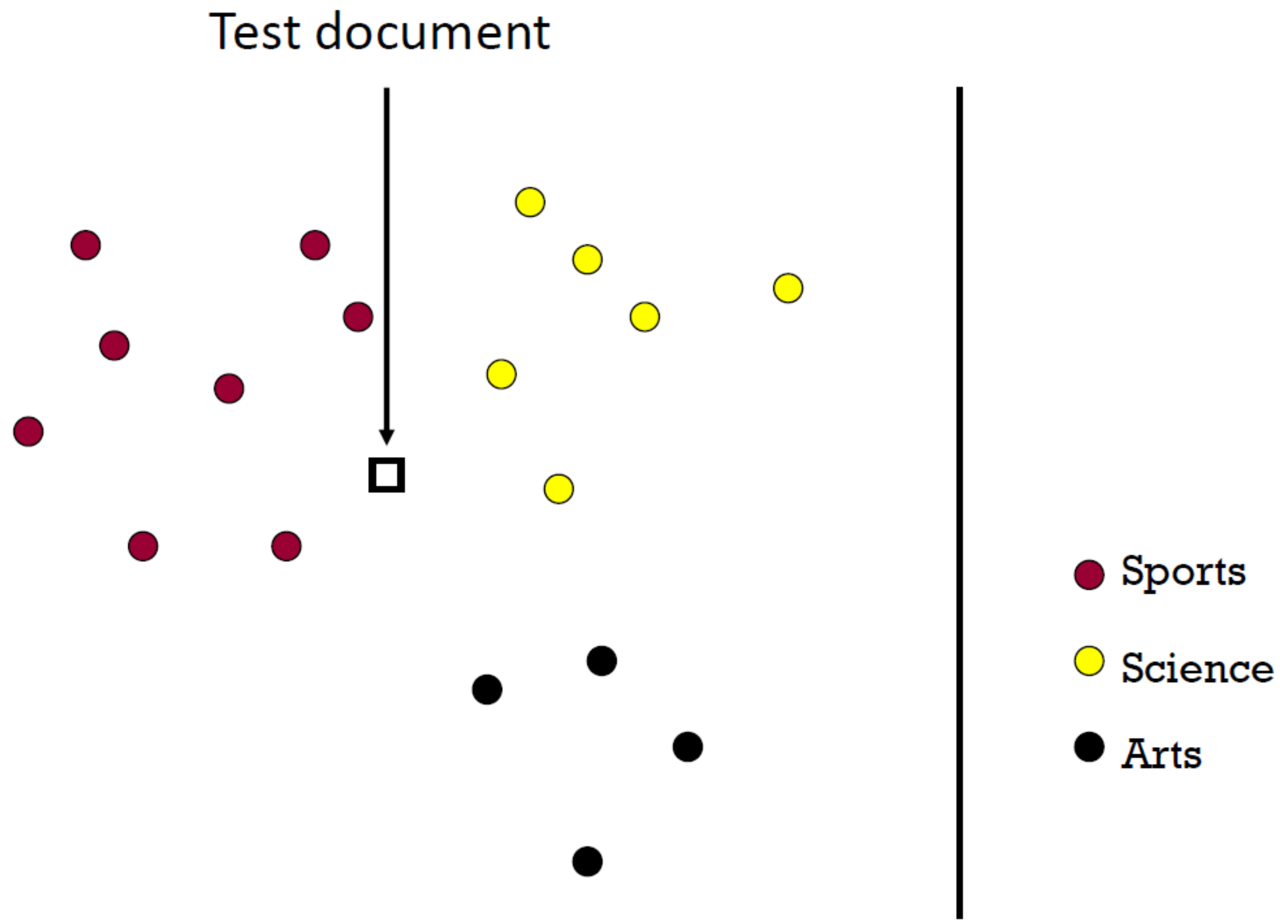


K Nearest Neighbor Classifier

$$\text{k-NN Classifier: } \widehat{f}_{kNN}(x) = \arg\max_y \ \widehat{p}_{kNN}(x|y)\widehat{P}(y)$$

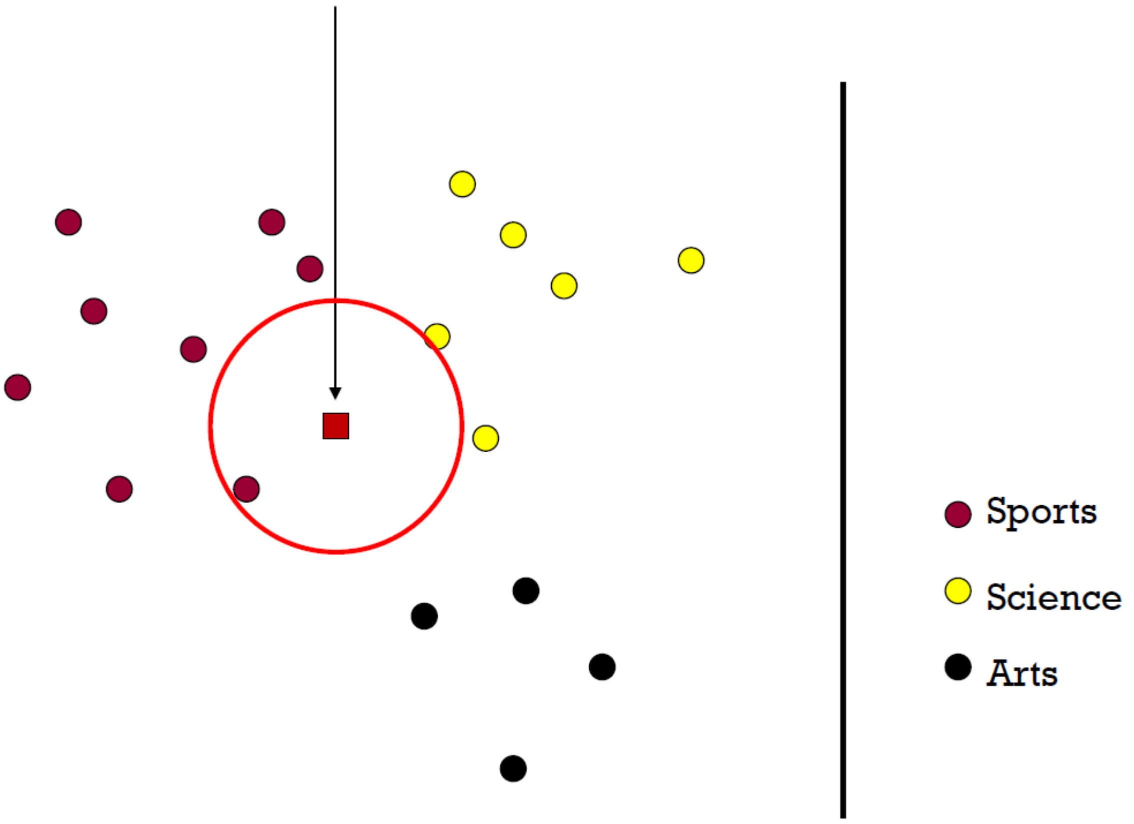
$$= \arg\max_y \ k_y \quad \textbf{(Majority vote)}$$

K Nearest Neighbor Classifier -Example



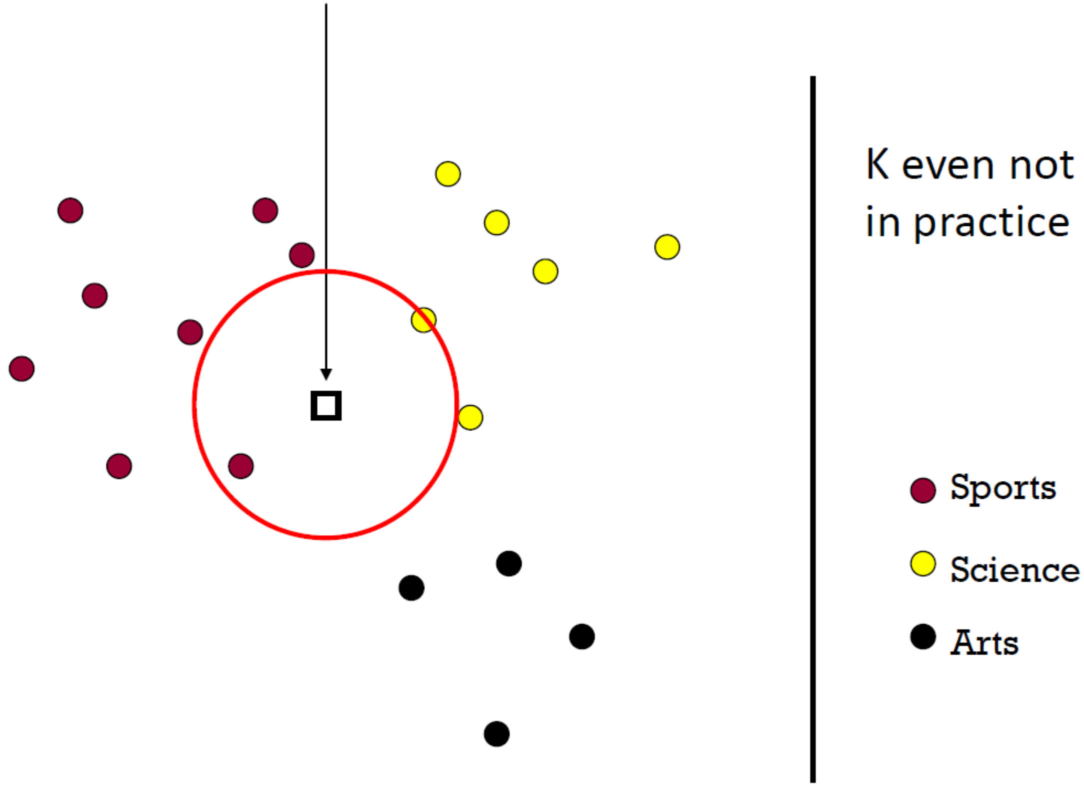
K Nearest Neighbor Classifier -Example

1-Nearest Neighbor (kNN) classifier



K Nearest Neighbor Classifier -Example

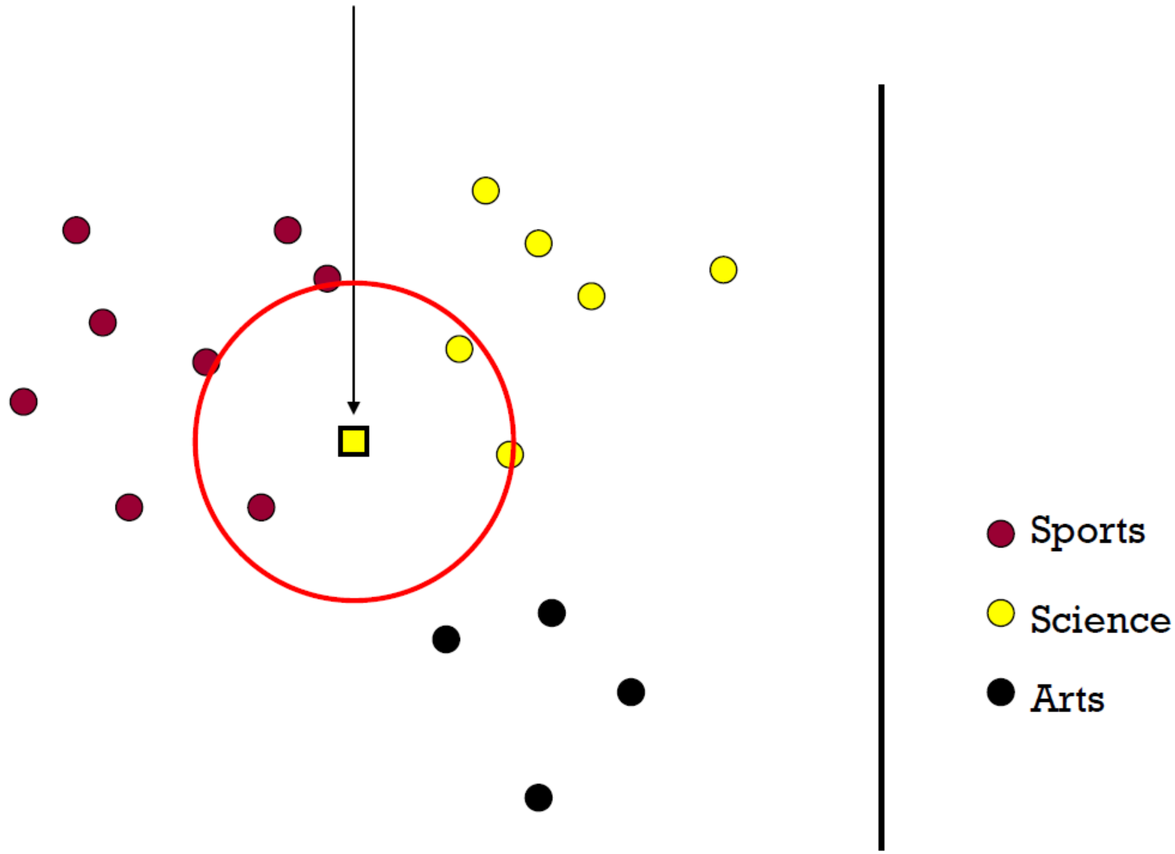
2-Nearest Neighbor (kNN) classifier



K even not used in practice

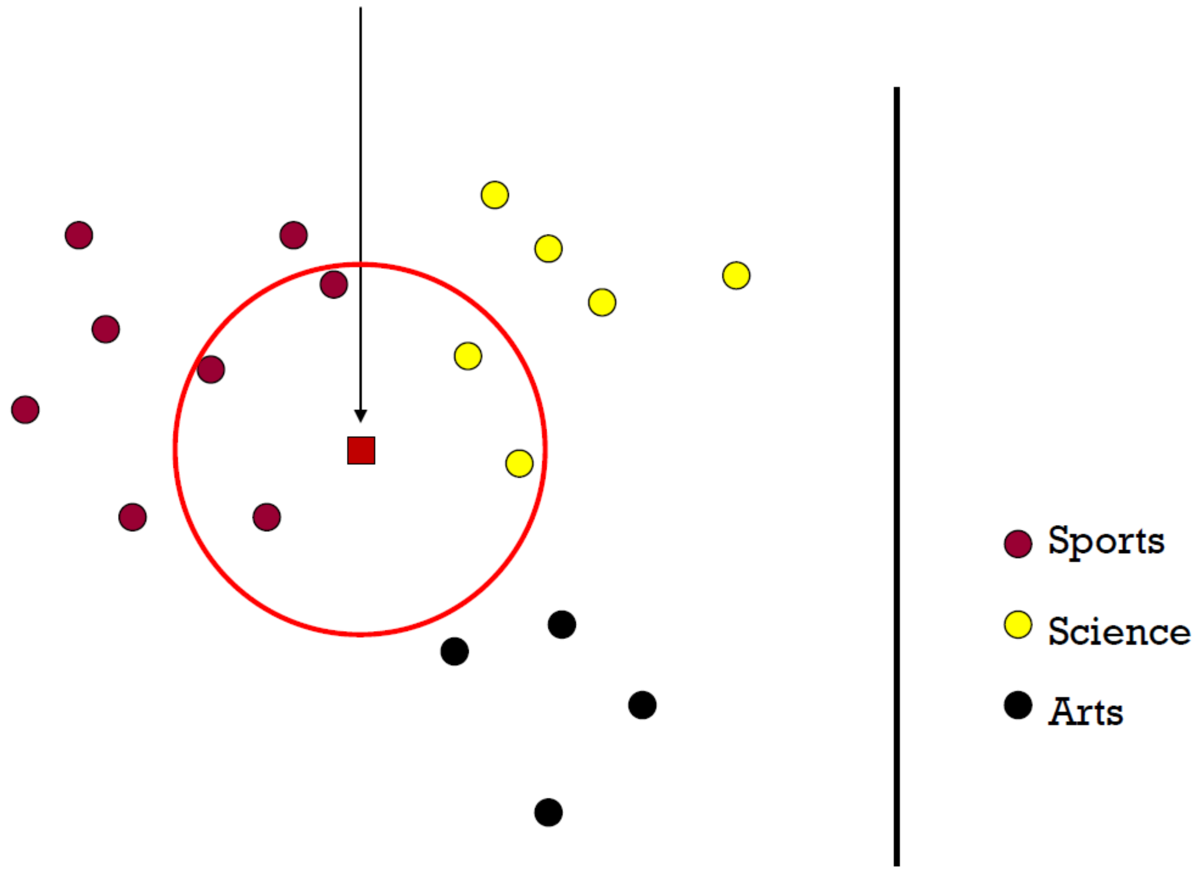
K Nearest Neighbor Classifier -Example

3-Nearest Neighbor (kNN) classifier



K Nearest Neighbor Classifier -Example

5-Nearest Neighbor (kNN) classifier



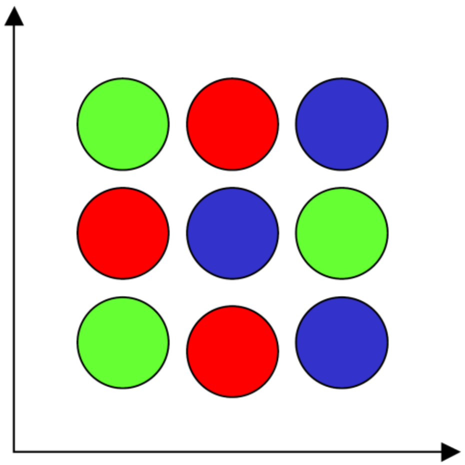
What is the best K

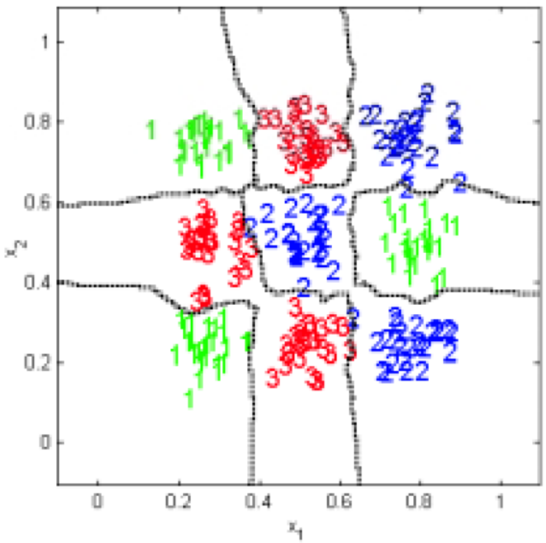
Bias-variance tradeoff

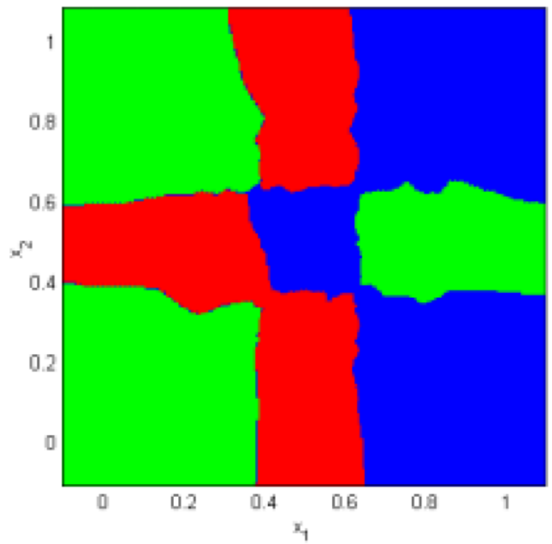
- Larger K => predicted label is more stable
- Smaller K => predicted label is more accurate

kNN in Action: Example 1

- **We generate data for a 2-dimensional 3-class problem, where the class-conditional densities are multi-modal, and non-linearly separable**
- **We used kNN with**
  - k = five
  - Metric = Euclidean distance

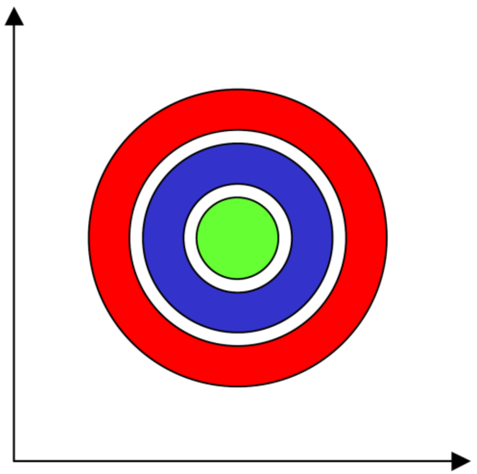


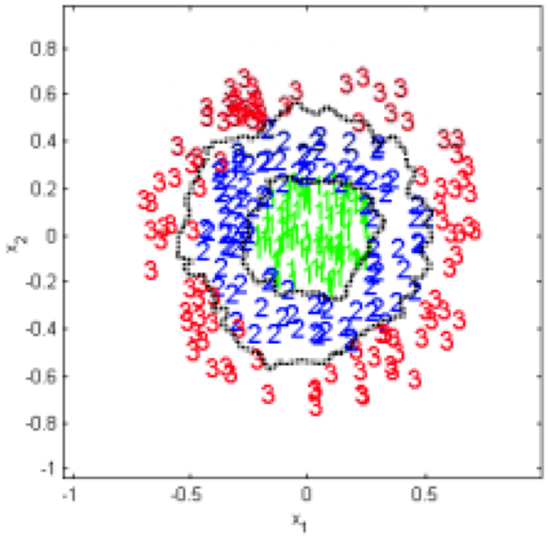


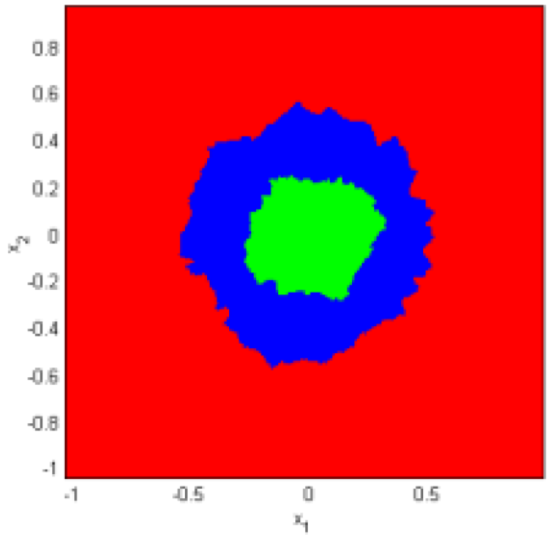


kNN in Action: Example 2

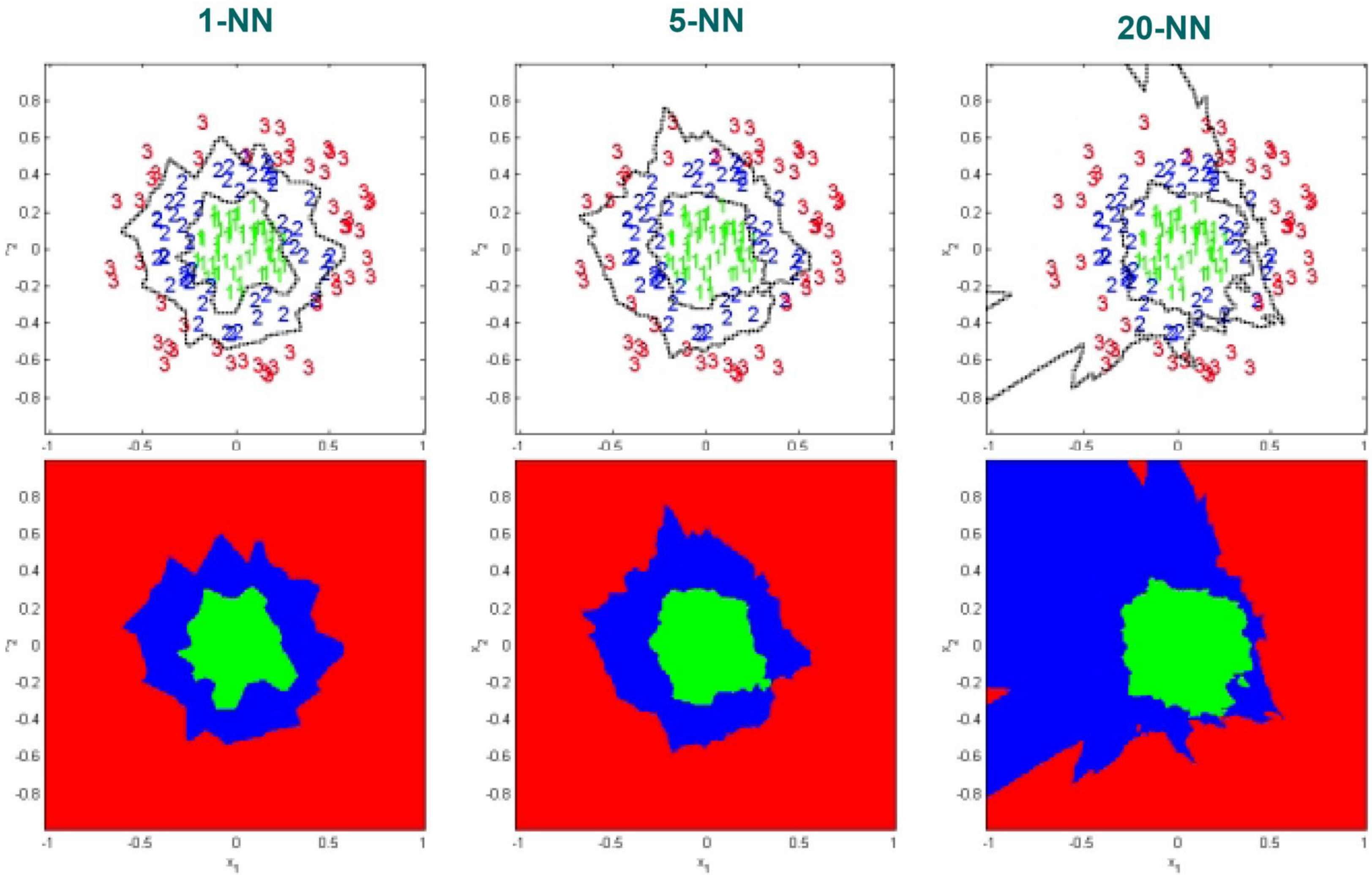
- **We generate data for a 2-dim 3-class problem, where the likelihoods are unimodal, and are distributed in rings around a common mean**
  - These classes are also non-linearly separable
- **We used kNN with**
  - k = five
  - Metric = Euclidean distance







kNN versus 1NN



kNN in Action: Example 3

Dataset

- 20 News Groups (20 classes)
- Download :(http://people.csail.mit.edu/jrennie/20Newsgroups/)
- 61,118 words, 18,774 documents
- Class labels descriptions

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

kNN in Action: Example 3

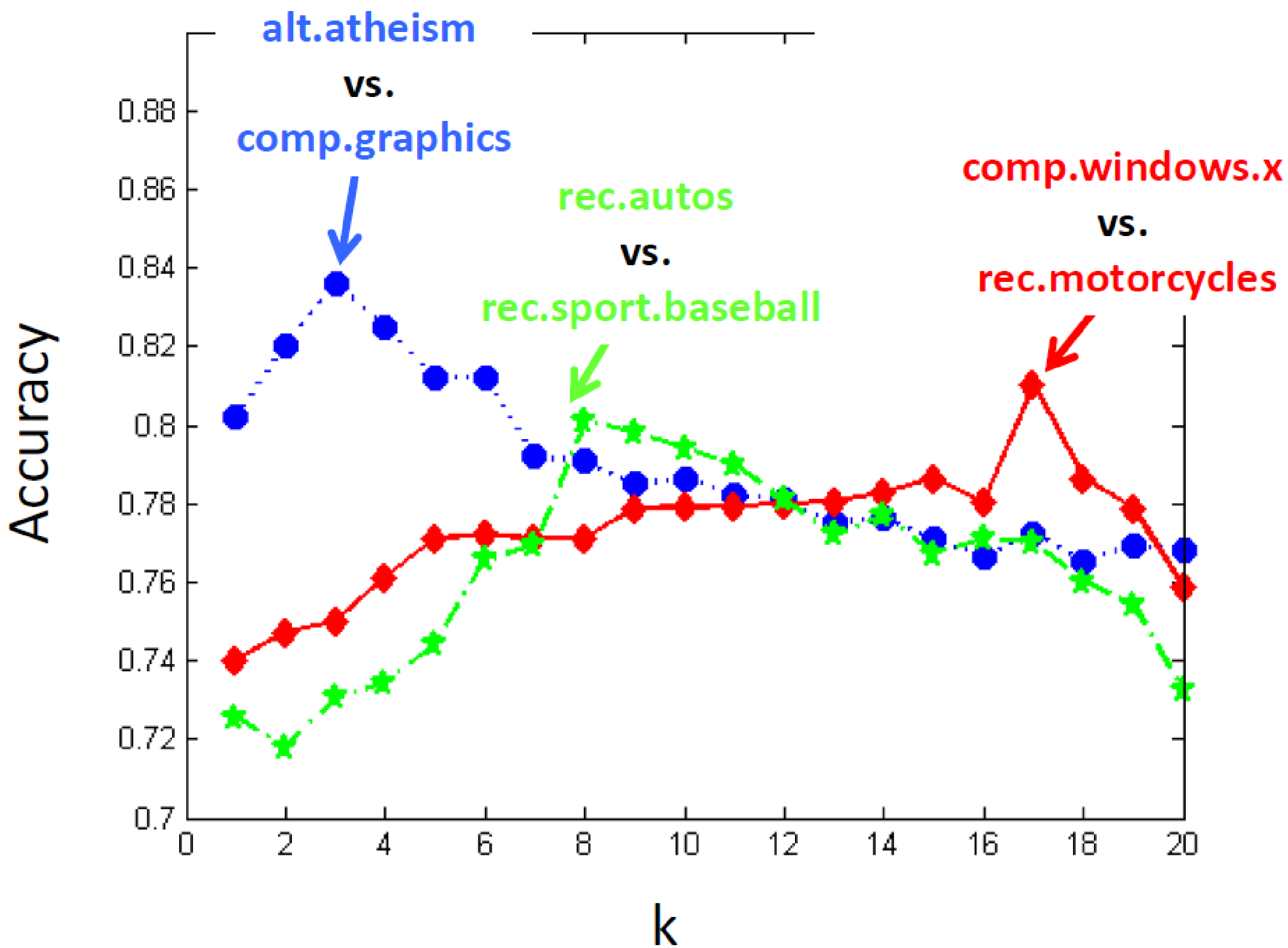
Training/Test Sets:

- 50%-50% randomly split.
- 10 runs
- report average results

Evaluation Criteria:

$$Accuracy = \frac{\sum_{i \in test\ set} \mathrm{I}(predict_i == true\ label_i)}{\#\ of\ test\ samples}$$

kNN in Action: Example 3



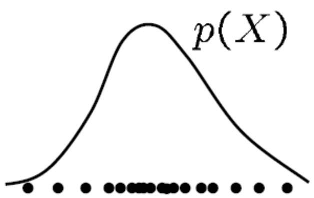
Unsupervised Learning

"Learning from unlabeled/unannotateddata"
(without supervision)

Training data $\{X_i\}_{i=1}^n$ → **Learning Algorithm** → Prediction rule $\widehat{f}_n$

What can we predict from unlabeled data?

- **Density estimation**



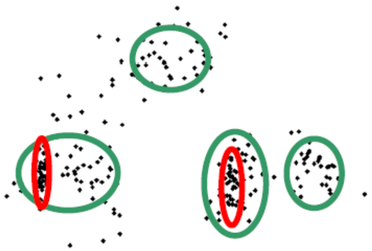
Unsupervised Learning

"Learning from unlabeled/unannotateddata"
(without supervision)

Training data $\{X_i\}_{i=1}^{n}$ → **Learning Algorithm** → Prediction rule $\widehat{f}_n$

What can we predict from unlabeled data?

- **Density estimation**
- **Groups or clusters in the data**



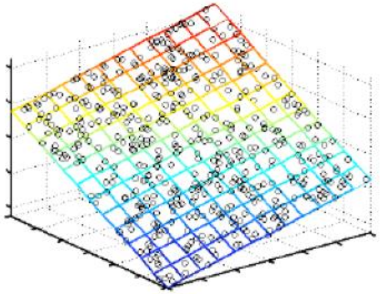
Unsupervised Learning

"Learning from unlabeled/unannotateddata"
(without supervision)

Training data $\{X_i\}_{i=1}^{n}$ → **Learning Algorithm** → Prediction rule $\widehat{f}_n$

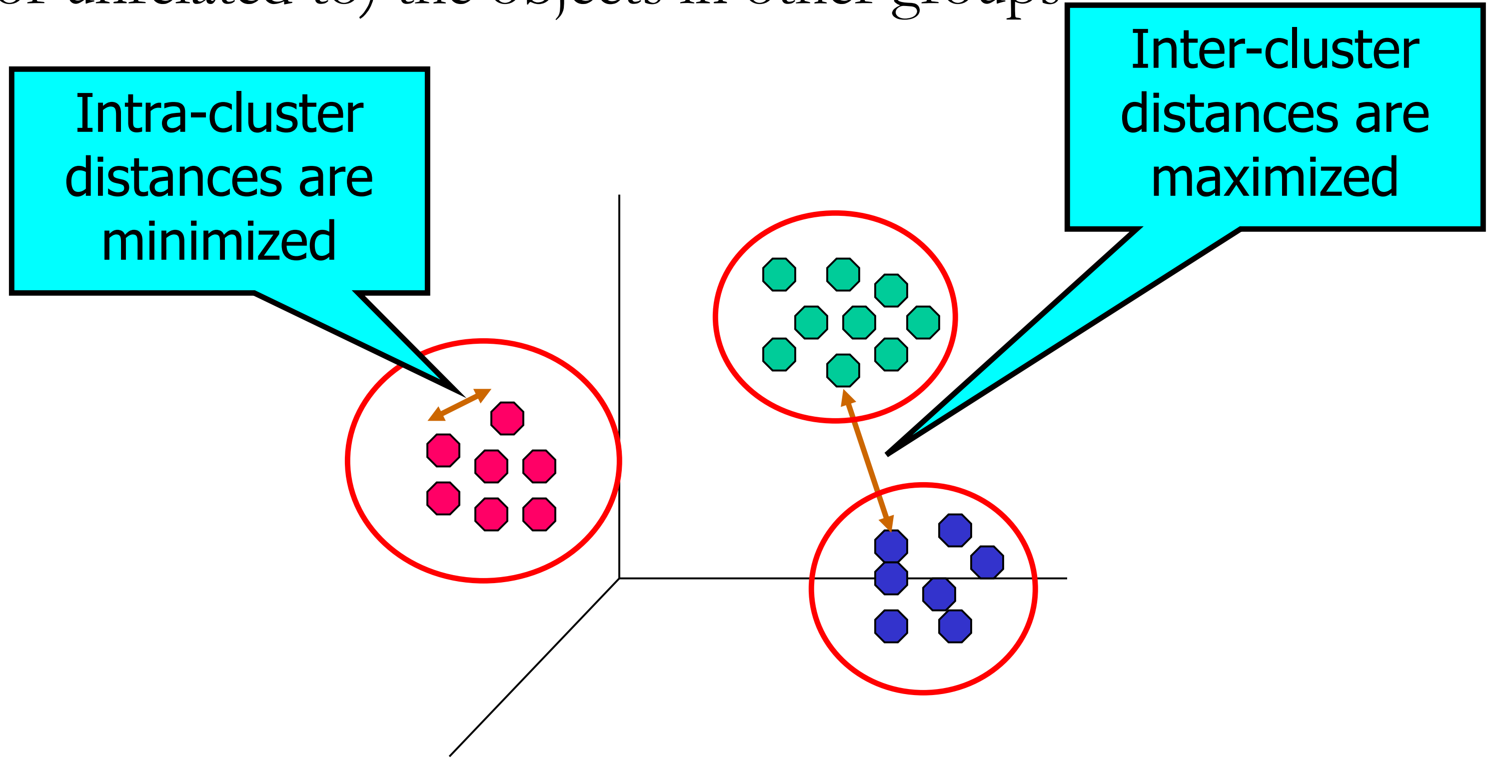
What can we predict from unlabeled data?

- **Density estimation**
- **Groups or clusters in the data**
- **Low-dimensional structure (PCA)**

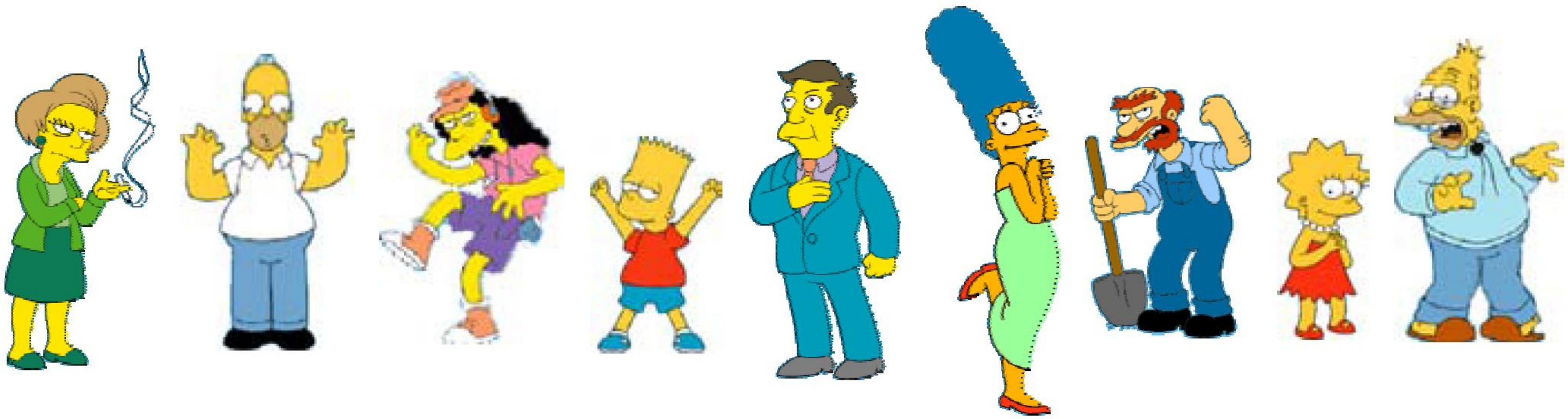


What is Cluster Analysis?

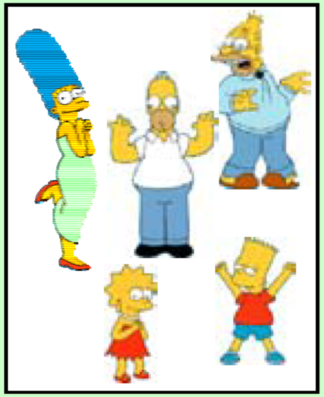
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



What is a natural grouping among these objects?

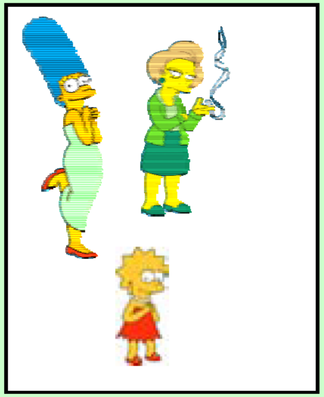


Clustering is subjective



Simpson's Family

School Employees



Females

Males

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

Four Clusters

What is Similarity?

- The quality or state of being similar
  likeness, resemblance - Webster's Dictionary

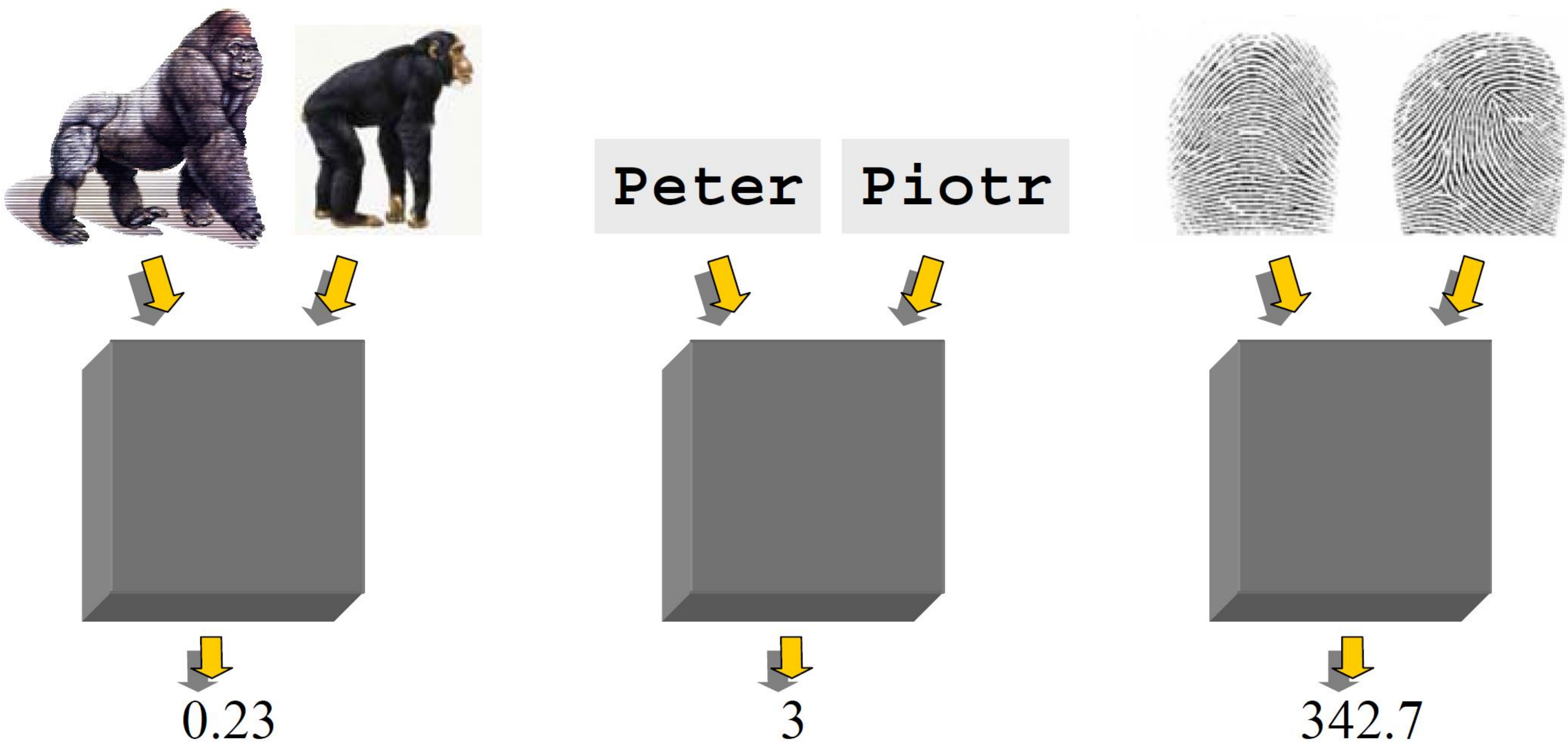
Hard to define!

"similarity"

"distance "

Defining Distance Measures

Definition Let $O1$ and $O2$ be two objects from the universe of possible objects. The distance dissimilarity between $O1$ and $O2$ is a real number denoted by $D(O1,O2)$



What properties should a distance measure have?

- D(A,B) = D(B,A) *Symmetry*

  Otherwise you could claim: Alex looks like Bob, but Bob looks nothing like Alex.

- D(A,A) = 0 *Constancy of Self-Similarity*

  Otherwise you could claim: Alex looks more like Bob, than Bob does.

- D(A,B) = 0 If A= B *Positivity Separation*

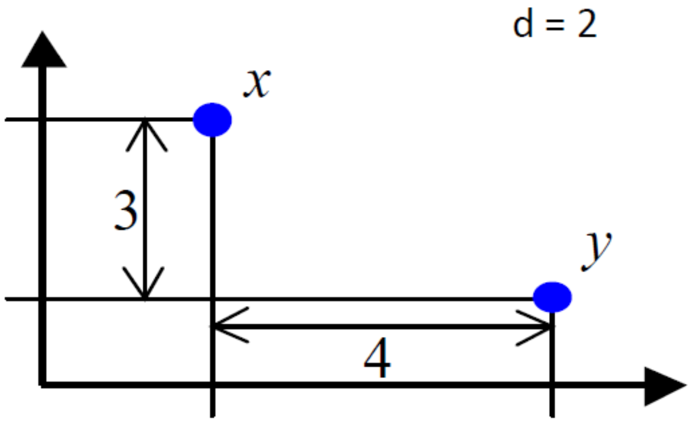
  Otherwise there are objects in your world that are different, but you cannot tell apart.

- D(A,B) <= D(A,C) + D(B,C) *Triangular Inequality*

  Otherwise you could claim: Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl

Distance metrics

$x = (x_1, x_2, \ldots, x_p)$
$y = (y_1, y_2, \ldots, y_p)$



Euclidean distance $$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$ **5**

Manhattan distance $$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$ **7**

Sup-distance $$d(x,y) = \max_{1 \le i \le p} |x_i - y_i|$$ **4**

Distance metrics

$x = (x_1, x_2, \ldots, x_p)$
$y = (y_1, y_2, \ldots, y_p)$

Random vectors (e.g. expression levels of two genes under various drugs)

Pearson correlation coefficient

$$\rho(x, y) = \frac{\sum_{i=1}^{p} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p} (x_i - \bar{x})^2 \times \sum_{i=1}^{p} (y_i - \bar{y})^2}}$$

where $\bar{x} = \frac{1}{p} \sum_{i=1}^{p} x_i$ and $\bar{y} = \frac{1}{p} \sum_{i=1}^{p} y_i$.