slides originally by
Dr. Richard Burns,
modified by
Dr. Stephanie Schwartz

NEAREST NEIGHBORS

CSCI 452: Data Mining

Similarity and Dissimilarity Measures

- Used by a number of data mining techniques:
  - Nearest neighbors
  - Clustering
  - Anomaly detection

How to measure "proximity"?

- Proximity: similarity or dissimilarity between two objects
  - Similarity: numerical measure of the degree to which two objects are *alike*
    - Usually in range [0,1]
      - 0 = no similarity
      - 1 = complete similarity
  - Dissimilarity: measure of the degree in which two objects are *different*

Similarity/Dissimilarity for Simple Attributes

- *p* and *q* are the attribute values for two data

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\|p-q\|}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\|p-q\|}{n-1}$
Interval or Ratio	$d = \|p - q\|$	$s = -d$, $s = \frac{1}{1+d}$ or

Dissimilarities between Data Objects

- A common measure for the proximity between two objects is the Euclidean Distance:

$$d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

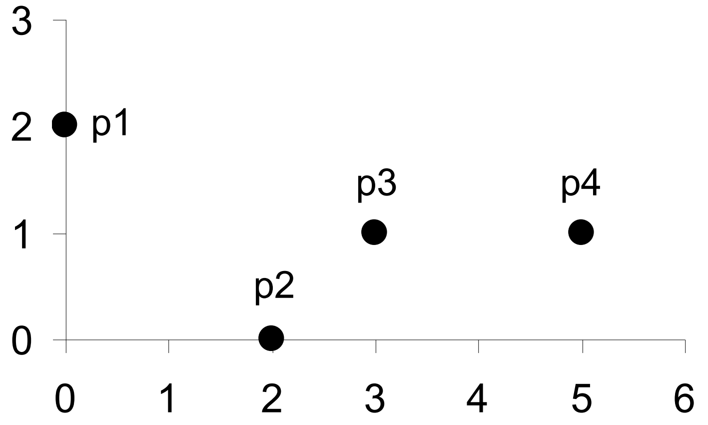
x and *y* are two data objects
n dimensions
Standardization necessary, if scales differ

  - In high school, we typically used this for calculating the distance between two objects, when there were only two dimensions.
  - Defined for one-dimension, two-dimensions, three-dimensions, …, any n-dimensional space

Distance Matrix

- Once a distance metric is chosen, the proximity between all of the objects in the dataset can be computed
  - Represented in a distance matrix
    - Pairwise distances between points

Distance Matrix



Euclidean Distance.

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Dissimilarities between Data Objects

- Typically the <u>Euclidean Distance</u> is used as a first choice when applying <u>Nearest Neighbors</u> and <u>Clustering</u>
- Other distance metrics:
  - Generalized by the <u>Minkowski</u> distance metric:

n is the number of dimensions (attributes) and

x_k and y_k are the kth attributes of objects x and y.

$$d(x,y) = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^r}^{\,1/r}$$

r is a parameter

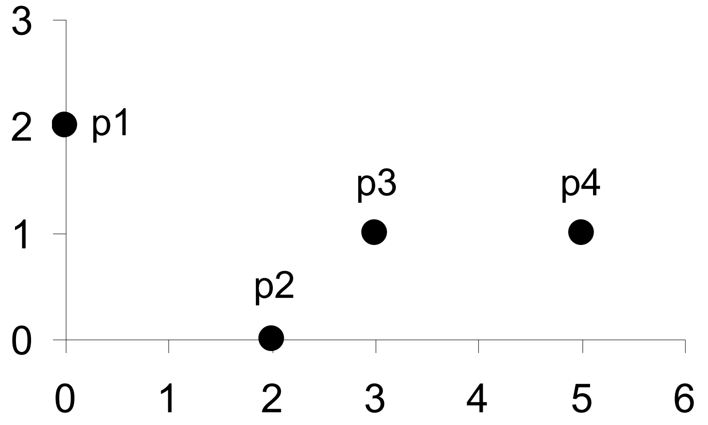
Dissimilarities between Data Objects

- **Minkowski Distance Metric:**
  - r = 1 (L_1 norm)
    - "Manhattan, taxicab" Distance
  - r = 2 (L_2 norm)
    - Euclidean Distance
  - $r \rightarrow \infty$. "supremum", L_{max}, L_∞ norm distance.
    - the max difference between any components of the vectors

$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

The *r* parameter should not be confused with the number of attributes/dimensions *n*

Minkowski Distance



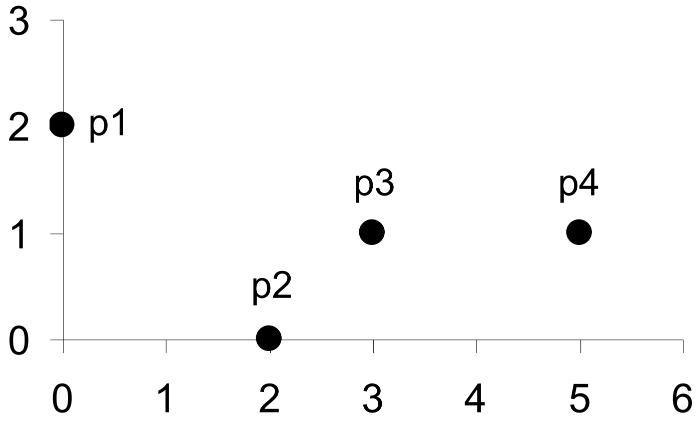
L1 norm distance. “Manhattan” Distance.

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2 norm distance. Euclidean Distance.

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Minkowski Distance



L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Using Weights

- So far, all attributes treated equally when computing proximity
- In some situations: some features are more important than others
  - Decision up to the analyst
- Modified Minkowski distance definition to include weights:

$$d(x,y) = \sqrt[1/r]{\sum_{k=1}^{n} \omega_k \left| x_k - y_k \right|^r}$$

Standardization

- In other situations, may want all features to be treated “equally”
  - One feature doesn’t dominate another (income vs age)
- Common treatment to all variables:
  - <u>Standardize</u> each variable:
    - Mean = 0
    - Standard Deviation = 1

Eager Learner Models

- So far in this course, we've performed prediction by:
  1. Taking a dataset
  2. Learning a model
  3. Using model to classify/predict test instances

- Sometimes called eager learners:
  - Designed to learn a model that maps the input attributes to the class label, as soon as training data becomes available.

Lazy Learner Models

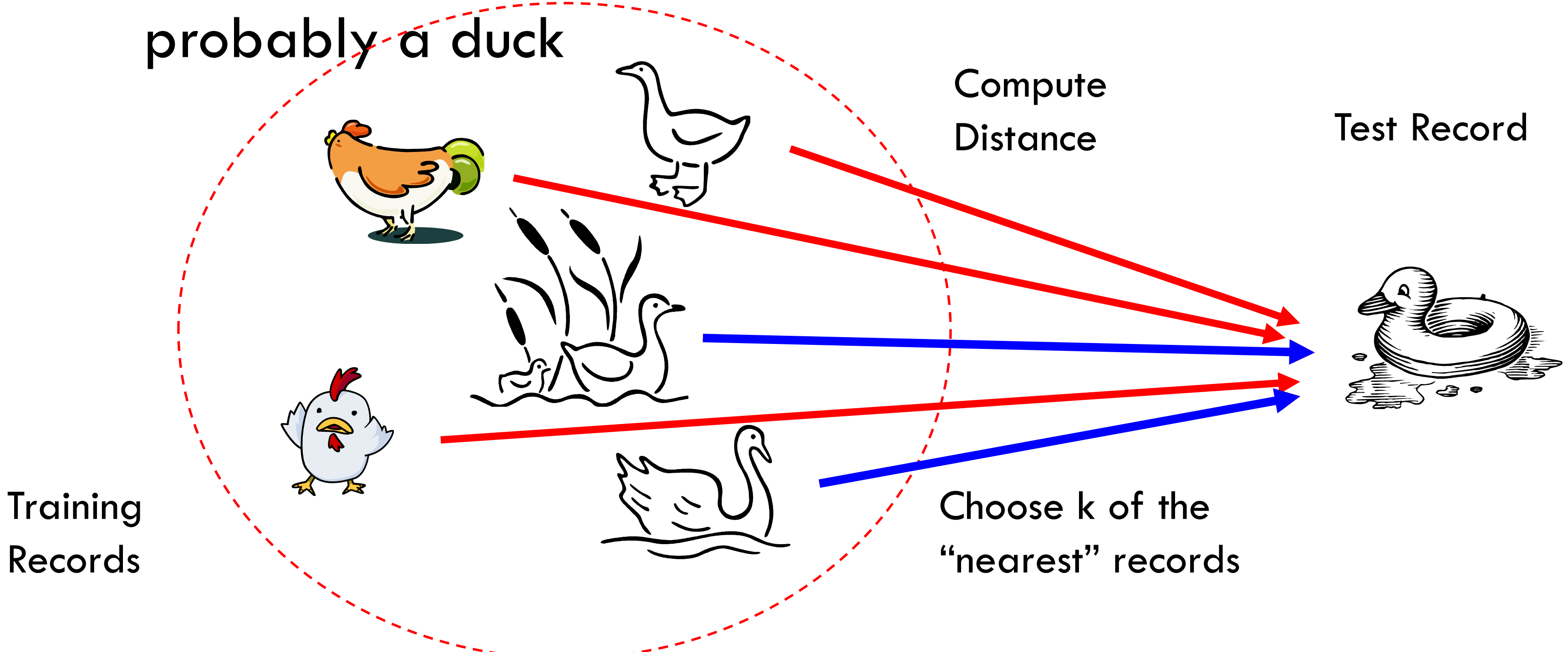
- Opposite strategy:
  - *Delay* process of modeling the training data, until it is necessary to classify/predict a test instance.
  - There is no “training” period, but each classification can be relatively expensive
  - Example:
    - Nearest neighbors

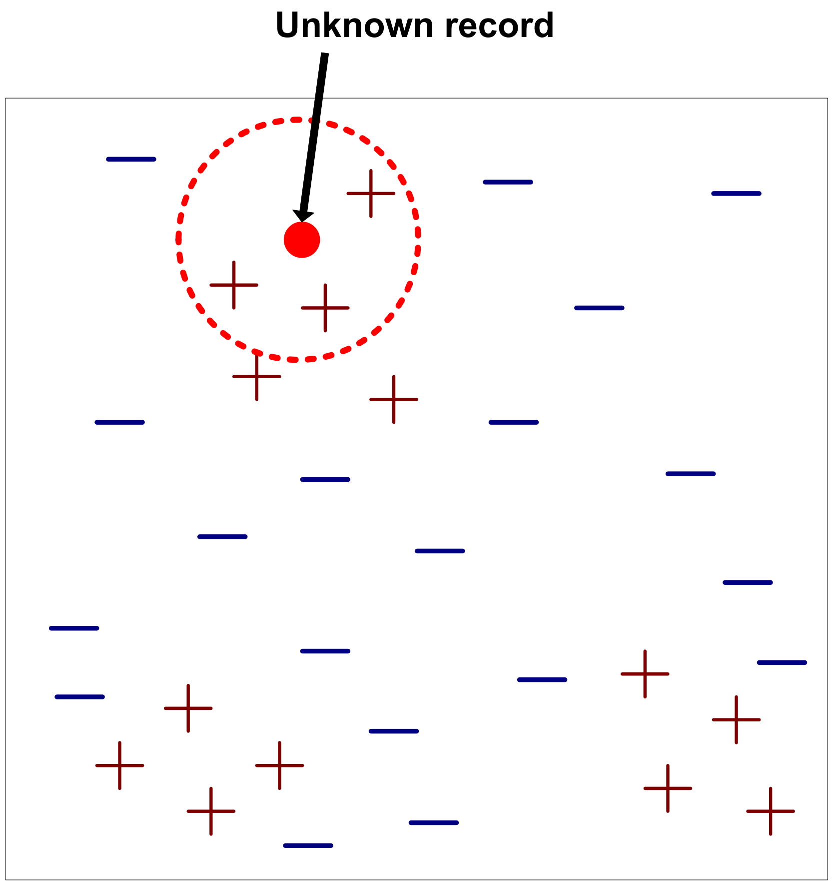
Nearest Neighbors

- *k*-nearest neighbors
  - k = parameter, chosen by analyst
  - For a given test instance, use the k "closest" points (nearest neighbors) for performing classification
    - "closest" points: defined by some proximity metric, such as Euclidean Distance

Rationale

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





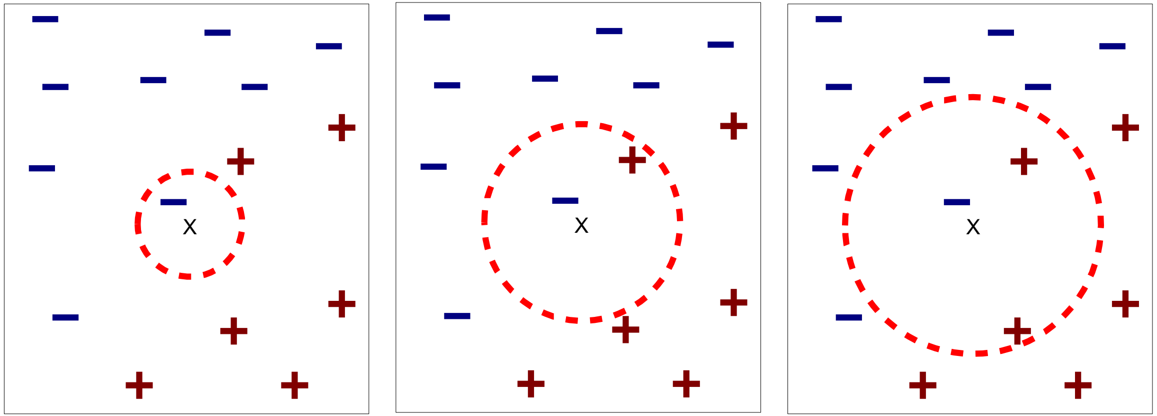
- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor

- The *k*-nearest neighbors of a given example *x* are the *k* points that are closest to *x*.

- Classification changes depending on the chosen *k*
- Majority Voting



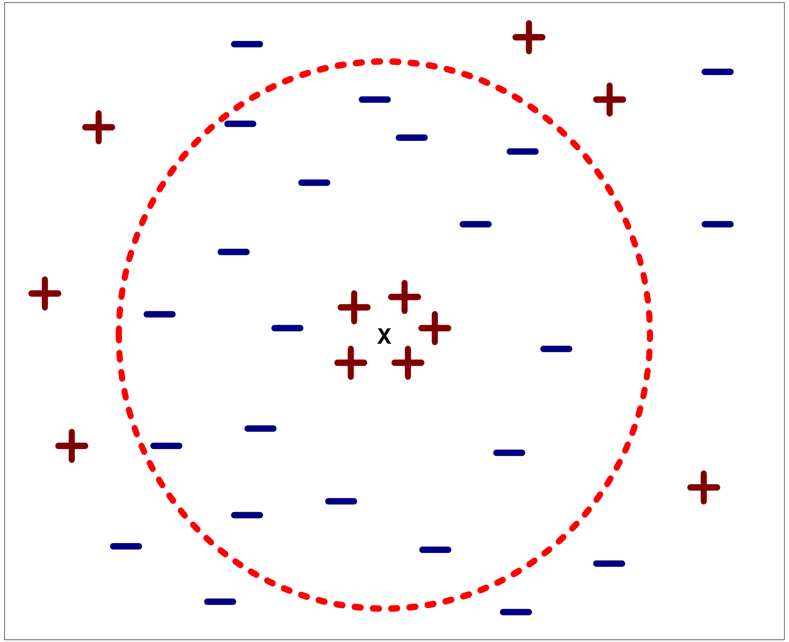
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Tie?

- Randomly choose classification.
- For binary problems, usually an odd *k* is used to avoid ties.

Choosing the right k

- If k is too small, sensitive to noise points in the training data
  - Susceptible to overfitting
- If k is too large, neighborhood may include points from other classes
  - Susceptible to misclassification



Algorithm

- Can't have a CS class without pseudocode!

Algorithm 5.2 The k-nearest neighbor classification algorithm.

Let k be the number of nearest neighbors and D be the set of training examples.
for each test example $z = (\mathbf{x}', y')$ **do**
    Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every example, $(\mathbf{x}, y) \in D$.
    Select $D_z \subseteq D$, the set of k closest training examples to z.
    $y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$
end for

Computational Issues?

- Computation can be costly if the number of training examples is large.
- Efficient indexing techniques are available to reduce the amount of computations needed when finding the nearest neighbors of a test example

Majority Voting

- Simply majority: Every neighbor has the same impact on the classification:

$$\text{Majority Voting: } y' = \arg\max_{v} \sum_{(x_i, y_i) \in D_z} I(v = y_i)$$

- Distance-weighted:
  - Far away neighbors have a weaker impact on the classification.

$$\text{Distance-Weighted Voting: } y' = \arg\max_{v} \sum_{(x_i, y_i) \in D_z} \omega_i \times I(v = y_i)$$

Scaling Issues

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

Income will dominate if these variables aren't standardized.

Final Thoughts on Nearest Neighbors

- Nearest-neighbors classification is part of a more general technique called instance-based learning
  - Use specific instances for prediction, rather than a model
- Nearest-neighbors is a lazy learner
  - Performing the classification can be relatively computationally expensive
  - *(no model is learned up-front)*

Classifier Comparison

Eager Learners

- Decision Trees, SVMs
- Model Building: *potentially slow*
- Classifying Test Instance: *fast*

Lazy Learners

- Nearest Neighbors
- Model Building: *fast (because there is none!)*
- Classifying Test Instance: *slow*

Classifier Comparison

Eager Learners

- Decision Trees, SVMs
- *finding a global model that fits the entire input space*

Lazy Learners

- Nearest Neighbors
- *classification decisions are made locally (small k values), are are more susceptible to noise*

References

- *Introduction to Data Mining*, 1st edition, Tan et al.
- *Data Mining and Business Analytics with R*, 1st edition, Ledolter