CS 484
Data Mining

Data

Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise
- Techniques
  - Principle Component Analysis
  - Singular Value Decomposition
  - Others: supervised and non-linear techniques

Principal Component Analysis

- Goal of PCA
  – To reduce the number of dimensions.
  – Transfer interdependent variables into single and independent components.
- What does PCA do ?
  – Transforms the data into a lower dimensional space, by constructing dimensions that are linear combinations of the input dimensions/ features.
  – Find independent dimensions along which data have the largest variance.

Goal is to find a projection that captures the largest amount of variation in data

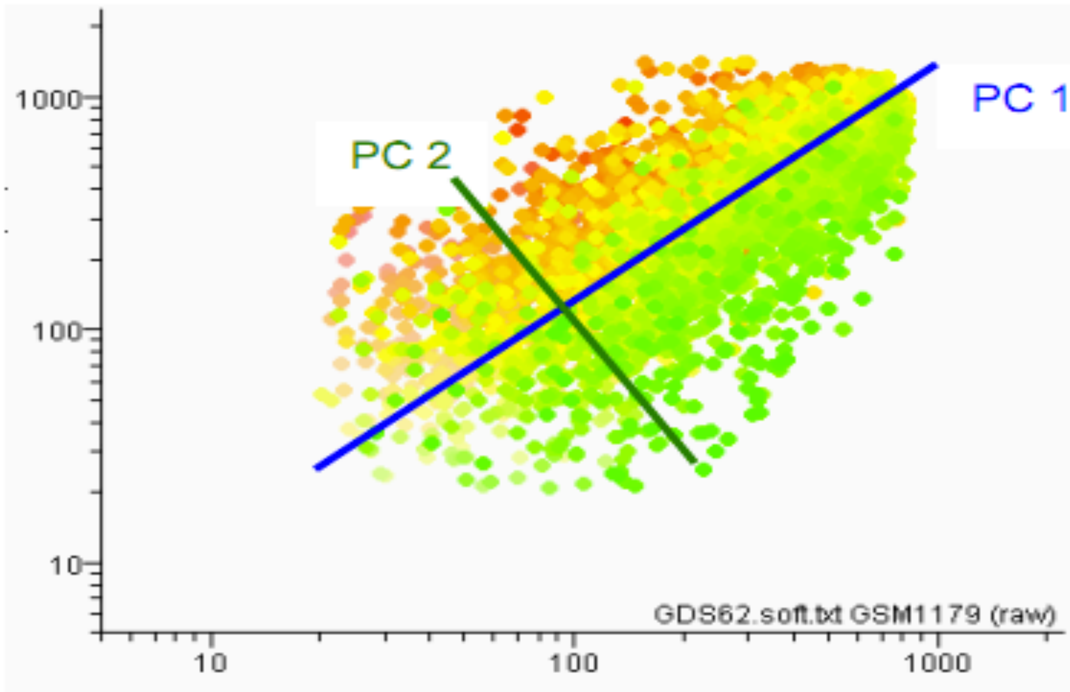


Fig 1: Football-shaped data set with two main components.

http://www.chem.agilent.com/cag/bsp/sig/downloads/pdf/pca.pdf

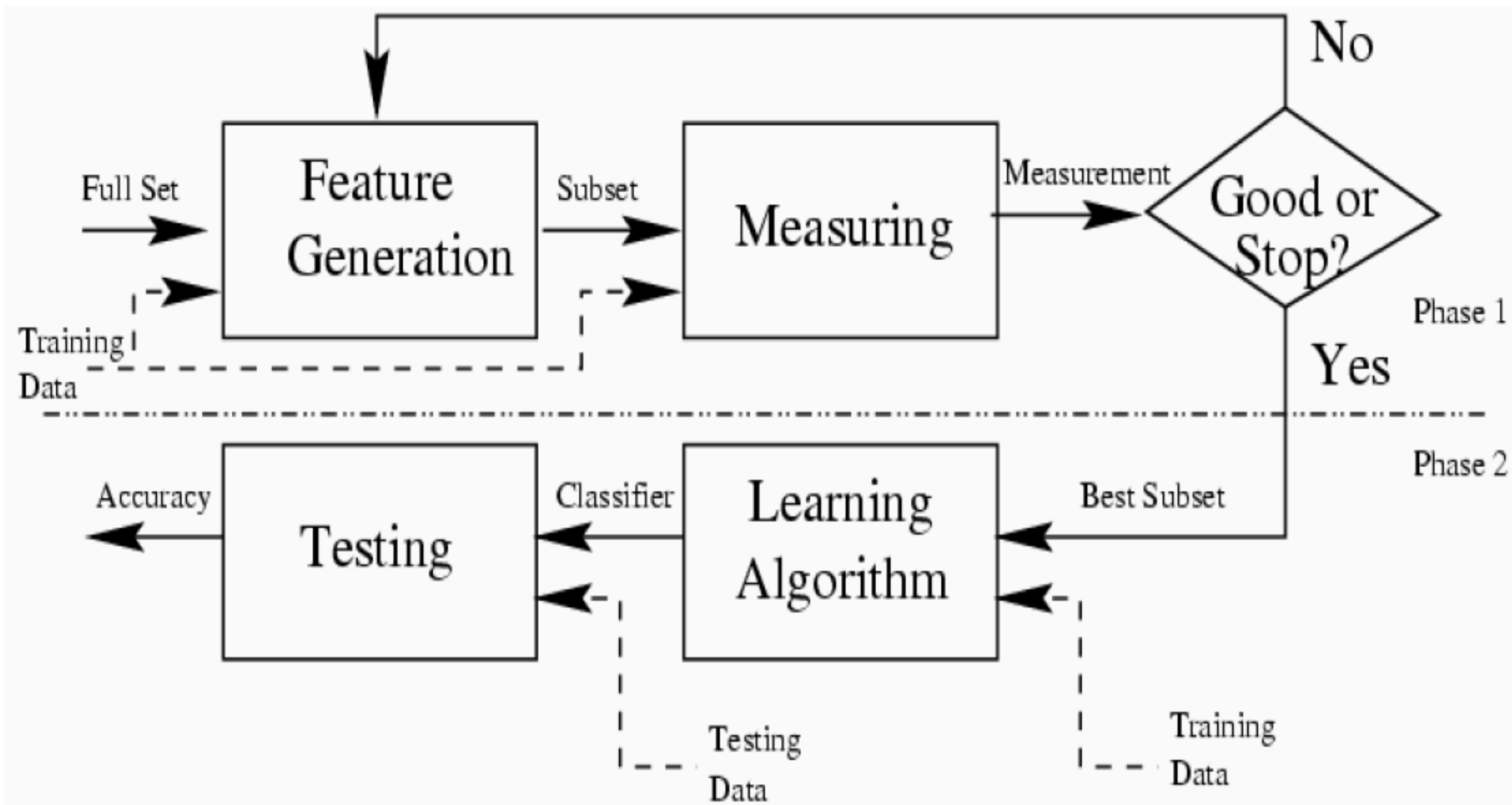
Feature Subset Selection

- Another way to reduce dimensionality of data
- Redundant features
  - duplicate much or all of the information contained in one or more other attributes
  - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
  - contain no information that is useful for the data mining task at hand
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

Feature Subset Selection

- Techniques:
  - Brute-force approach:
    - Try all possible feature subsets as input to data mining algorithm
  - Embedded approaches:
    - Feature selection occurs naturally as part of the data mining algorithm
  - Filter approaches:
    - Features are selected before data mining algorithm is run
  - Wrapper approaches:
    - Use the data mining algorithm as a black box to find best subset of attributes
  - Feature Weighting

Filter Approach



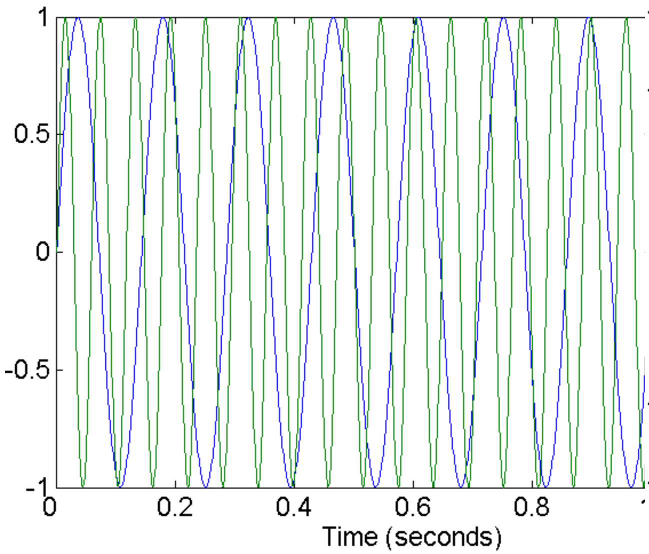
Feature Creation

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes

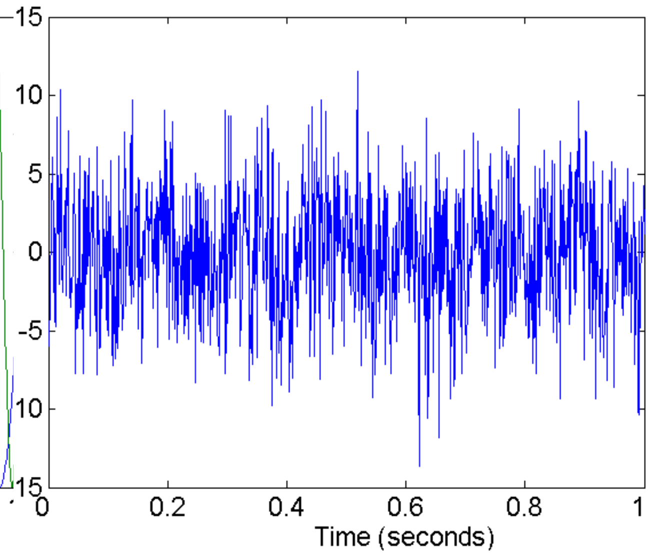
- Three general methodologies:
  - Feature Extraction
    - domain-specific
  - Mapping Data to New Space
  - Feature Construction
    - combining features

Mapping Data to a New Space

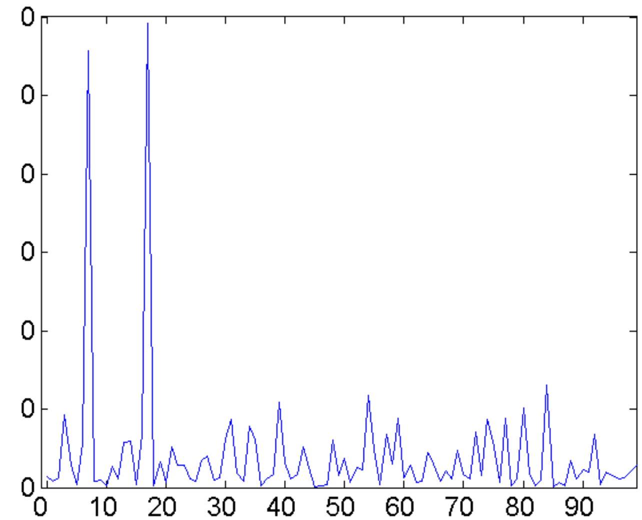
- Fourier transform
- Wavelet transform



Two Sine Waves



Two Sine Waves + Noise



Frequency

Dangers of Dimensionality Reduction

- https://cs.gmu.edu/~jessica/DimReducDanger.htm

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features. **Webster's Dictionary**

Similarity is hard to define, but…
"*We know it when we see it*"

The real meaning of similarity is a philosophical question.

We will take a more pragmatic approach.

Similarity and Dissimilarity

- Similarity
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Similarity/Dissimilarity for Simple Attributes

p and q are the attribute values for two data objects.

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 $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

Defining Distance Measures

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) is denoted by $D(O_1,O_2)$

What properties should a distance measure have?

- $D(A,B) = D(B,A)$ — *Symmetry*
- $D(A,A) = 0$ — *Constancy of Self-Similarity*
- $D(A,B) = 0$ Iff $A = B$ — *Positivity*
- $D(A,B) \leq D(A,C) + D(B,C)$ — *Triangular Inequality*

Measures for which all properties hold are referred to as distance *metrics*.

Intuitions behind desirable distance measure properties I

$D(\mathrm{A,B}) = D(\mathrm{B,A})$ *Symmetry*

Otherwise you could claim:

"Fairfax is close to D.C., but D.C is not close to Fairfax."

Intuitions behind desirable distance measure properties II

$D(\mathrm{A},\mathrm{A}) = 0$ *Constancy of Self-Similarity*

Otherwise you could claim:

"Fairfax is closer to D.C than D.C. itself!".

Intuitions behind desirable distance measure properties III

$D(A,B) = 0$ iff $A=B$ *Positivity*

Otherwise you could claim:

"Fairfax is exactly at the same location as DC"

Intuitions behind desirable distance measure properties IIII

$D(\mathrm{A},\mathrm{B}) \leq D(\mathrm{A},\mathrm{C}) + D(\mathrm{B},\mathrm{C})$ *Triangular Inequality*

Otherwise you could claim:

"My house is very close to Fairfax, your house is very close to Fairfax, but my house is very far from your house".

Euclidean Distance

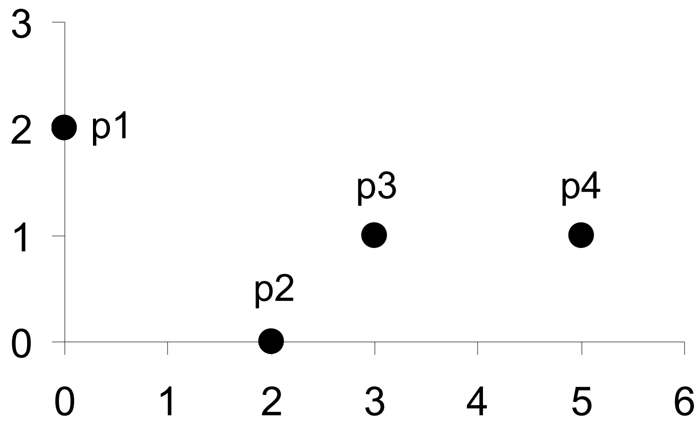
- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k^{th} attributes (components) or data objects p and q.

- Standardization is necessary, if scales differ.

Euclidean Distance



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

	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

Distance Matrix

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Where r is a parameter, n is the number of dimensions (attributes) and p_k and q_k are, respectively, the kth attributes (components) or data objects p and q.

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L1 norm) distance.
  - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $r = 2$. Euclidean distance
- $r \rightarrow \infty$. "supremum" (Lmax norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors
- Do not confuse r with n, i.e., all these distances are defined for all numbers of dimensions.

Minkowski Distance

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

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

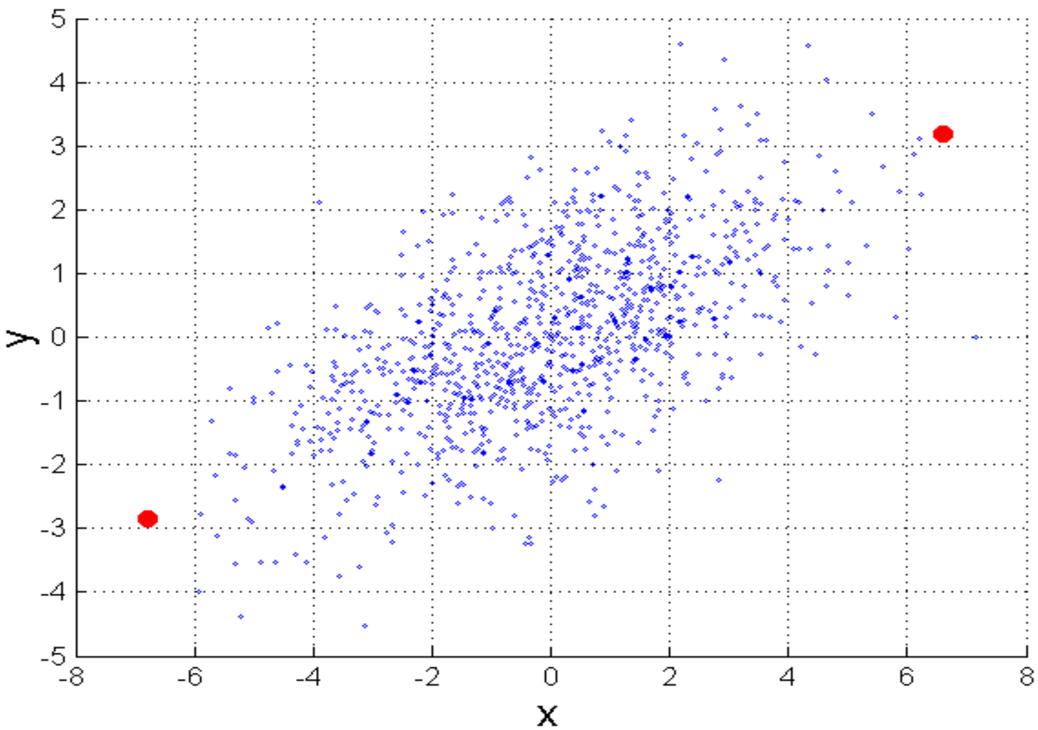
L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

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

Distance Matrix

Mahalanobis Distance

$$*mahalanobis(p,q) = (p-q)\sum^{-1}(p-q)^T$$



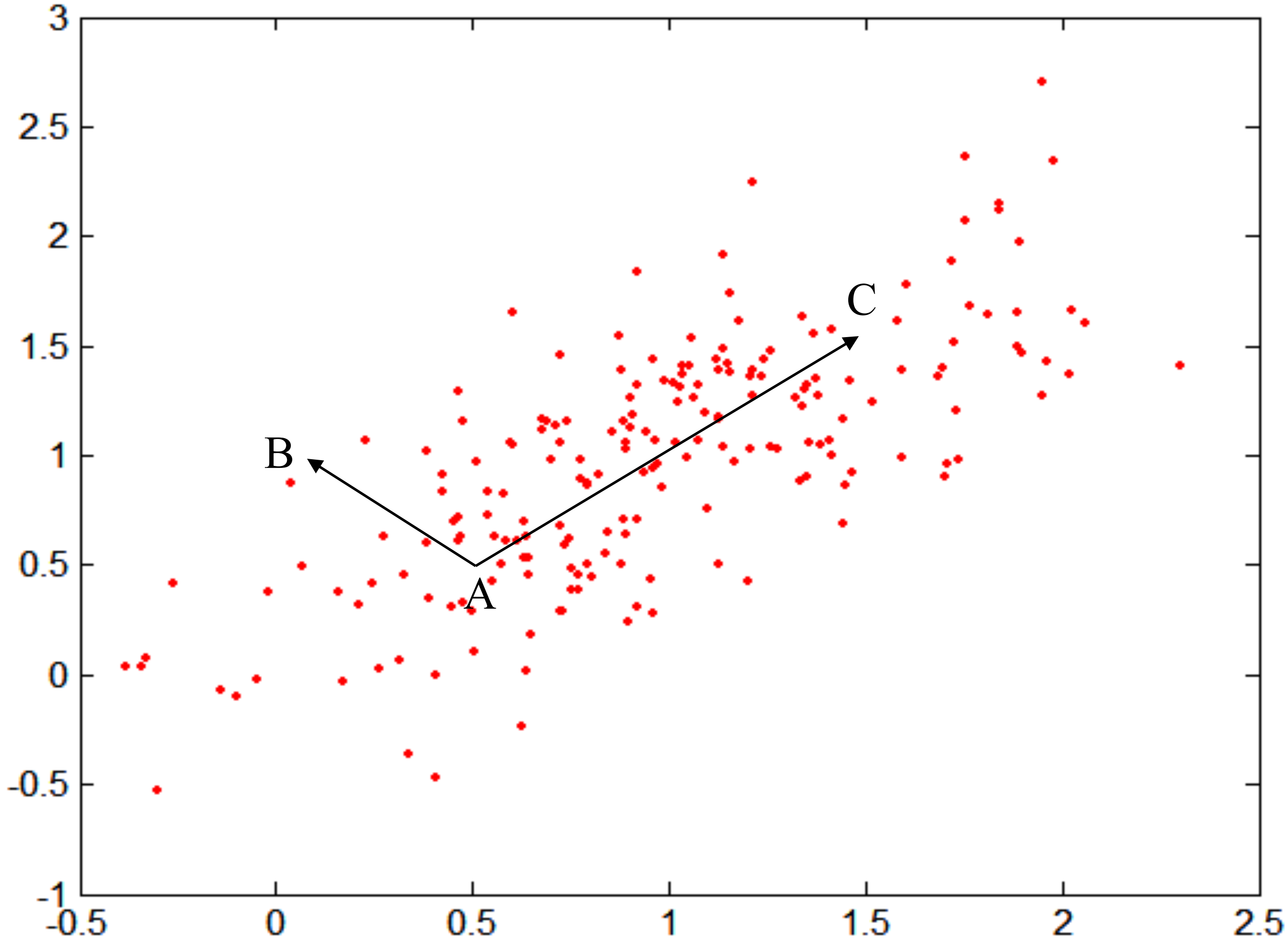
Σ is the covariance matrix of the input data X

$$\Sigma_{j,k} = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij}-\overline{X}_j)(X_{ik}-\overline{X}_k)$$

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

*** In some literature, this is the "squared" distance**

Mahalanobis Distance



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

Common Properties of Similarity

- Similarities also have some well known properties.

  – $s(p, q) = 1$ (or maximum similarity) only if $p = q$.

  – $s(p, q) = s(q, p)$ for all p and q. (Symmetry)

where $s(p, q)$ is the similarity between points (data objects), p and q.

Similarity Between Binary Vectors

- Common situation is that objects, *p* and *q*, have only binary attributes
- Compute similarities using the following quantities
  M_{01} = the number of attributes where p was 0 and q was 1
  M_{10} = the number of attributes where p was 1 and q was 0
  M_{00} = the number of attributes where p was 0 and q was 0
  M_{11} = the number of attributes where p was 1 and q was 1
- Simple Matching and Jaccard Coefficients
  SMC = number of matches / number of attributes
  $= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

  J = number of 11 matches / number of not-both-zero attributes values
  $= (M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

$p =$ 1 0 0 0 0 0 0 0 0 0
$q =$ 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)
$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)
$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)
$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

$SMC = (M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$

$J = (M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

  $\cos(d_1, d_2) = (d_1 \bullet d_2) / (||d_1||\ ||d_2||)$,

  where $\bullet$ indicates vector dot product and $||\ d\ ||$ is the length of vector d.

- Example:

  $d_1 =$ **3 2 0 5 0 0 0 2 0 0**
  $d_2 =$ **1 0 0 0 0 0 0 1 0 2**

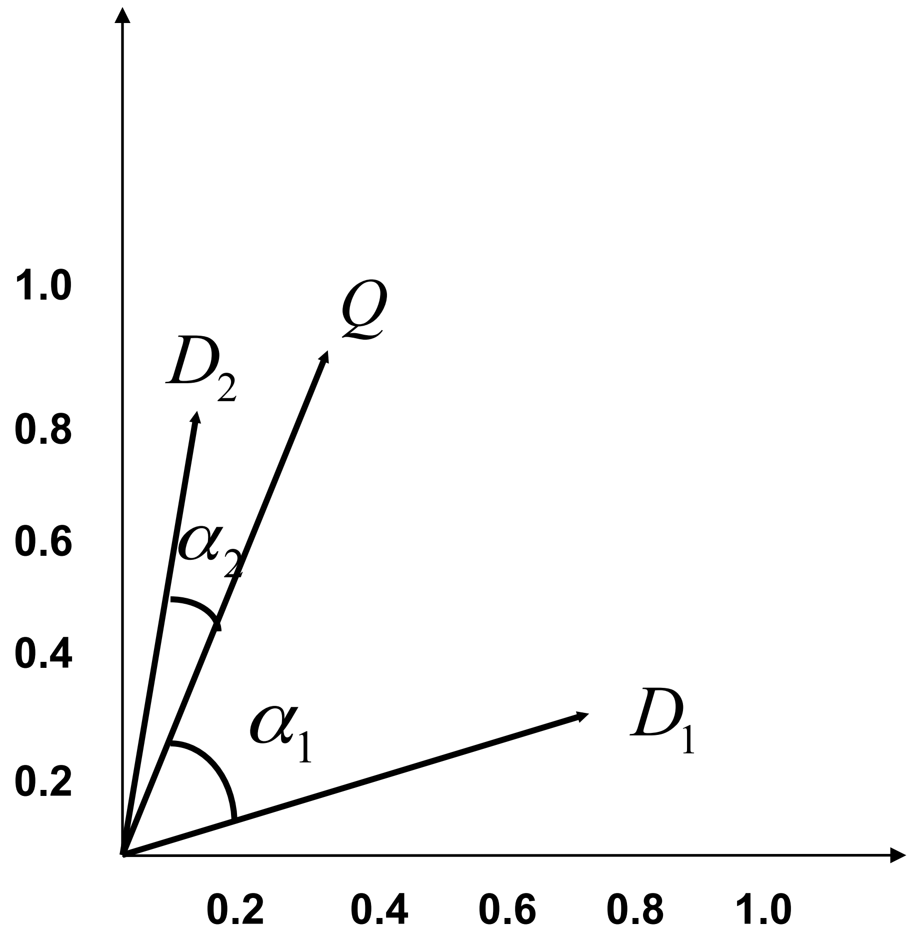
$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

$||d_1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{\mathbf{0.5}} = (42)^{\mathbf{0.5}} = 6.481$

$||d_2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{\mathbf{0.5}} = (6)^{\mathbf{0.5}} = 2.45$

$\cos(d_1, d_2) = .3150$

Cosine Similarity



$$D_1 = (0.8, 0.3)$$

$$D_2 = (0.2, 0.7)$$

$$Q = (0.4, 0.8)$$

$$\cos\alpha_1 = 0.74$$

$$\cos\alpha_2 = 0.98$$

Extended Jaccard Coefficient (Tanimoto)

- Variation of Jaccard for continuous or count attributes
  - Reduces to Jaccard for binary attributes

$$T(p, q) = \frac{p \bullet q}{\|p\|^2 + \|q\|^2 - p \bullet q}$$

Correlation

Correlation measures the linear relationship between objects

$$corr(x, y) = \frac{\text{Covariance}(x, y)}{\text{standard\_dev(x)*standard\_dev(y)}}$$

$$= \frac{S_{xy}}{S_x S_y}$$

Correlation (cont.)

$$\text{covariance(x,y)} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})(y_k - \bar{y})$$

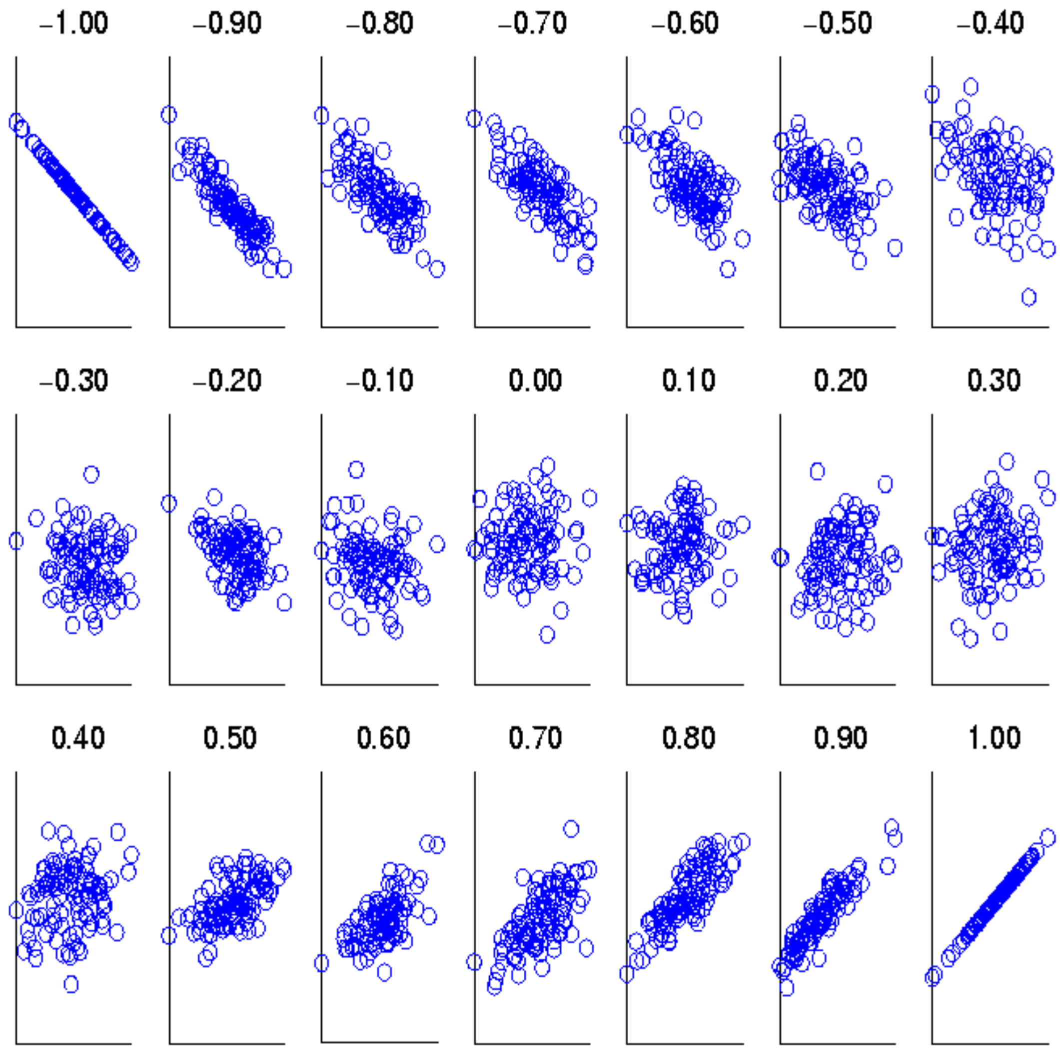
$$\text{standard\_dev(x)} = S_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})^2}$$

$$\text{standard\_dev(y)} = S_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k - \bar{y})^2}$$

Exercise

- x = (1 1 0 0 0), y = (0 0 0 1 1). Compute their correlation.

Visually Evaluating Correlation



General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^{th} attribute, compute a similarity, s_k, in the range $[0, 1]$.
2. Define an indicator variable, δ_k, for the k_{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of } 0\text{, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^{n} \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
  - Use weights wk which are between 0 and 1 and sum to 1.

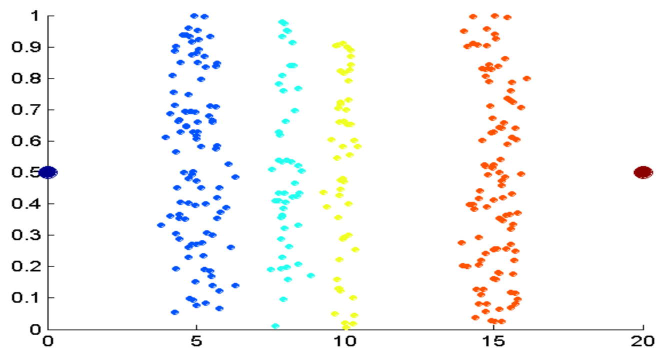
$$similarity(p,q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

$$distance(p,q) = \left(\sum_{k=1}^{n} w_k |p_k - q_k|^r \right)^{1/r}$$

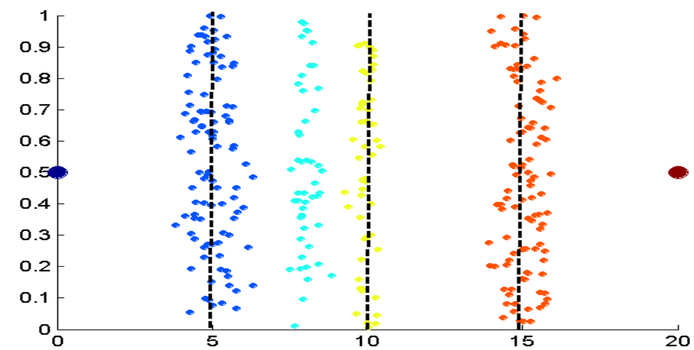
Which similarity function to use ?

- Depends on the application.
  - Analyze the attributes.
  - See their properties, min, max, etc
  - See their dependency on other attributes
  - Do you need similarity or distance ?
  - Do you need a metric ?
  - Try several functions.
  - Combine/merge.
- Active area of research!

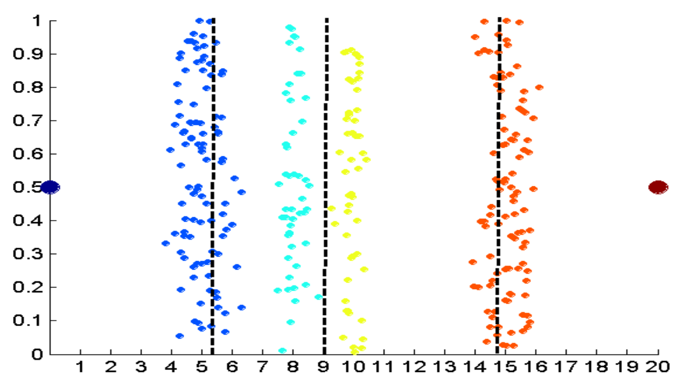
Discretization Without Using Class Labels



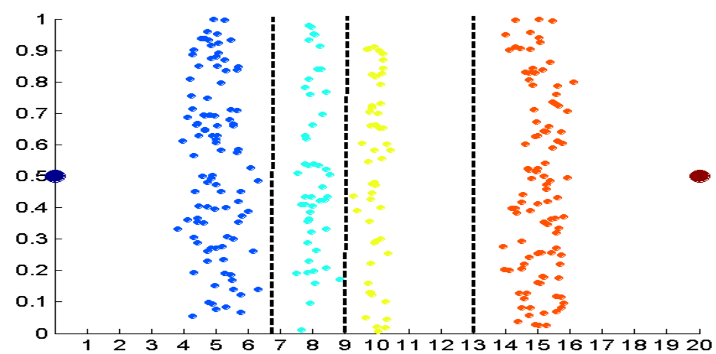
Data



Equal interval width



Equal frequency



K-means

Discretization Using Class Labels

- Entropy based approach:
  - If you have class labels, compute the entropy per discretized bin, and then try to minimize the same.
  - The entropy e_i for the i^{th} bin is given by (k = # of classes):

$$e_i = \sum_{j=1}^{k} p_{ij} \log_2 p_{ij}$$

where p_{ij} = prob(class j in the i^{th} interval)

  - If entropy = 0 then it is a pure grouping

Attribute Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
  - Simple functions: x^k, $\log(x)$, e^x, $|x|$
  - Standardization and Normalization