
Lecture Notes for Machine Learning in Python

Professor Eric Larson
Week Two, Lecture Four

Class Logistics and Agenda

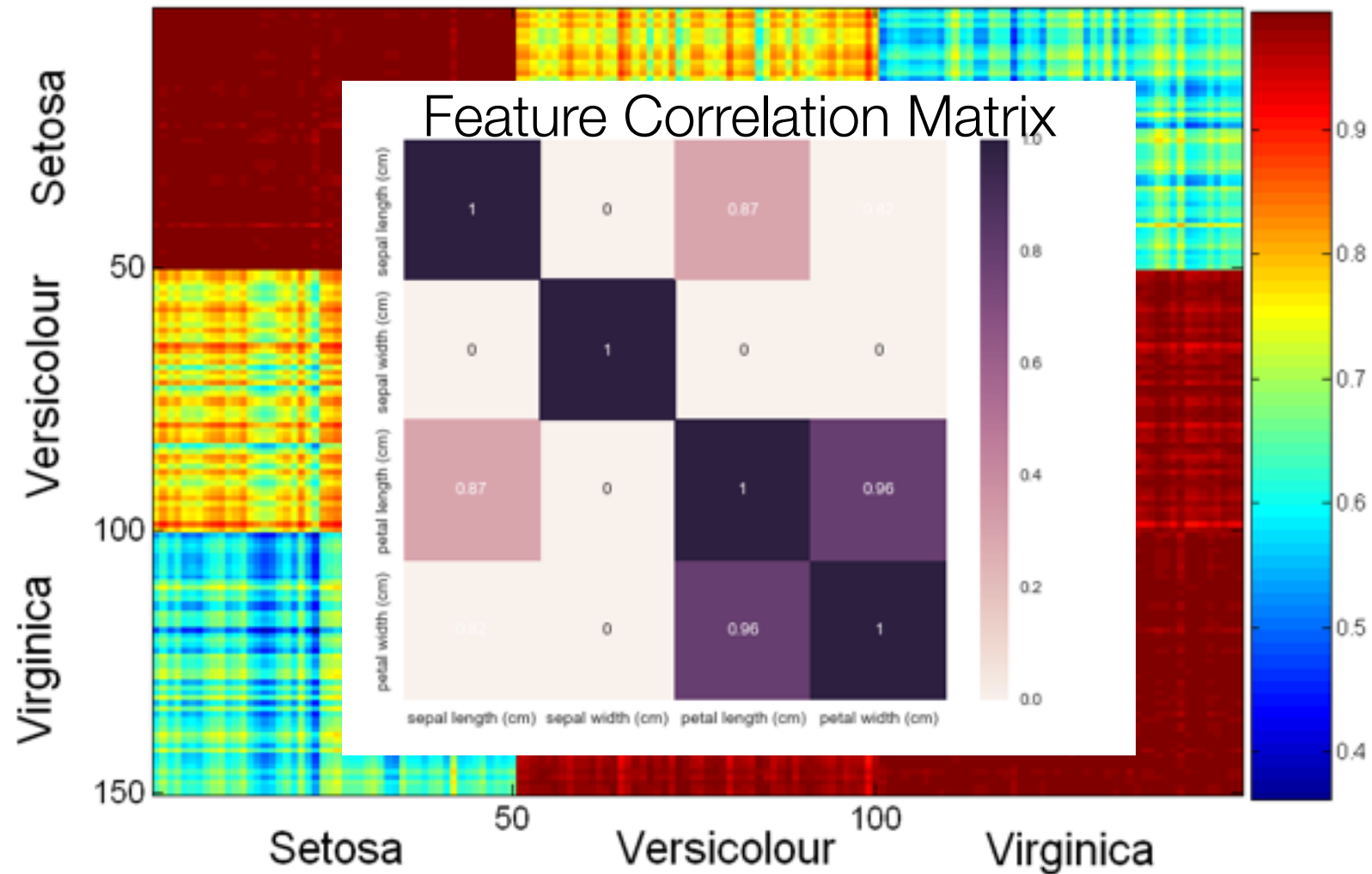
- Visualization
 - finish up
 - demo
- Dimensionality Reduction
 - PCA and LDA
 - Kernel Methods

Continued: Visualization

Visualization Techniques: Matrix Plots

- Matrix plots (typically heatmaps)
 - Plot some data matrix
 - This can be useful when objects are sorted well
 - Typically, the attributes are normalized to prevent one attribute from dominating the plot
 - Plots of similarity or distance matrices can also be useful for visualizing the relationships between objects

Instance Correlation Matrix

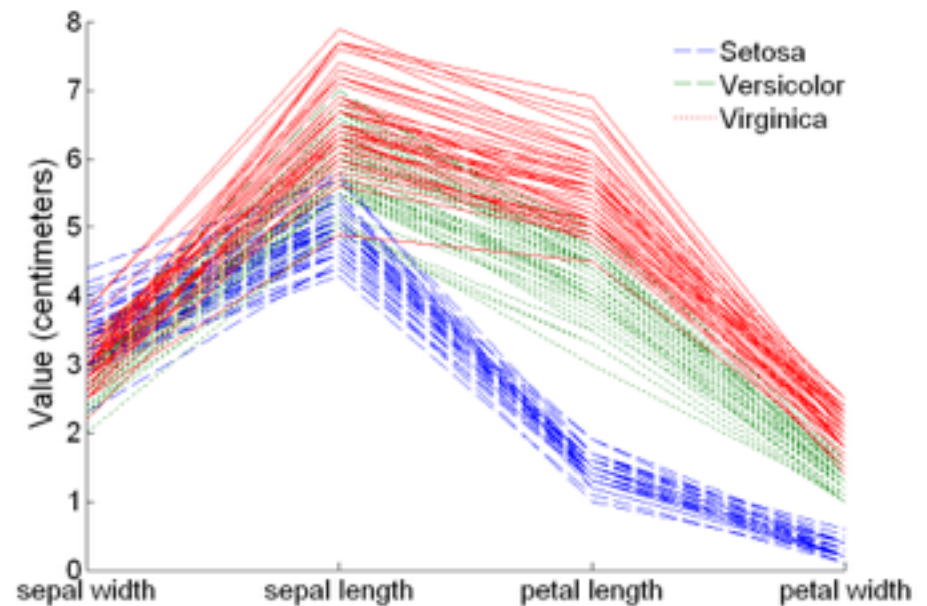
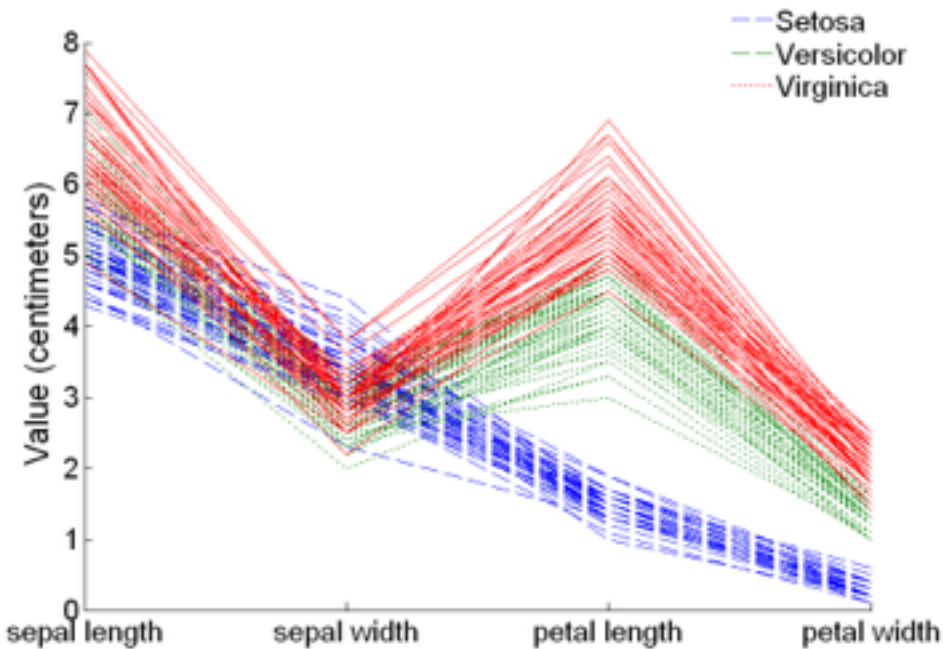


Visualization Techniques: Parallel Coordinates

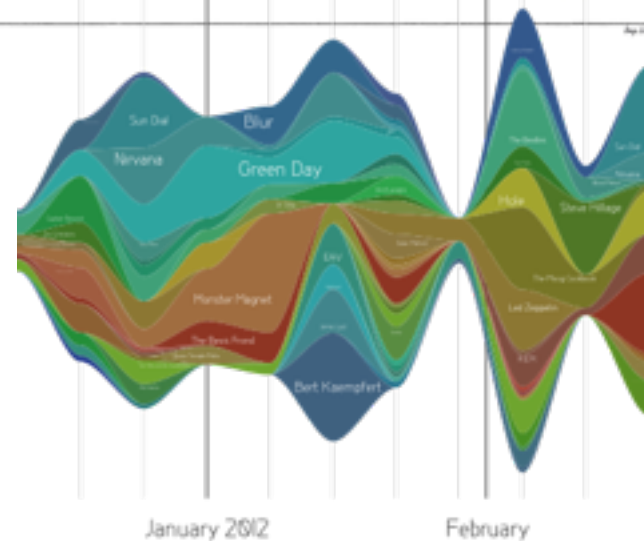
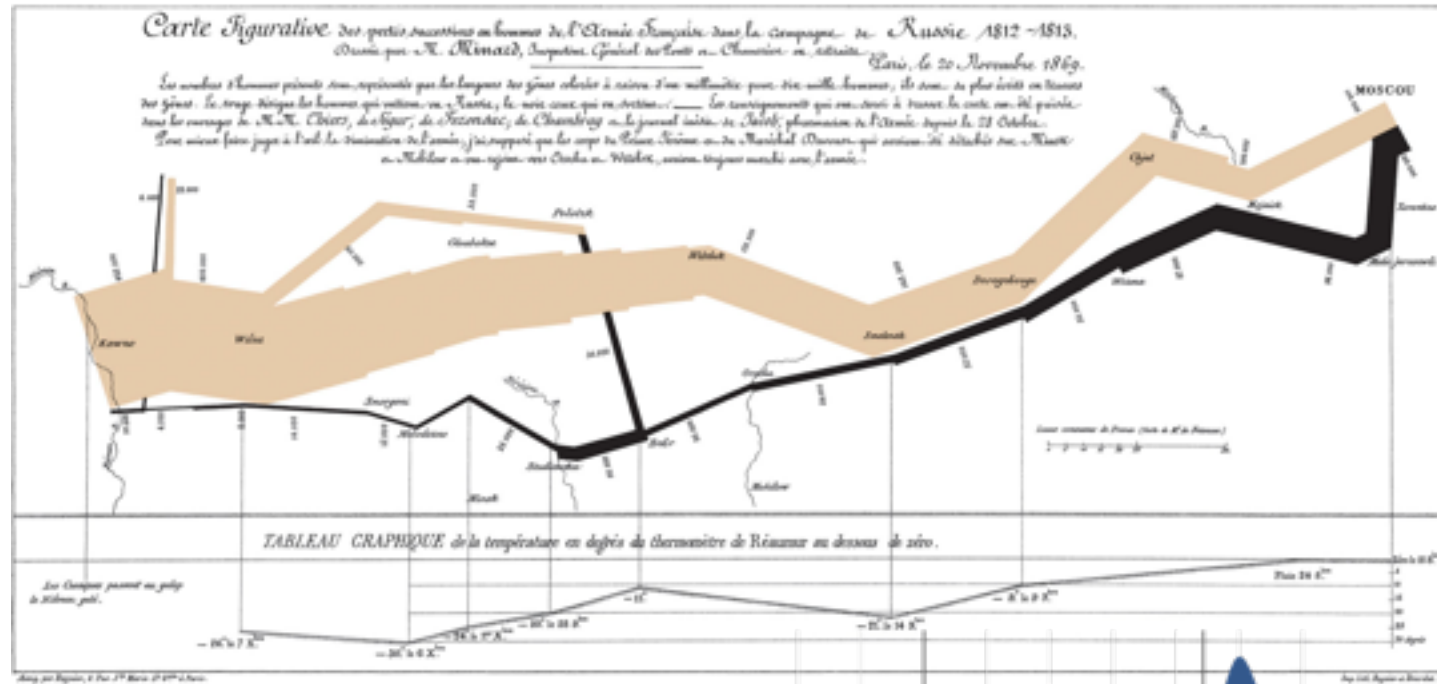
- Parallel Coordinates

- Used to plot the attribute values of multi-dimensional data
- Instead of using perpendicular axes, use a set of parallel axes
- The attribute values of each object are plotted as a point on each corresponding coordinate axis and the points are connected by a line
- Thus, each object is represented as a line
- Often, the lines representing a distinct class of objects group together, at least for some attributes
- Ordering of attributes is important in seeing such groupings

Parallel Coordinates Plots for Iris Data



There are lots of plots out there!



Matplotlib

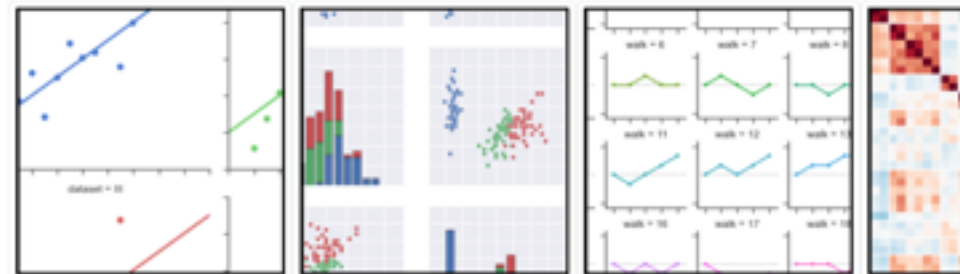
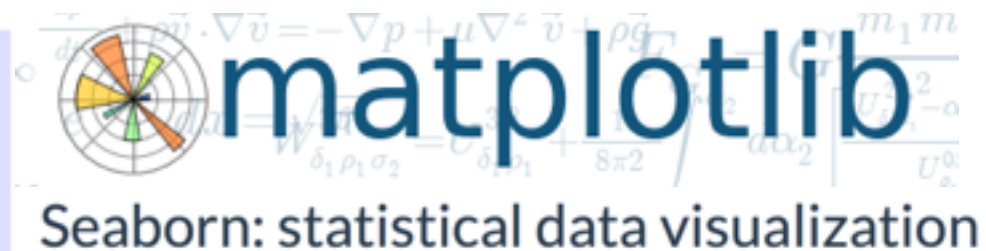
- Python plotting utility
 - Has low level plotting functionality
 - Highly similar to Matlab and R for plotting
- Extended for visually be more beautiful by
 - **seaborn**: stanford data visualization group

John Hunter (1968-2012)



On August 28 2012, John D. Hunter, the creator of matplotlib, died from complications arising from cancer treatment, after a brief but intense battle with this terrible illness. John is survived by his wife Miriam, his three daughters Rahel, Ava and Clara, his sisters Layne and Mary, and his mother Sarah.

If you have benefited from John's many contributions, please say thanks in the way that would matter most to him. Please consider making a donation to the [John Hunter Memorial Fund](#).



Visualization

Matplotlib

Seaborn

Plotly



Other Tutorials:

<http://stanford.edu/~mwaskom/software/seaborn/index.html>

<http://pandas.pydata.org/pandas-docs/stable/visualization.html>

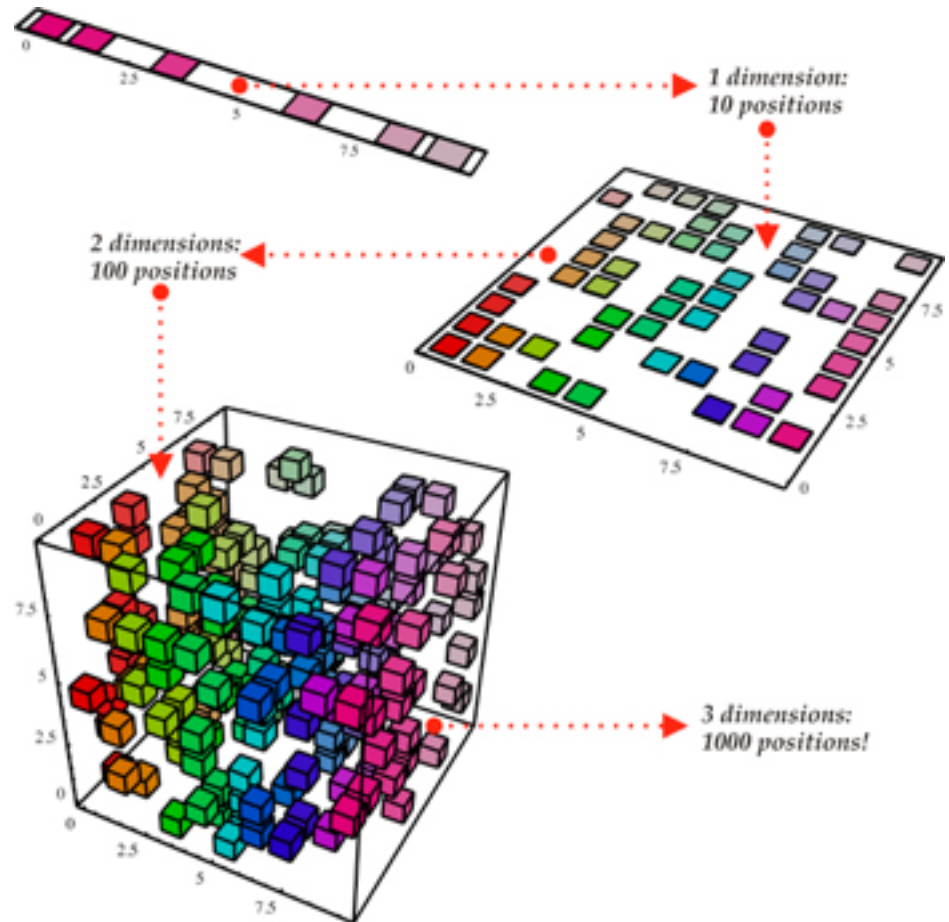
<http://matplotlib.org/examples/index.html>

http://nbviewer.ipynb.org/github/mwaskom/seaborn/blob/master/examples/plotting_distributions.ipynb

Dimensionality Reduction: PCA and LDA

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



Dimensionality Reduction

- Purpose:
 - Avoid curse of dimensionality
 - Select subsets of independent features
 - 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
 - Discriminant Analysis
 - Others: supervised and non-linear techniques

Karl Pearson

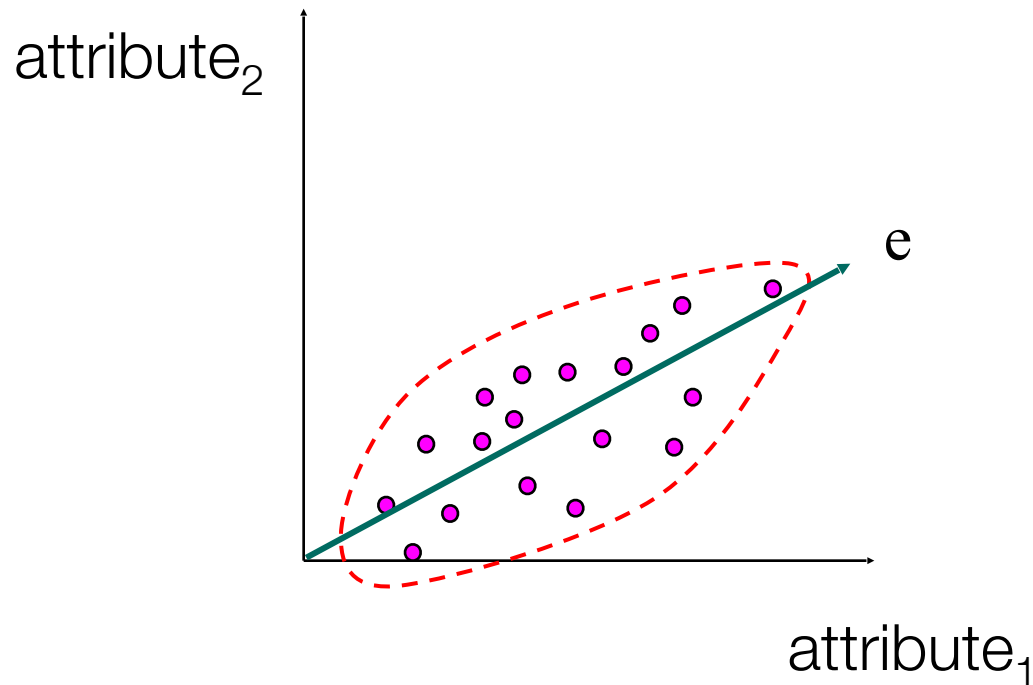


I invented PCA...
and *social Darwinism*

1857-1936

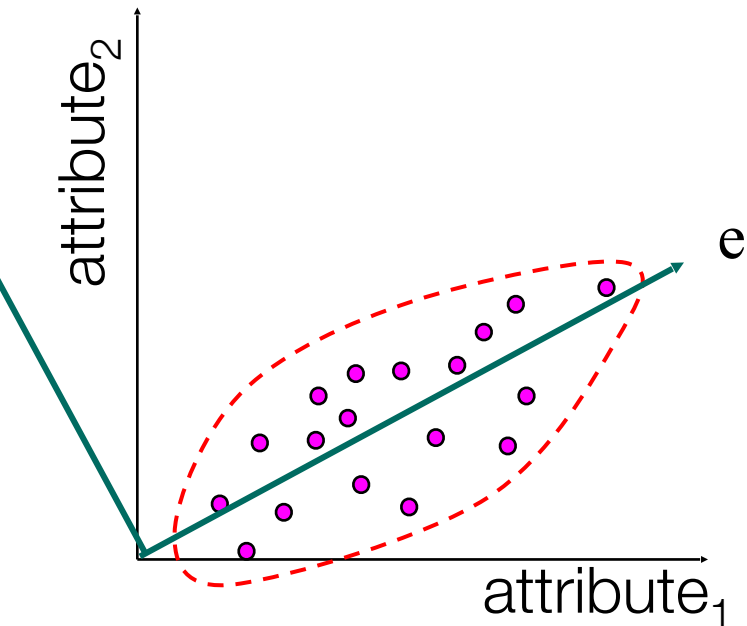
Dimensionality Reduction: PCA

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



Dimensionality Reduction: PCA

- Find the **eigenvectors** of the **covariance** matrix
- keep the “k” **largest** eigenvectors



$E1$	$E2$
0.85	0.85
0.52	-0.52

covariance

37.1	-6.7
-6.7	43.9

	$A1$	$A2$
1	66	33.6
2	54	26.6
3	69	23.3
4	73	28.1
5	61	43.1
6	62	25.6

	$A1$	$A2$
1	1.83	3.55
2	-10.1	-3.45
3	4.83	-6.75
4	8.83	-1.95
5	-3.17	13.05
6	-2.17	-4.45

zero mean

Dimensionality Reduction: PCA

attribute₂

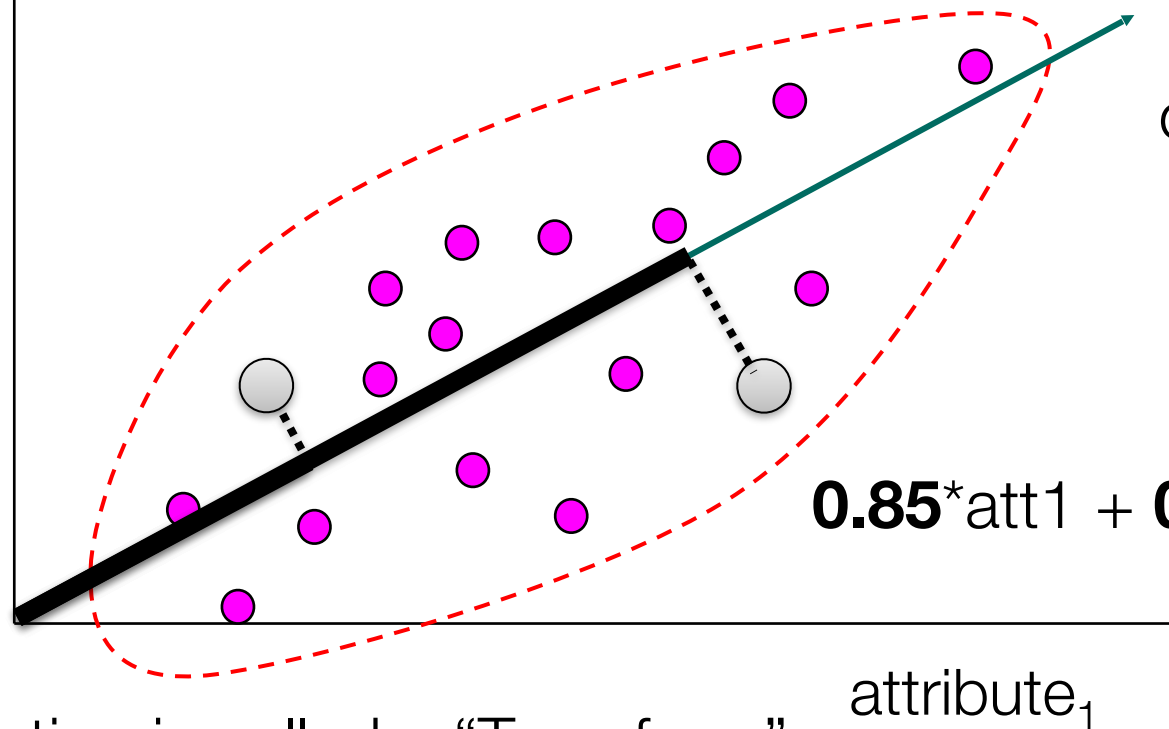
now I can represent data
as if it only had one attribute!!
vector weights are also meaningful...

E1

0.85

0.52

component



$$0.85^*att1 + 0.52^*att2$$

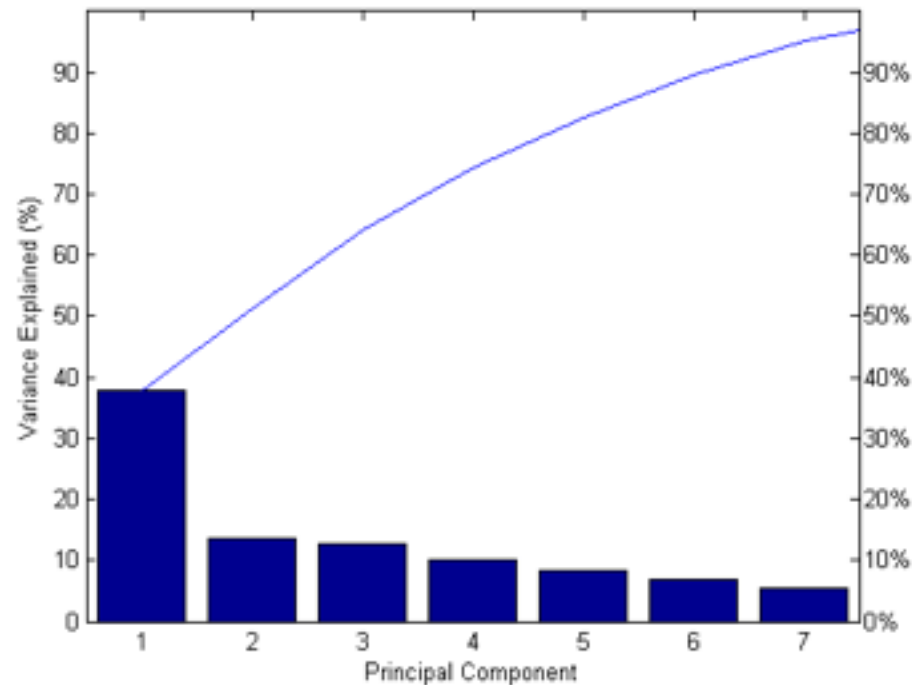
This projection is called a “Transform”

known as the **Karhunen-Loève Transform**

Explained Variance

- Each principle component explains a certain amount of variation in the data.
- This explained variation is embedded in the eigenvalues for each eigenvector

$$r_q = \frac{\sum_{j=1}^q \lambda_j}{\sum_{j=1}^p \lambda_j}$$



Dimensionality Reduction: PCA

- Genetic profiles distilled to 2 components

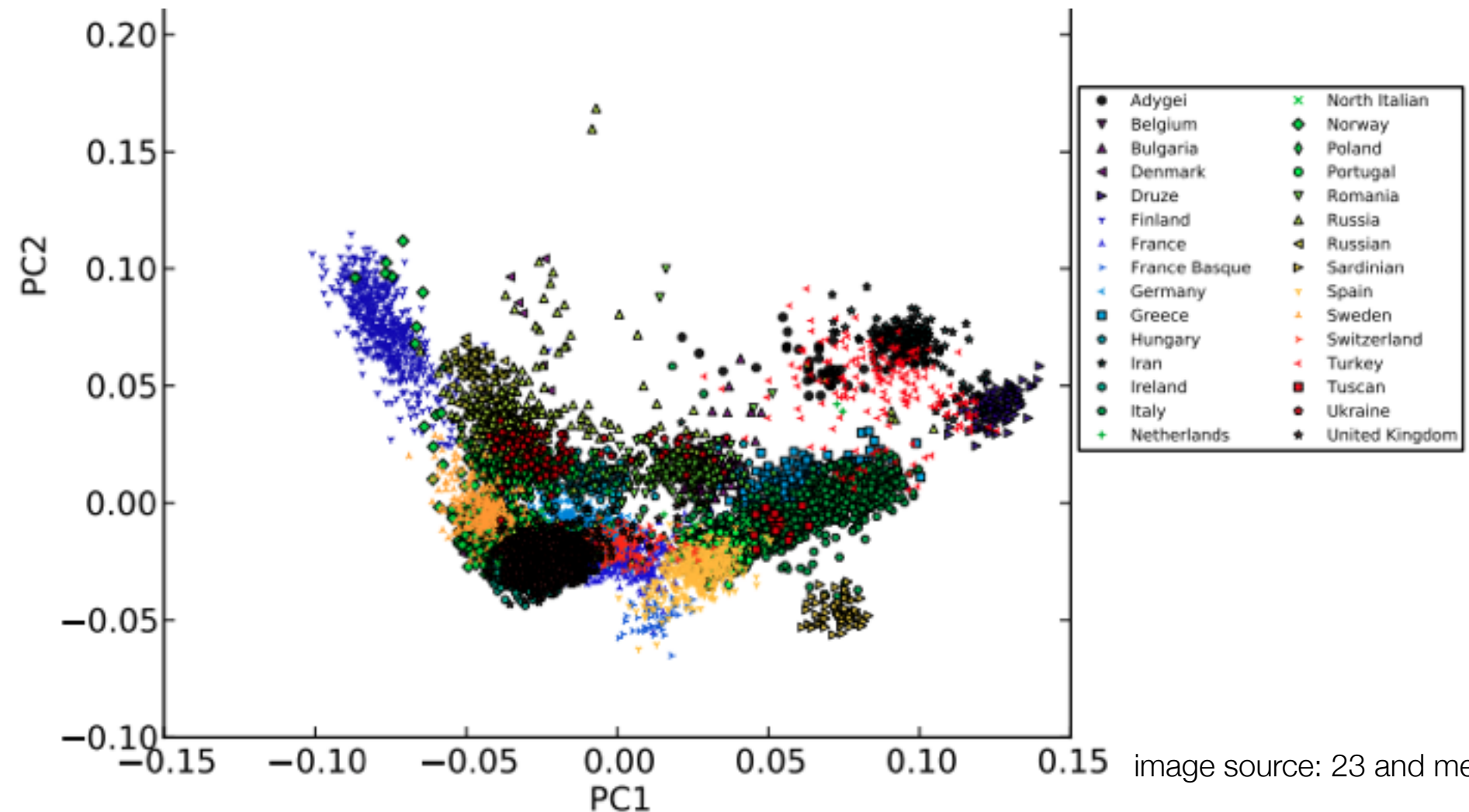


image source: 23 and me

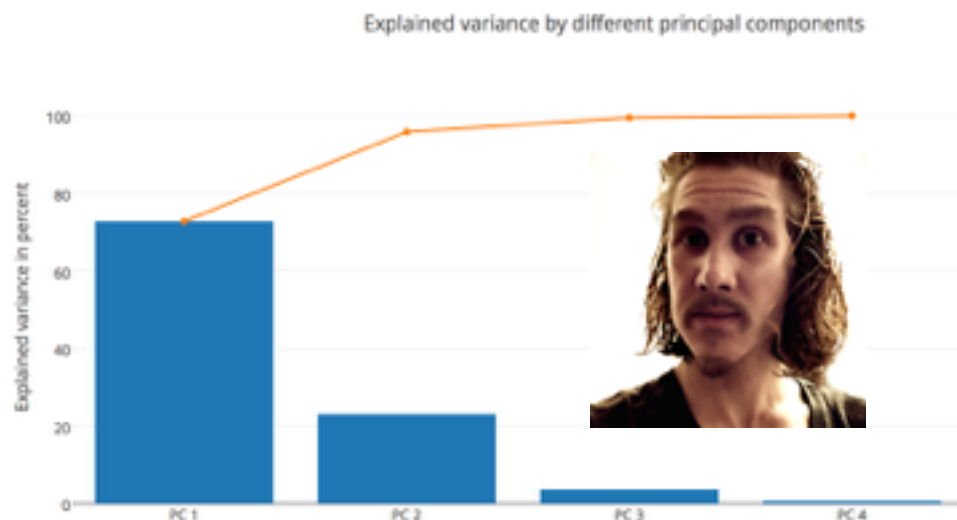
section 1: moved down

Dimensionality Reduction: PCA

- Need more help with the PCA algorithm (and python)?
 - check out Sebastian Raschka's notebooks:
http://nbviewer.ipython.org/github/rasbt/pattern_classification/blob/master/dimensionality_reduction/projection/principal_component_analysis.ipynb

Or check out PCA for dummies:

<https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>



Self Test ML2b.1

Principal Components Analysis works well for categorical data by design.

- A. True
- B. False
- C. It doesn't but people do it anyway

Dimensionality Reduction: Randomized PCA

- **Problem:** PCA on all that data can take a while to compute
 - What if the number of dimensions is gigantic?
 - Actually, **that's okay**: there are **iterative** algorithms for finding the **largest eigenvalues** that scales well with the number of data dimensions, but **not** the **number of instances...**
 - What if the number of instances is gigantic?
- What if we construct the covariance matrix with a subsample of the data?
 - By **randomly sampling from the dataset**, we can get something representative of the covariance for the entire dataset (if we sample correctly)

source: Halko, et al., (2009) Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions.

For Next Lecture

- Next Lecture: Labor Day!
- Next Next Lecture: Crash-course Image Feature Extraction

Supplemental Slides

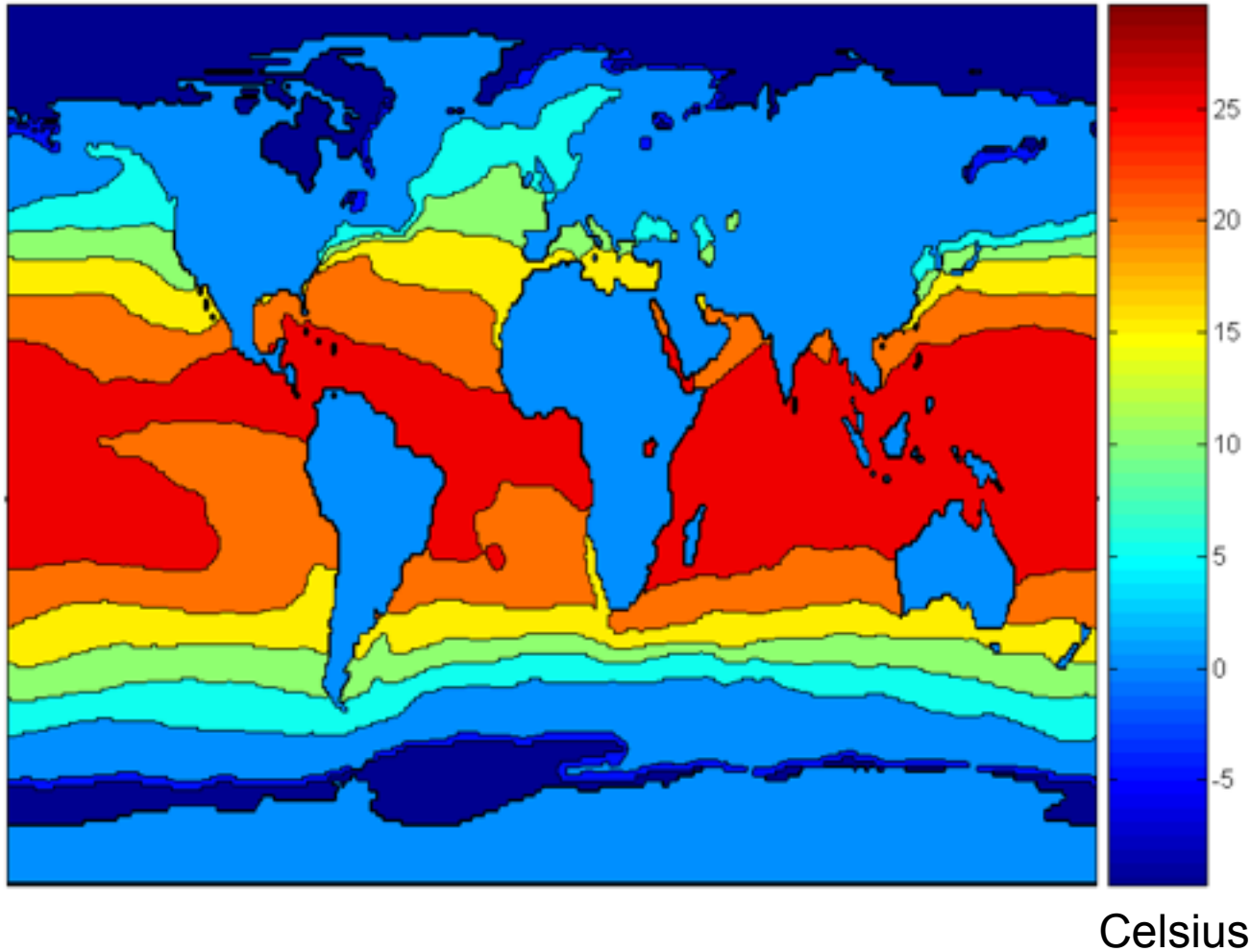
- Peruse these at your own leisure!
- These slides might assist you as additional visual aides
- **Slides courtesy of Tan, Steinbach, Kumar**
 - **Introduction to Data Mining**

Visualization Techniques: Contour Plots

- Contour plots

- Useful when a continuous attribute is measured on a spatial grid
- They partition the plane into regions of similar values
- The contour lines that form the boundaries of these regions connect points with equal values
- The most common example is contour maps of elevation
- Can also display temperature, rainfall, air pressure, etc.
 - ◆ An example for Sea Surface Temperature (SST) is provided on the next slide

Contour Plot Example: SST Dec, 1998



Other Visualization Techniques

- Star Plots

- Similar approach to parallel coordinates, but axes radiate from a central point
- The line connecting the values of an object is a polygon

- Chernoff Faces

- Approach created by Herman Chernoff
- This approach associates each attribute with a characteristic of a face
- The values of each attribute determine the appearance of the corresponding facial characteristic
- Each object becomes a separate face
- Relies on human's ability to distinguish faces

Challenges of Data Mining

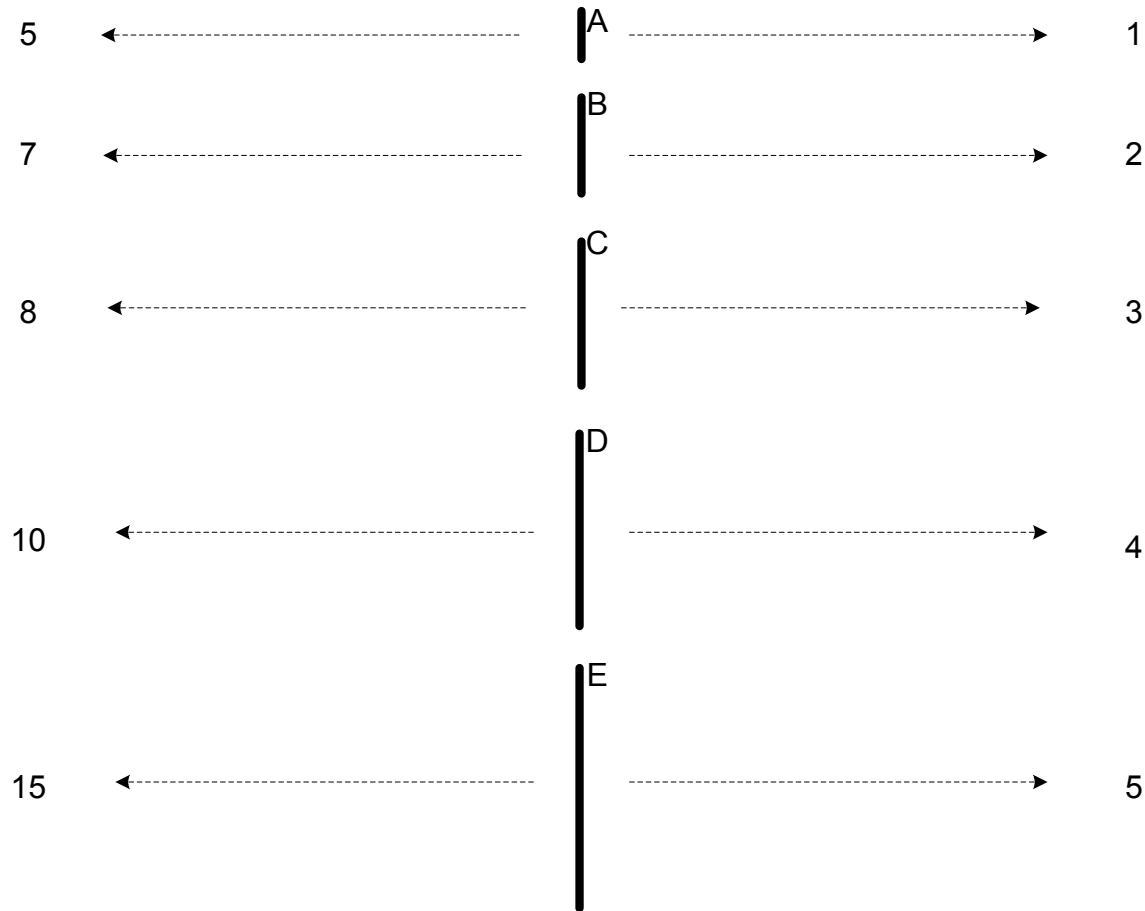
- Scalability
- Dimensionality
- Complex and Heterogeneous Data
- Data Quality
- Data Ownership and Distribution
- Privacy Preservation
- Streaming Data

Important Characteristics of Structured Data

- Dimensionality
 - ◆ Curse of Dimensionality
- Sparsity
 - ◆ Only presence counts
- Resolution
 - ◆ Patterns depend on the scale

Measurement of Length

- The way you measure an attribute is somewhat may not match the attributes properties.



Sampling

- **Sampling is the main technique employed for data selection.**
 - **It is often used for both the preliminary investigation of the data and the final data analysis.**
- **Statisticians sample because **obtaining** the entire set of data of interest is too expensive or time consuming.**
- **Sampling is used in data mining because **processing** the entire set of data of interest is too expensive or time consuming.**

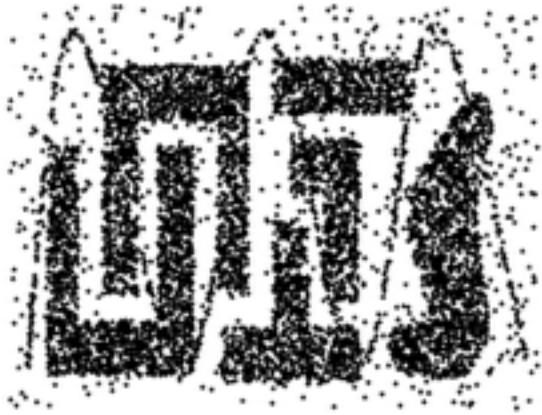
Sampling ...

- The key principle for effective sampling is the following:
 - using a sample will work almost as well as using the entire data sets, if the sample is representative
 - A sample is representative if it has approximately the same property (of interest) as the original set of data

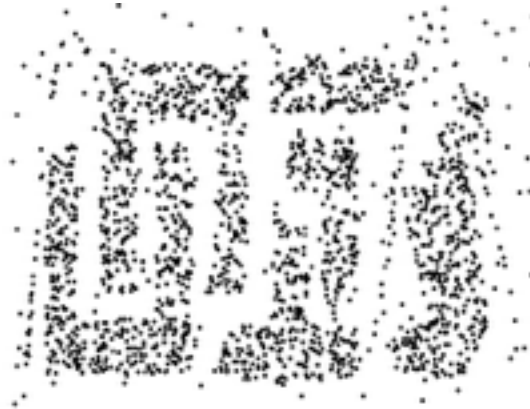
Types of Sampling

- Simple Random Sampling
 - There is an equal probability of selecting any particular item
- Sampling without replacement
 - As each item is selected, it is removed from the population
- Sampling with replacement
 - Objects are not removed from the population as they are selected for the sample.
 - ♦ In sampling with replacement, the same object can be picked up more than once
- Stratified sampling
 - Split the data into several partitions; then draw random samples from each partition

Sample Size



8000 points



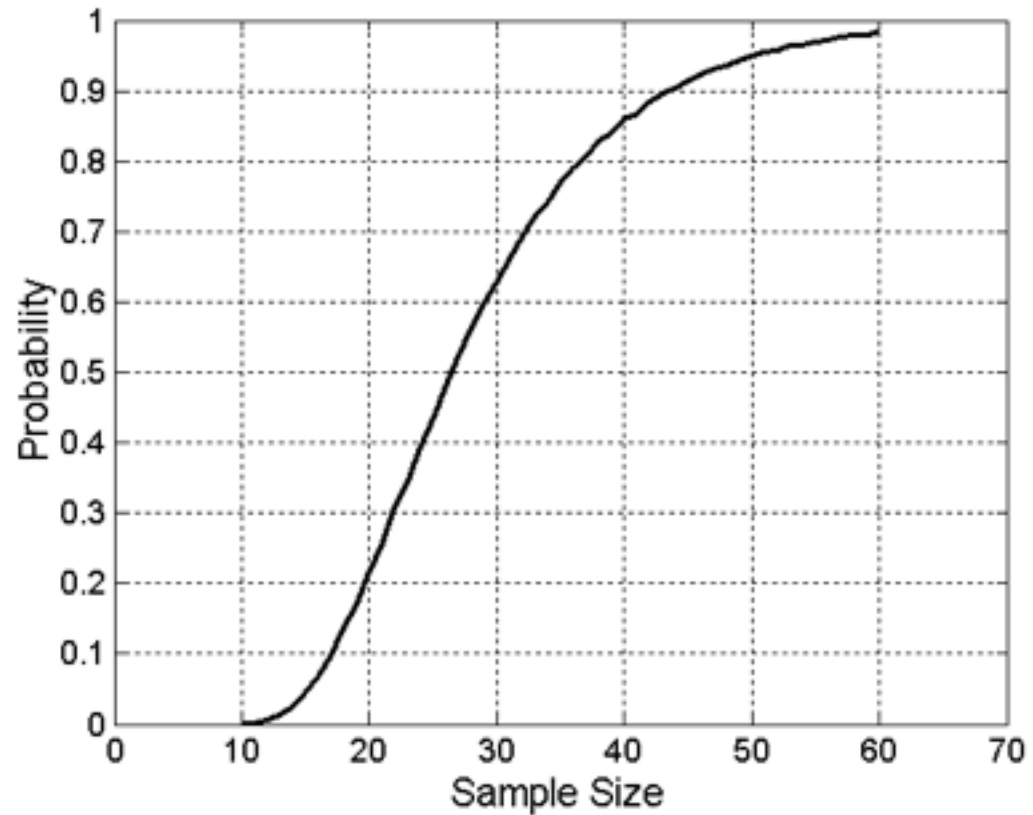
2000 Points



500 Points

Sample Size

- What sample size is necessary to get at least one object from each of 10 groups.



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 $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

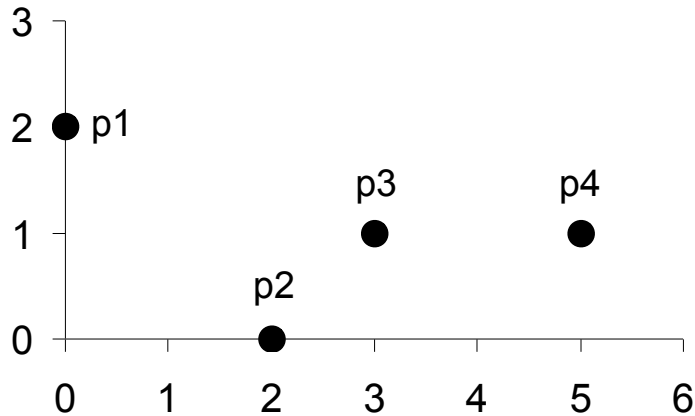
Table 5.1. Similarity and dissimilarity for simple attributes

Euclidean Distance

- Euclidean Distance

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

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

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

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 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” (L_{\max} 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

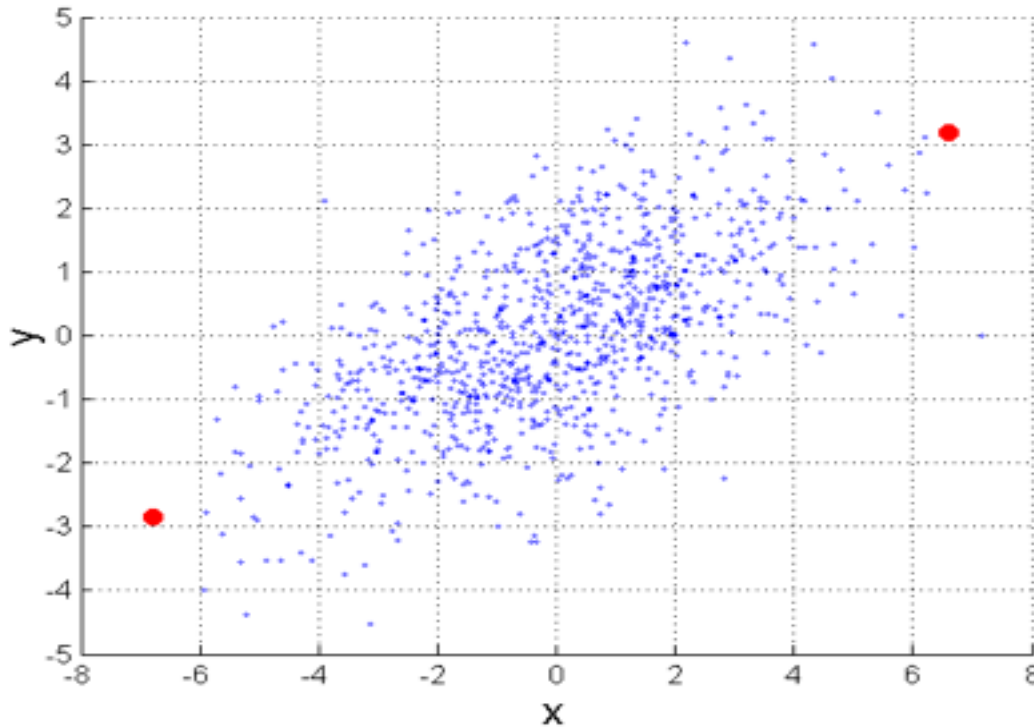
Distance Matrix

Mahalanobis Distance

$$\text{mahalanobis}(p, q) = (p - q) \Sigma^{-1} (p - q)^T$$

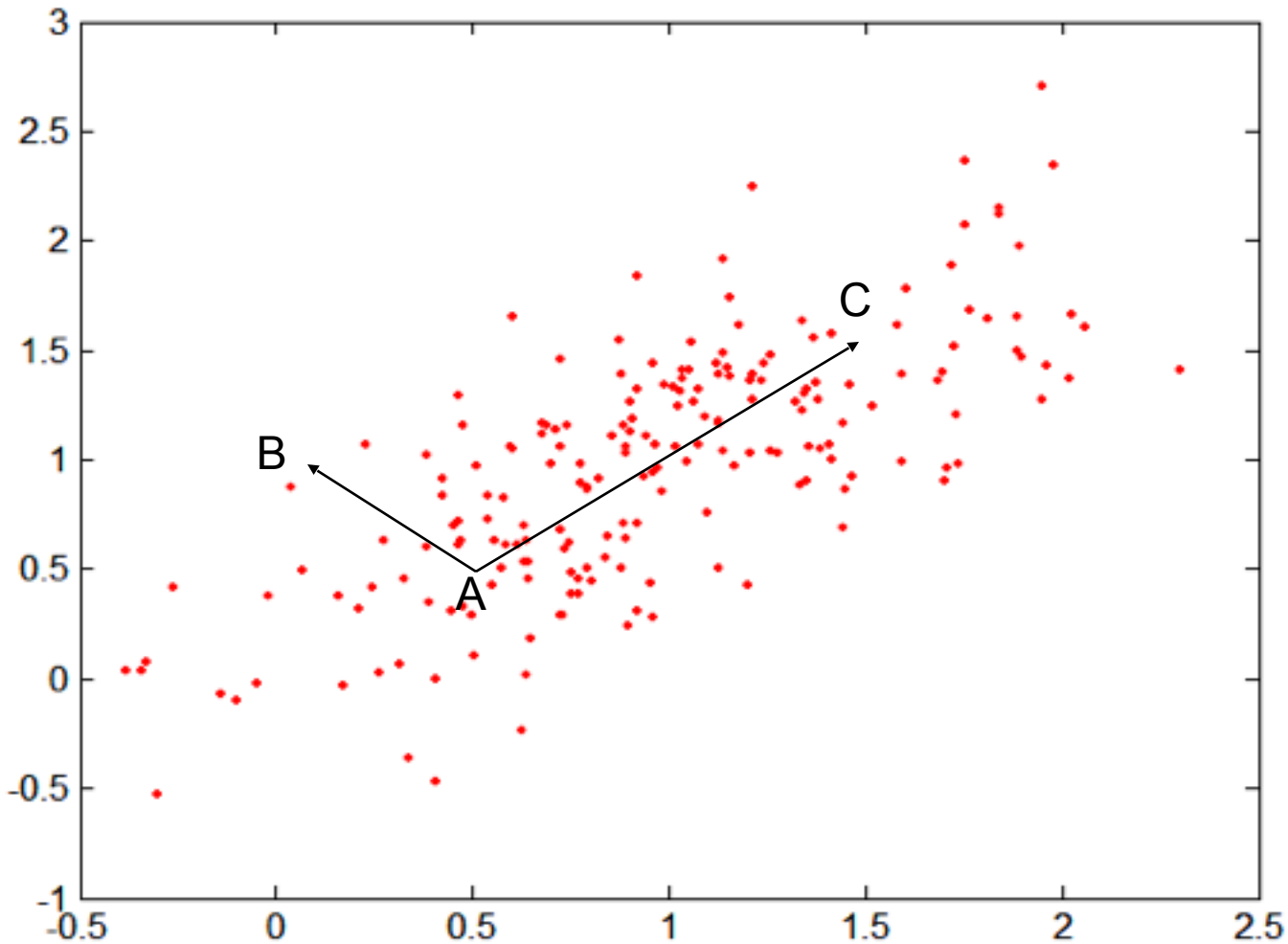
Σ is the covariance matrix of the input data X

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



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

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 a Distance

- Distances, such as the Euclidean distance, have some well known properties.

1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
- i. $d(p, q) = d(q, p)$ for all p and q . (Symmetry)
1. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p, q , and r . (Triangle Inequality)

where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q .

1. A distance that satisfies these properties is a **metric**

Common Properties of a Similarity

- Similarities, also have some well known properties.

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

$$p = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$$

$$q = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$$

$M_{01} = 2$ (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:

$$d_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$d_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

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

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

$$\|d_2\| = (1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2)^{0.5} = (6)^{0.5} = 2.245$$

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

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

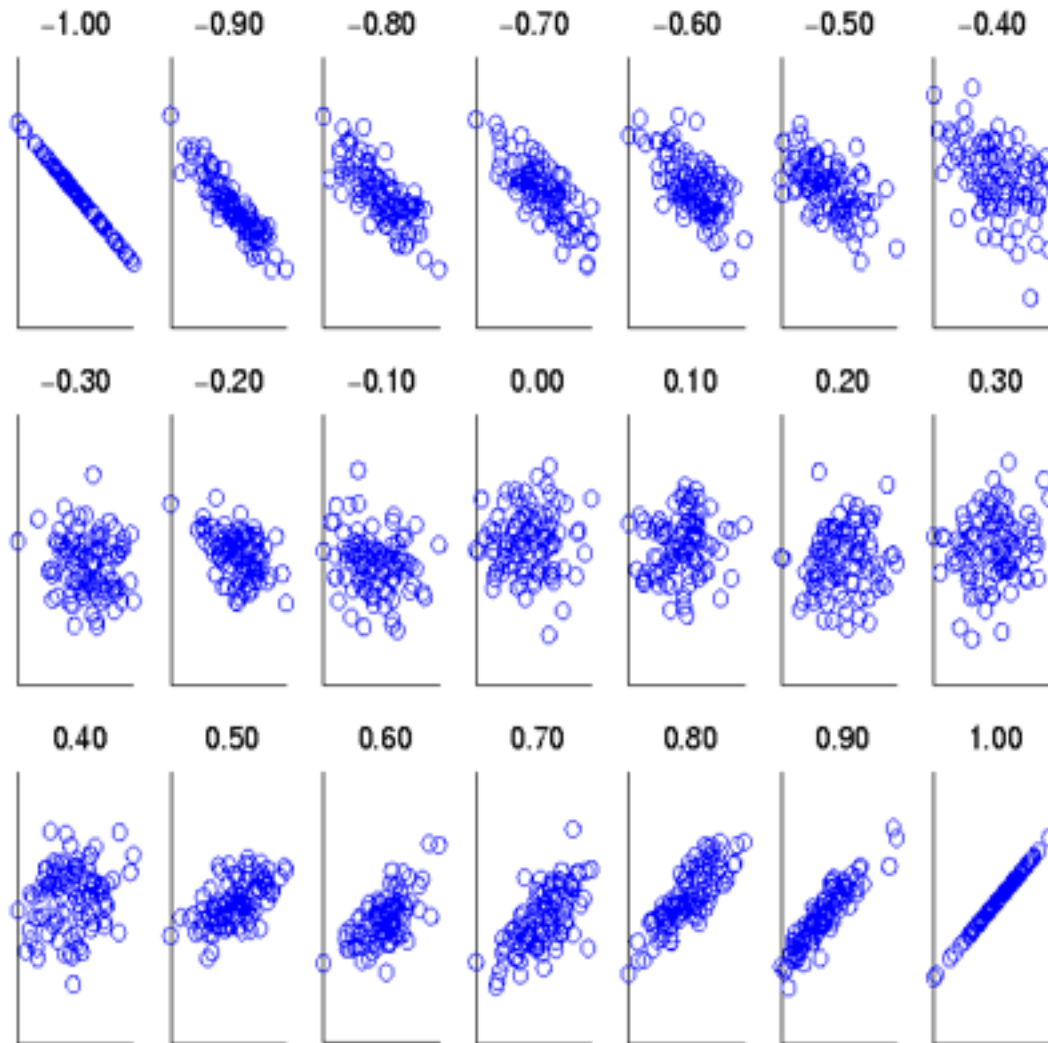
- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, p and q , and then take their dot product

$$p'_k = (p_k - \textit{mean}(p)) / \textit{std}(p)$$

$$q'_k = (q_k - \textit{mean}(q)) / \textit{std}(q)$$

$$\textit{correlation}(p, q) = p' \bullet q'$$

Visually Evaluating Correlation



Scatter plots showing the similarity from -1 to 1 .

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:

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

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 w_k which are between 0 and 1 and sum to 1.

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

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

Density

- Density-based clustering require a notion of density
- Examples:
 - Euclidean density
 - ◆ Euclidean density = number of points per unit volume
 - Probability density
 - Graph-based density

Euclidean Density – Cell-based

- Simplest approach is to divide region into a number of rectangular cells of equal volume and define density as # of points the cell contains

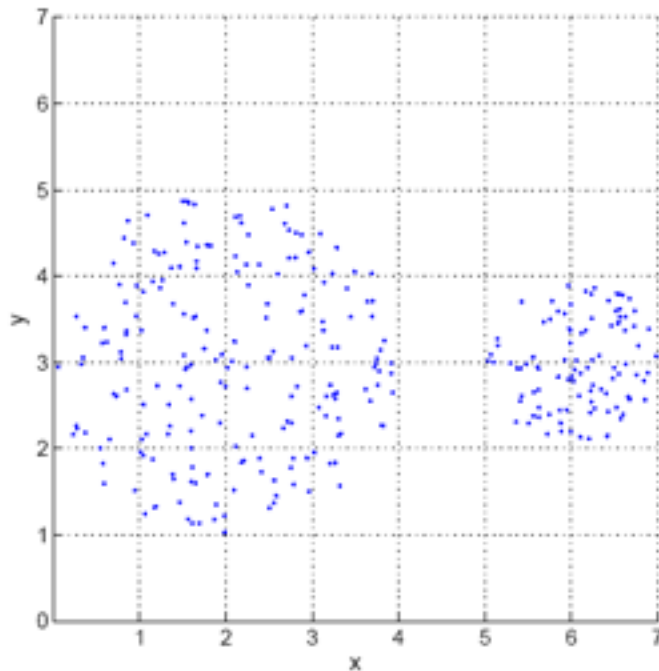


Figure 7.13. Cell-based density.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

Table 7.6. Point counts for each grid cell.

Euclidean Density – Center-based

- Euclidean density is the number of points within a specified radius of the point

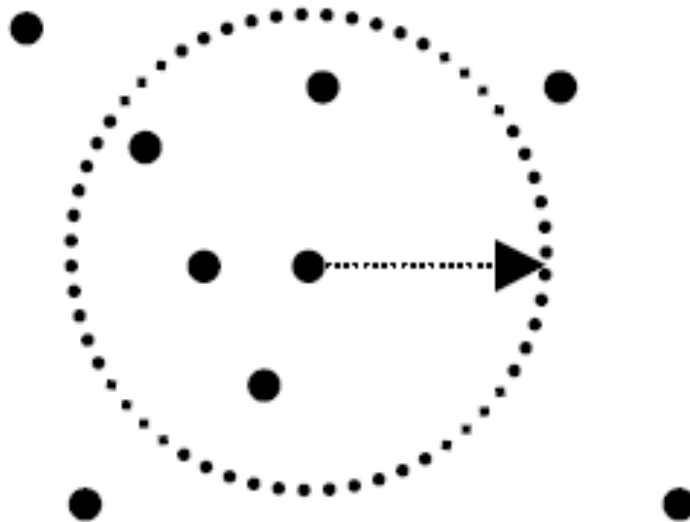


Figure 7.14. Illustration of center-based density.

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 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