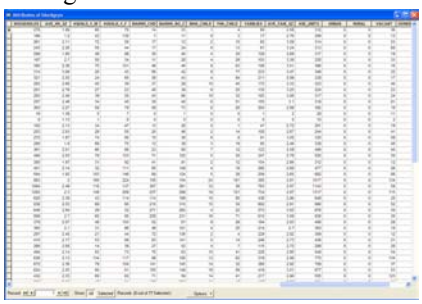
Multivariate Analysis

Outline

- Introduction
  - Multivariate data & Multidimensional space
  - Similarity (dissimilarity)
- Data Quantization
- Data Projection

A Multivariate Dataset

- Just big tables of numbers!



Data Matrix

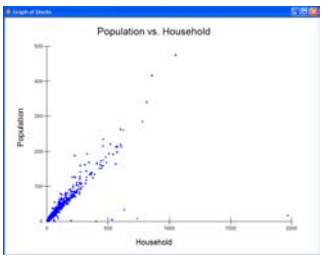
- A table can be essentially viewed as a matrix, with
  - Each **column** as an attribute describing an aspect of a particular case (e.g. geographic unit)
  - Each **row** as the list of all attributes

Visualizing Multivariate Data

- Obviously we can look at individual attribute separately
  - Histogram, box plot
- Or we can examine two attributes simultaneously
  - Scatter plot

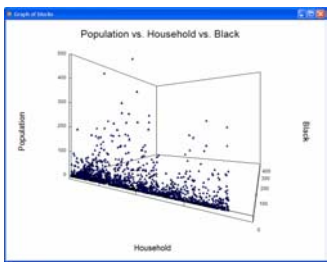
The Concept of Data Space

- Treat each attribute as a spatial coordinate → Data Space



The Concept of Data Space

- 3 dimensions are the most we can handle
- Shortcoming:
  - Hard to interpret in static view; often need interactive visualization tools to rotate around axis



What About More Attributes?

- We have multidimensional data space
  - Each **row** in the data matrix is considered as a **point** in this data space
- It is hard to visualize beyond 3 dimensions
  - **Data Projection**!!!!: ***projecting*** to 2-D or 3-D

What About Rows?

- # of rows can also be very large
  - **Data Quantization**!!!!: ***grouping*** based on similarity
    - What combinations of data are typical?
    - What combinations of data are unusual?
    - Can the cases be grouped into classes (clusters)?

Data Complexity Reduction

- **Data Projection**: ***projecting*** to 2-D or 3-D
- **Data Quantization**: ***grouping*** based on similarity

Similarity (Dissimilarity)

- A cluster is a group of cases
  - Similar to each other in the same cluster
  - but dissimilar to the cases in other clusters
- **Cluster analysis**!

Similarity (Dissimilarity)

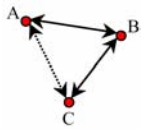
- Statistical distance:
  - In most cases, equivalent to spatial distance
- A generalized distance is *Minkowski distnace*
  - Euclidian distance: $m = 2$
  - Manhattan distance: $m = 1$

$$d(a,b) = [\sum_{i=1}^{p} (a_i - b_i)^m]^{1/m}$$

Similarity (Dissimilarity)

- Even more generalized distance: anything measures the **differences** between cases
- Whatever distance is used, similar cases must have shorter distance and dissimilar cases must have longer distance
- As long as it meets these 4 requirements:

$$d(a,a) = 0$$
$$d(a,b) > 0$$
$$d(a,b) = d(b,a)$$
$$d(a,c) \le d(a,b) + d(b,c)$$



Similarity (Dissimilarity)

- Sometimes, transformation is necessary
- Scale of unit: two attributes with different measurement
  - Solution I: *z*-score → Standardization
  - Solution II: scale to (0-1) → Normalization

$$d(a,b) = [\sum_{i=1}^{p} (a_i - b_i)^2]^{1/2}$$

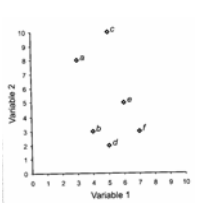
Similarity (Dissimilarity)

- What about nonmetric attributes?
  - **Hamming Distance**: the number of different values in two cases
  - Contingency table: *(a+d)/(a+b+c+d)* or *a/(a+b+c+d)* or *(a+d)/(b+c)*

	Location A	
Location B	1	0
1	a	b
0	c	d

Similarity (Dissimilarity)

- Distance matrix: one-one/a pair of data points
  - Symmetric



	a	b	c	d	e	f
a	0	5·1	2·8	6·3	6·4	4·2
b	5·1	0	7·1	1·4	3·0	2·8
c	2·8	7·1	0	8·0	7·3	5·1
d	6·3	1·4	8·0	0	2·2	3·2
e	6·4	3·0	7·3	2·2	0	2·2
f	4·2	2·8	5·1	3·2	2·2	0

Summary

- Geographic data are often with multiple attributes → form a multidimensional data space
- Geographic data are often large size with many records (cases)
- Necessary for complexity reduction
  - Data projection: dealing with dimensionality
  - Data quantization: dealing with clustering
- Rely on certain definition of similarity (dissimilarity)

Cluster Analysis

- A simple method: permutation of the distance matrix
  - Repeatedly swap rows and columns so that small values are as near to main diagonal as possible.
  - Result: similar cases will be close to each other
  - Not suitable for a larger data set

	a	b	c	d	e	f
a	0	5·1	2·8	6·3	6·4	4·2
b	5·1	0	7·1	1·4	3·0	2·8
c	2·8	7·1	0	8·0	7·3	5·1
d	6·3	1·4	8·0	0	2·2	3·2
e	6·4	3·0	7·3	2·2	0	2·2
f	4·2	2·8	5·1	3·2	2·2	0

	b	d	e	f	a	c
b	0	1·4	3·0	2·8	5·1	7·1
d	1·4	0	2·2	3·2	6·3	8·0
e	3·0	2·2	0	2·2	6·4	7·3
f	2·8	3·2	2·2	0	4·2	5·1
a	5·1	6·3	6·4	4·2	0	2·8
c	7·1	8·0	7·3	5·1	2·8	0

Cluster Analysis

- **Hierarchical** cluster analysis: iteratively decomposes or groups all the cases to form a nested hierarchy of clusters
  - We assume smaller-size clusters can form a larger-size cluster: *e.g.* a cluster A with 2 members is nested within a larger-size cluster B with 4 members (2 of them are members of cluster A)
- **Partitioning** cluster analysis: partition cases into *k* clusters
  - *k-means*, *k-medoid*

Hierarchical Cluster Analysis

- Two main categories:
  - ***Divisive*** (top-down) approach: **decomposes** the whole set (all the cases) into smaller subsets until each subset consists of only one case
  - ***Agglomerative*** (bottom-up) approach: **groups** all the cases into larger sets until all cases form just one group
- Both will result a treelike cluster diagram: dendrogram

Agglomerative Cluster Analysis

- Steps of a basic algorithm:
  1. Initially, each case forms its own cluster (only one member) → the original intercluster distance (d_{XY}) matrix (**D**)
  2. Find the smallest d_{XY} between any pair of clusters X and Y
  3. Merge X and Y into a new cluster, XY; construction a new **D**
     - Replace X and Y's columns and rows with a new column and a new row which contains the distances between XY to all other clusters ($d_{(XY)Z}$)
  4. Repeat Steps 2 and 3 (n-1) times, until all cases form a single large cluster

Agglomerative Cluster Analysis

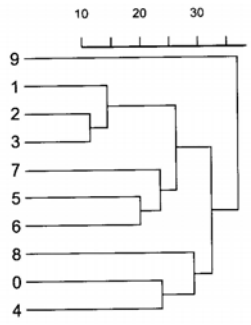
- How to calculate new intercluster distance ($d_{(XY)Z}$)?:
  - ***Single linkage***: **minimum** of the intercluster distances of Cluster X and Cluster Y to other clusters
    - $d_{(XY)Z}$ = **min**(d_{XZ}, d_{YZ})
  - ***Complete linkage***: **maximum** of the intercluster distances of Cluster X and Cluster Y to other clusters
    - $d_{(XY)Z}$ = **max**(d_{XZ}, d_{YZ})
  - ***Average linkage***: **average** of the intercluster distances of Cluster X and Cluster Y to other clusters
    - $d_{(XY)Z}$ = **Average**(d_{XZ}, d_{YZ})

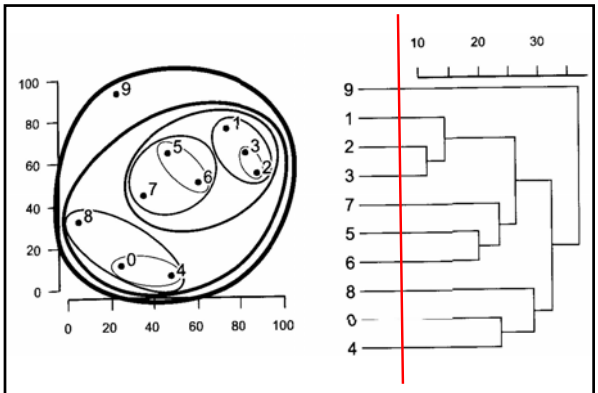
Agglomerative Cluster Analysis

- Ward's algorithm:
  - Joins clusters to minimize how spread out they become

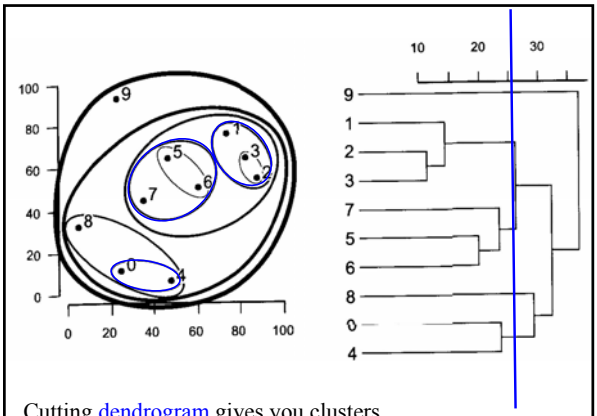
Hierarchical Cluster Analysis

- Result: dendrogram
  - Treelike diagram showing the sequence in which the set is aggregated (or divided)

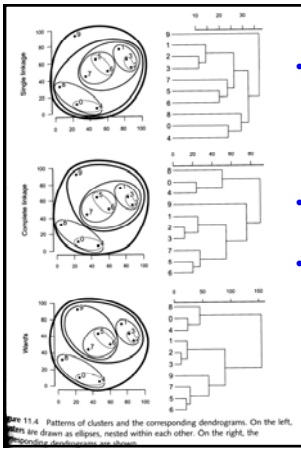




Cutting dendrogram gives you clusters



Cutting dendrogram gives you clusters



- ***Single linkage***: "stepped" appearance
  - When tree grows, a larger cluster is more likely to be closest to the next case
- ***Complete linkage***: more useful
- **Ward's algorithm**:
  - Produces "well-balanced" clusters

Exercise 21

- **Hierarchical cluster analysis**

Partitioning Cluster Analysis

- ***k-means***: value of *k* is predefined
  - We assume that there are *k* clusters in the data
  - Initially, **randomly** assign each case to one of the *k* clusters and then calculate the mean center of each cluster
  - For each case, determine which cluster's mean center is the **closest**, and then **reassign** the case to that cluster; **Recalculate** the mean center of clusters affected (that is, the cluster losing a member and the cluster gaining a member)
  - Repeat until no more reassignments are required

Issues with k-means

- *k* value?: how many clusters
  - Try different values to find the one that can minimize the within-cluster difference and maximize the between-cluster difference
- Initial membership assignment
  - Run several times to check the stability of the resulting clusters
- No structure in the clustering result
  - A case either belongs to a cluster or not

k-means vs. Hierarchical

- *k-means* is less computation intensive since it does not calculate the distance between any pair of cases
  - 1000 by 1000 = 1000000
- Also very hard to examine dendrogram by hierarchical methods if the dataset is large
- For both, the challenges:
  - Inclusion (exclusion) of certain attributes
  - Scale of the attributes: standardized to *z*-scores

Interpretation of Clusters

- Clusters in attribute space may form:
  - Regions: where clusters are also spatially continuous; → indicates spatial structures
  - Zones: where clusters are not spatially continuous; → indicates no significant spatial structures
- Mapping clustering results: with same color/pattern for the same cluster

Exercise 22

- **k-means**

Summary

- Two methods
  - Hierarchical
  - Partitioning

More Multivariate Methods

- Multiple Regression Analysis: confirmatory
- **Data Quantization: exploratory**
  - Hierarchical
  - Partitioning
- **Data Projection: exploratory**
  - Simple visual methods
  - Multidimensional Scaling (MDS)
  - Principle Component Analysis (PCA)
  - Factoring Analysis (FA)

Multiple Regression

- Discussed in Trend Surface Analysis
- Multiple independent variables
- Not covered in this class, HOWEVER:
  - It is widely used, perhaps the most popular statistical method of all
  - It has limitations: correlative effects; independent variables must be independent.

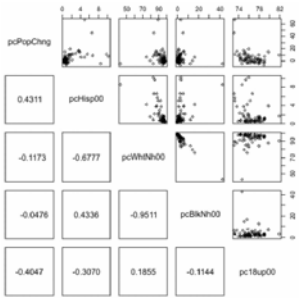
$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_px_p$$

Simple Visual Methods

- Scatter Plot Matrix
- Parallel Coordinate Plot (PCP)

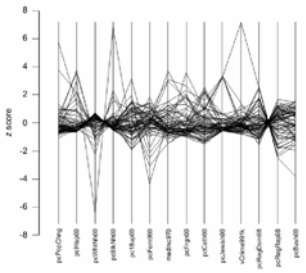
Scatter Plot Matrix

- Simply present all possible bivariate scatter plot in a matrix



PCP

- Each vertical axis represents an individual attribute
- All axis are parallel to each other
- Each case is displayed as a "*string*' connecting the value of that case at each attribute axis



User Interaction Methods

- User interaction techniques
  - *Assignment*: choose attributes to visualize
  - *Color Manipulation*: change color symbology;
  - *Focusing*: set the value range in display;
  - *Linking*: multiple visualization are linked;
  - *Brushing*: same case highlighted in one visualization will also highlighted in the other(s); often applied in conjunction with brushing

MDS

- Construct one-to-one mapping (projection) of each case from multidimensional data space to a lesser dimensional (usually, 2D) space → **reduced** dimensionality
  - Similar to map projection (3D → 2D)
- One criterion: **preserve** certain inter-case relationships among cases as much as possible
- However distortions (***stress***) are inevitable
  - Aim: to minimize a certain *stress* function

MDS

- Certain inter-case relationship: relative relationship → rankings of inter-case distances
- A ***stress*** function commonly used:
  - Try to minimize the difference between the rankings in the original multidimensional space and the rankings in the projected space (2-D or 3-D)

$$stree = \left[\frac{\sum_{i<j}\sum_{i}(r_{ij}^{(q)} - r_{ij}^{(q=p)})^2}{\sum_{i<j}\sum_{i}(r_{ij}^{(q)})^2}\right]^{1/2}$$

MDS

PCA

- Most widely used data projection method
- Based on the concepts of *eigenvectors* and *eigenvalues* of variance-covariance or correlation matrix (Appendix B)
- Identify "*principle components*" in the data
  - A set of variates that account for the most variation in the data
  - They are uncorrelated: remove the correlations among original attributes

PCA

- Construct a variance-covariance matrix based on the entire data first
  - To remove the scale among original attributes
  - Via standardization, variance-covariance matrix → correlation matrix

$$\begin{bmatrix} \sum(x_{1i}-\mu_1)^2 & (x_{1i}-\mu_1)(x_{2i}-\mu_2) \\ \sum(x_{1i}-\mu_1)(x_{2i}-\mu_2) & \sum(x_{2i}-\mu_2)^2 \end{bmatrix} \quad \begin{bmatrix} 47.738 & 48.221 \\ 48.221 & 128.015 \end{bmatrix}$$

$$\begin{bmatrix} \sigma_1^2 & \mathrm{cov}(x_1,x_2) & \mathrm{cov}(x_1,x_3) \\ \mathrm{cov}(x_1,x_2) & \sigma_2^2 & \mathrm{cov}(x_2,x_3) \\ \mathrm{cov}(x_1,x_3) & \mathrm{cov}(x_2,x_3) & \sigma_3^2 \end{bmatrix}$$

PCA

- Based this variance-covariance matrix, calculate *eigenvectors* and *eigenvalues*

eigenvalues *eigenvectors*

$$(\lambda_1, e_1) = (150.62, \begin{bmatrix} 0.4244 \\ 0.9055 \end{bmatrix})$$

$$(\lambda_2, e_2) = (25.137, \begin{bmatrix} -0.9055 \\ 0.4244 \end{bmatrix})$$

Principle Components

- Each component is in fact a **linear** combination of the original attributes
  - ***e***: the **component loading**; the value at the corresponding ***eigenvectors***;
    - Interpreted as the contribution of each original attribute to that component
  - PCs: Weighted summation of all attributes
    - The lager **e** value, the more closely related that original attribute
  - Same number as the original attributes

$$PC = e_1x_1 + e_2x_2 + \ldots + e_px_p$$

PCA

- Based *eigenvectors* and *eigenvalues*, calculate PCs

$$PC_1 = 0.4244x_1 + 0.9055x_2$$
$$PC_2 = -0.9055x_1 + 0.4244x_2$$

Illustration

- Suppose the data form a so-called hyper-ellipse
- PCA basically identify the major axes of the variation in this ellipsoidal cloud
- The variance of each PC is its corresponding ***eigenvalue***

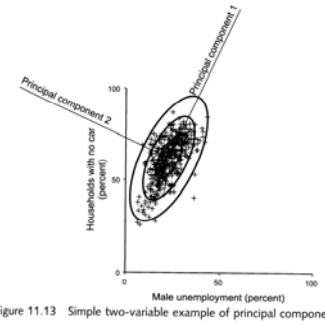
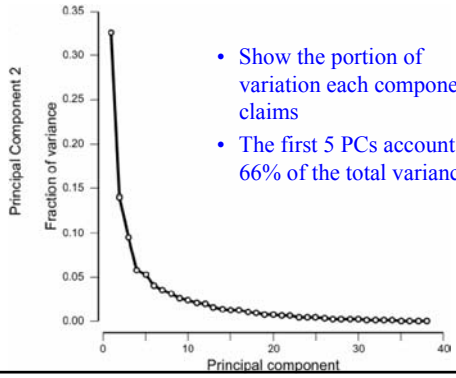


Figure 11.13 Simple two-variable example of principal components.

Properties of PCs

- The components are **orthogonal** to each other → uncorrelated or independent to each other
- The components are usually **ordered** based on how much (percentage) it accounts for the overall variance in the data
- **Data projection**: we can choose a smaller set of PCs that account for the majority of the data variation
- **Drawback**: sometimes it is hard to interpret individual PCs

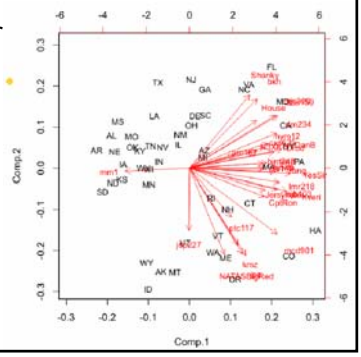
Scree Plot



- Show the portion of variation each component claims
- The first 5 PCs account for 66% of the total variance

Biplot

- Standard scatter plot + component loading vectors



Exercise 23

- **PCA**

Summary

- Multivariate methods are increasingly important in Geography
- Various methods
  - Data Quantization: clustering analysis
  - Data Projection: PCA, MDS
- Still poorly coupled in GIS