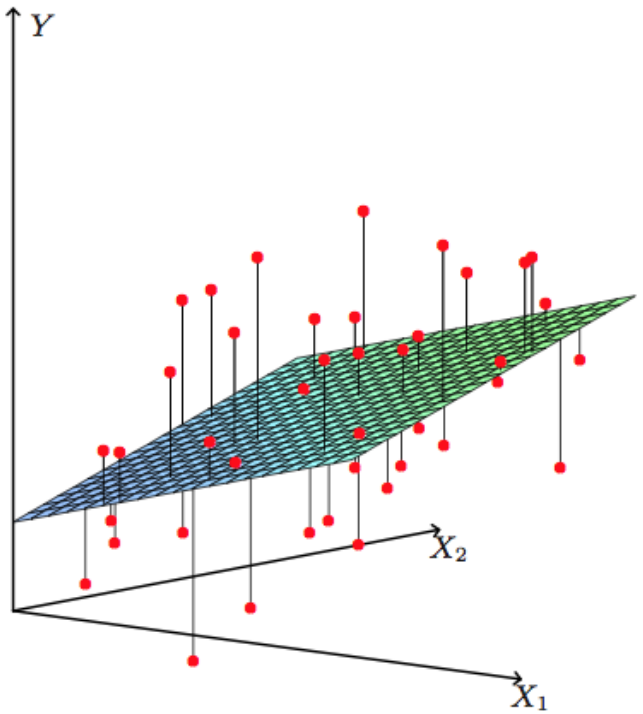
Data Mining and Machine Learning

Kuangnan Fang

Department of Statistics, Xiamen University
Email: xmufkn@xmu.edu.cn



Unsupervised Learning

Unsupervised vs Supervised Learning:

- Most of this course focuses on *supervised learning* methods such as regression and classification.
- In that setting we observe both a set of features $X_1, X_2, \ldots, X_p$ for each object, as well as a response or outcome variable Y. The goal is then to predict Y using $X_1, X_2, \ldots, X_p$.
- Here we instead focus on *unsupervised learning*, we where observe only the features $X_1, X_2, \ldots, X_p$. We are not interested in prediction, because we do not have an associated response variable Y.

The Goals of Unsupervised Learning

- The goal is to discover interesting things about the measurements: is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations?
- We discuss two methods:
  - *principal components analysis*, a tool used for data visualization or data pre-processing before supervised techniques are applied, and
  - *clustering*, a broad class of methods for discovering unknown subgroups in data.

The Challenge of Unsupervised Learning

- Unsupervised learning is more subjective than supervised learning, as there is no simple goal for the analysis, such as prediction of a response.
- But techniques for unsupervised learning are of growing importance in a number of fields:
  - subgroups of breast cancer patients grouped by their gene expression measurements,
  - groups of shoppers characterized by their browsing and purchase histories, 客户细分
  - movies grouped by the ratings assigned by movie viewers.

分成几组，打上标签

Another advantage

- It is often easier to obtain *unlabeled data* — from a lab instrument or a computer — than *labeled data*, which can require human intervention.
- For example it is difficult to automatically assess the overall sentiment of a movie review: is it favorable or not?

PCA vs Clustering

- PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance.
- Clustering looks for homogeneous subgroups among the observations.

Principal Components Analysis

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
- Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.

研究p个变量关系，做两两散点图，需要做 P取2个图，此外两两散点图包含的信息很少，不能很好反应数据之间的关系！
需要一种低维的表示方法，但又包含了数据足够多的信息

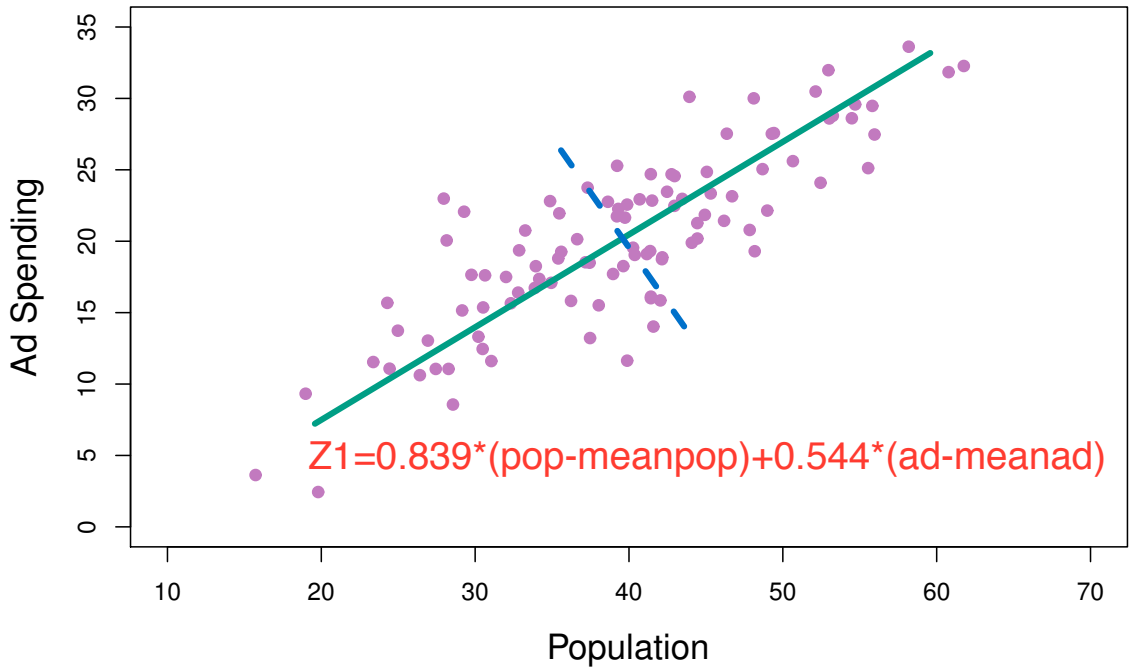
Principal Components Analysis: details

- The *first principal component* of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

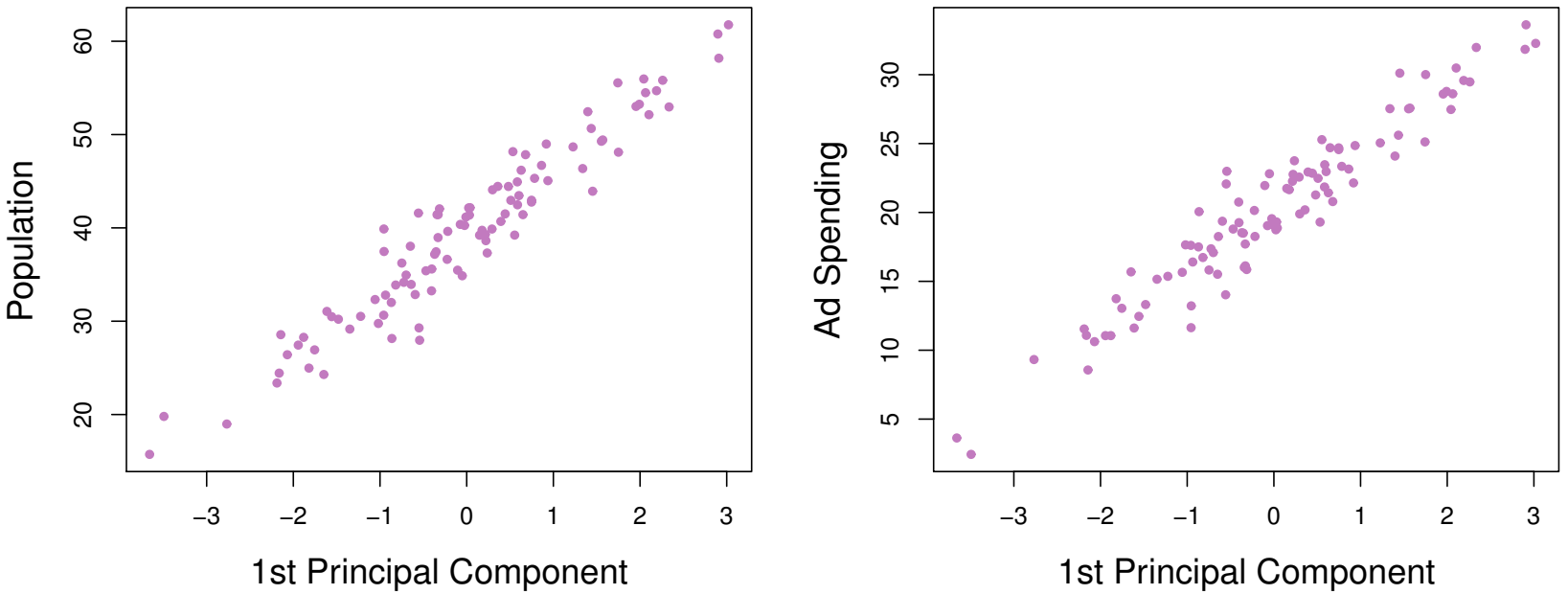
  that has the largest variance. By *normalized*, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the loadings of the first principal component; together, the loadings make up the principal component loading vector, $\phi_1 = (\phi_{11}\ \phi_{21}\ \ldots\ \phi_{p1})^T$.
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

PCA: example



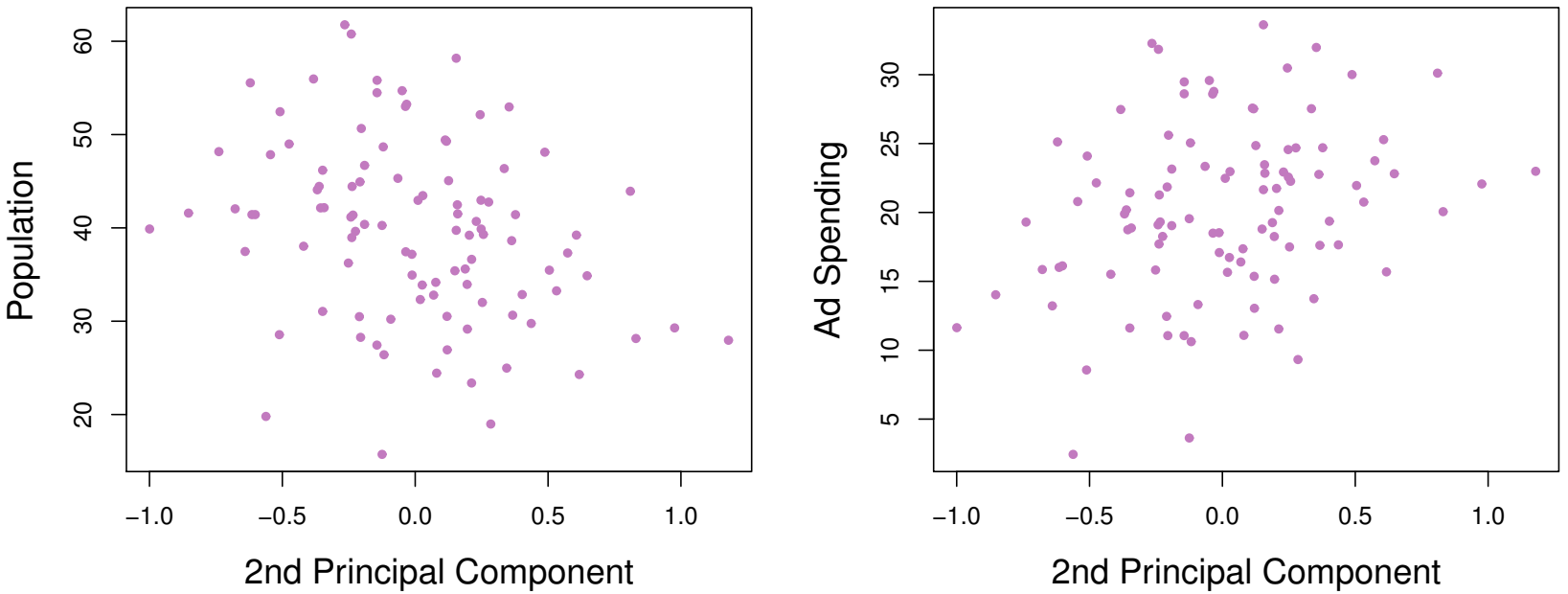
The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component direction, and the blue dashed line indicates the second principal component direction.

Pictures of PCA: continued



Plots of the first principal component scores z_{i1} versus `pop` *and* `ad`*. The relationships are strong.*

Pictures of PCA: continued



Plots of the second principal component scores z_{i2} versus `pop` *and* `ad`*. The relationships are weak.*

Computation of Principal Components

- Suppose we have a $n \times p$ data set $\mathbf{X}$. Since we are only interested in variance, we assume that each of the variables in $\mathbf{X}$ has been centered to have mean zero (that is, the column means of $\mathbf{X}$ are zero).
- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip} \tag{1}$$

  for $i = 1, \ldots, n$ that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- Since each of the x_{ij} has mean zero, then so does z_{i1} (for any values of ϕ_{j1}). Hence the sample variance of the z_{i1} can be written as $\frac{1}{n} \sum_{i=1}^{n} z_{i1}^2$.

Computation: continued

- Plugging in (1) the first principal component loading vector solves the optimization problem

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1.$$

- This problem can be solved via a singular-value decomposition of the matrix $\mathbf{X}$, a standard technique in linear algebra.
- We refer to Z_1 as the first principal component, with realized values $z_{11}, \ldots, z_{n1}$

Geometry of PCA

- The loading vector ϕ_1 with elements $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ defines a direction in feature space along which the data vary the most.
- If we project the n data points $x_1, \ldots, x_n$ onto this direction, the projected values are the principal component scores $z_{11}, \ldots, z_{n1}$ themselves.

Further principal components

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are *uncorrelated* with Z_1.
- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$z_{i2} = \phi_{12} x_{i1} + \phi_{22} x_{i2} + \ldots + \phi_{p2} x_{ip},$$

where ϕ_2 is the second principal component loading vector, with elements $\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$.

Further principal components: continued

- It turns out that constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction ϕ_2 to be orthogonal (perpendicular) to the direction ϕ_1. And so on.
- The principal component directions ϕ_1, ϕ_2, ϕ_3, ... are the ordered sequence of right singular vectors of the matrix $\mathbf{X}$, and the variances of the components are $\frac{1}{n}$ times the squares of the singular values. There are at most $\min(n-1, p)$ principal components.

Illustration

- `USAarrests` data: For each of the fifty states in the United States, the data set contains the number of arrests per $100,000$ residents for each of three crimes: `Assault`, `Murder`, and `Rape`. We also record `UrbanPop` (the percent of the population in each state living in urban areas).
- The principal component score vectors have length $n = 50$, and the principal component loading vectors have length $p = 4$.
- PCA was performed after standardizing each variable to have mean zero and standard deviation one.

USAarrests data: PCA plot

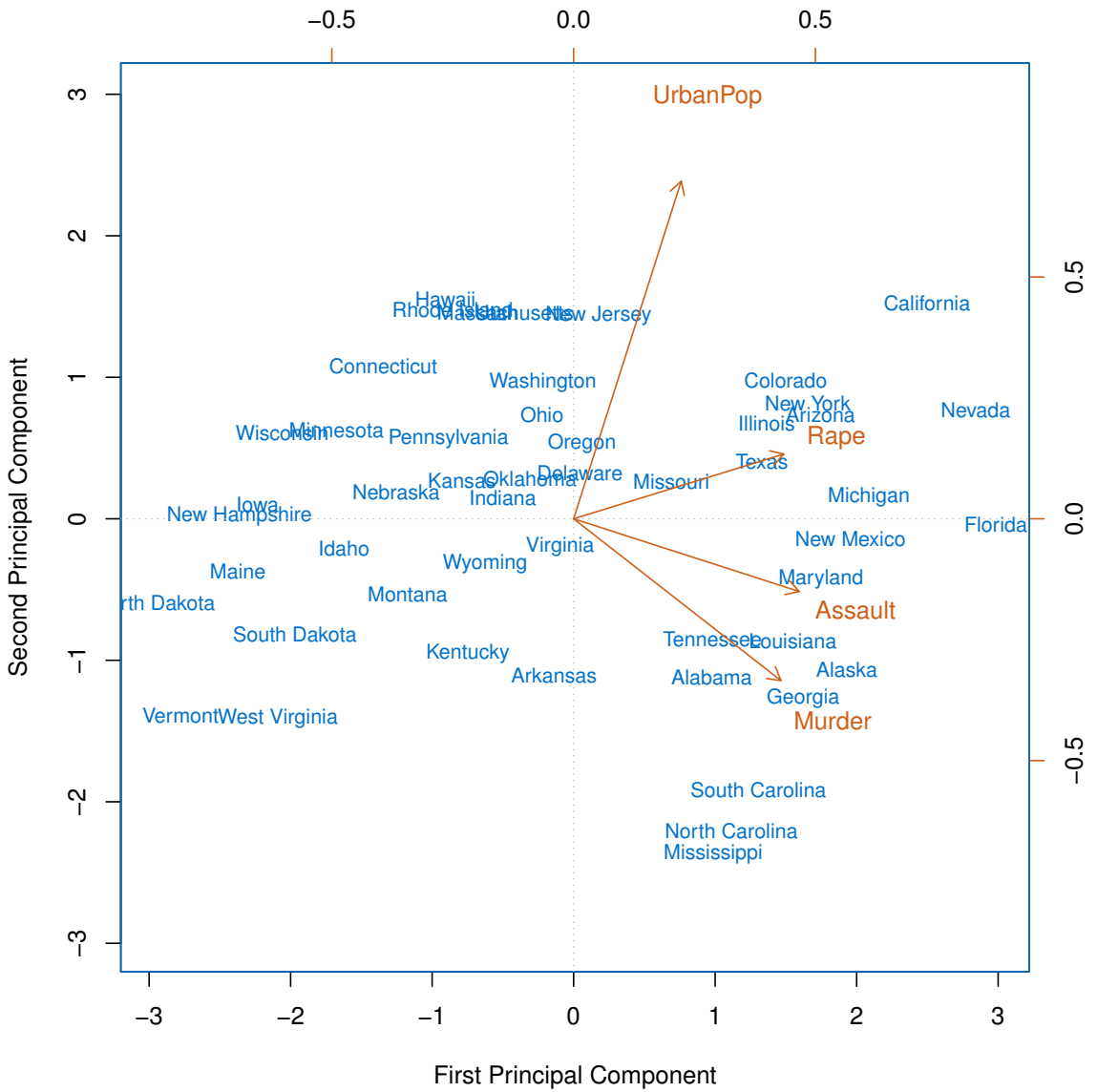


Figure details

The first two principal components for the USArrests data.

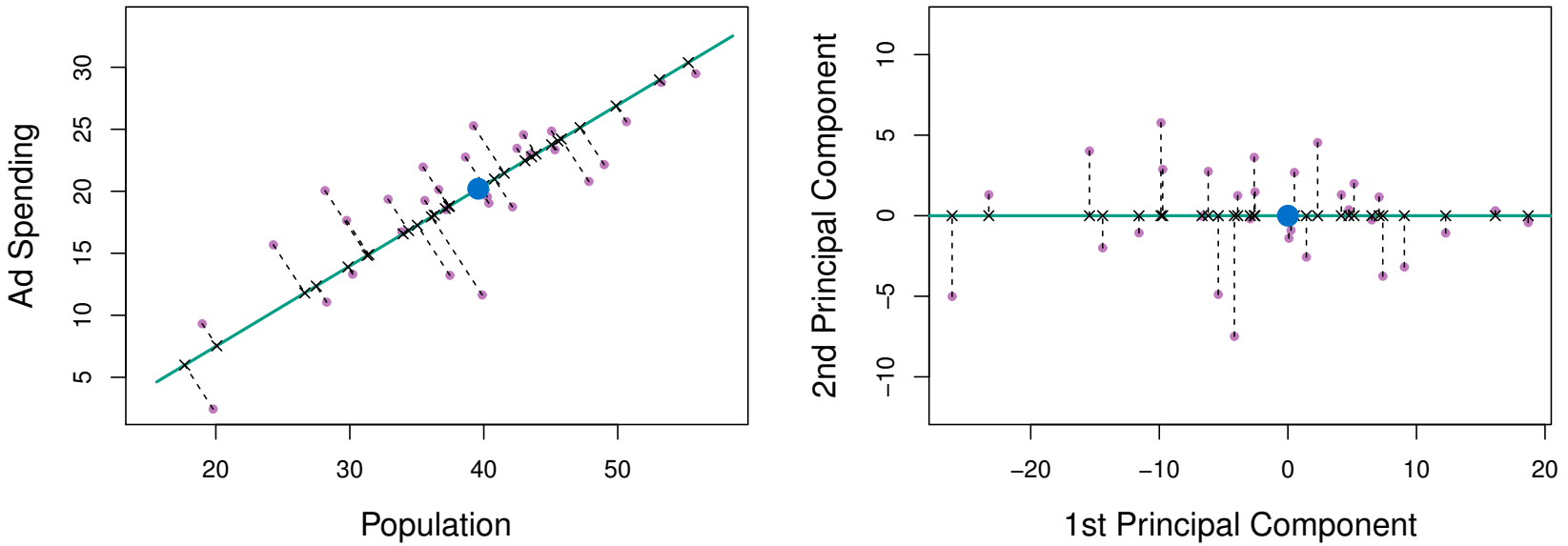
- The blue state names represent the scores for the first two principal components.
- The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for `Rape` on the first component is 0.54, and its loading on the second principal component 0.17 [the word `Rape` is centered at the point $(0.54, 0.17)$].
- This figure is known as a *biplot*, because it displays both the principal component scores and the principal component loadings.

PCA loadings

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

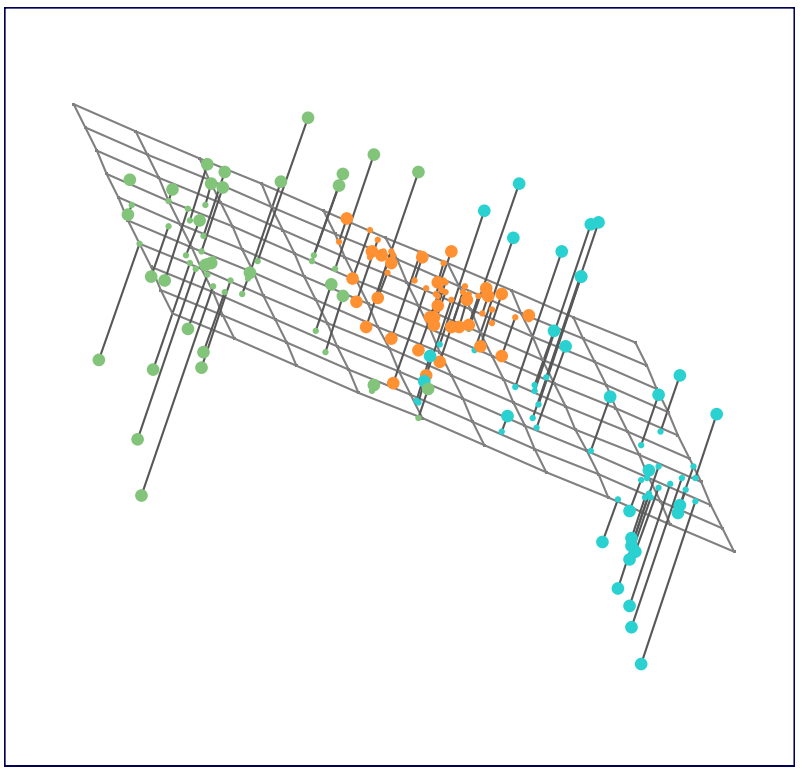
Pictures of PCA: continued

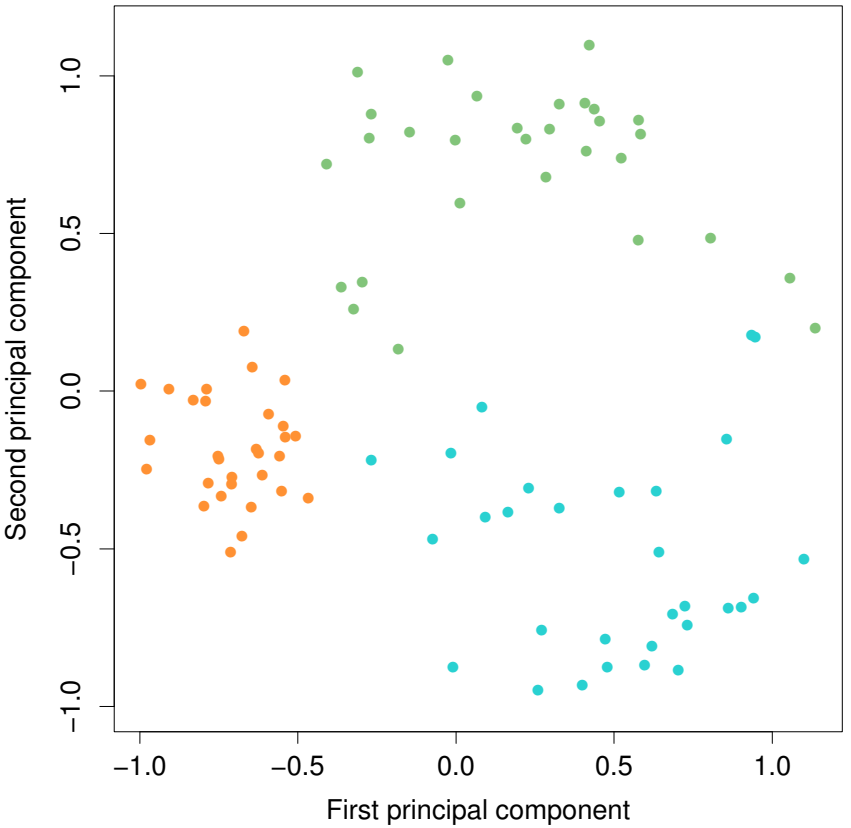
Another Interpretation of PCA



A subset of the advertising data. Left: *The first principal component, chosen to minimize the sum of the squared perpendicular distances to each point, is shown in green. These distances are represented using the black dashed line segments.* Right: *The left-hand panel has been rotated so that the first principal component lies on the x-axis.*

Another Interpretation of Principal Components



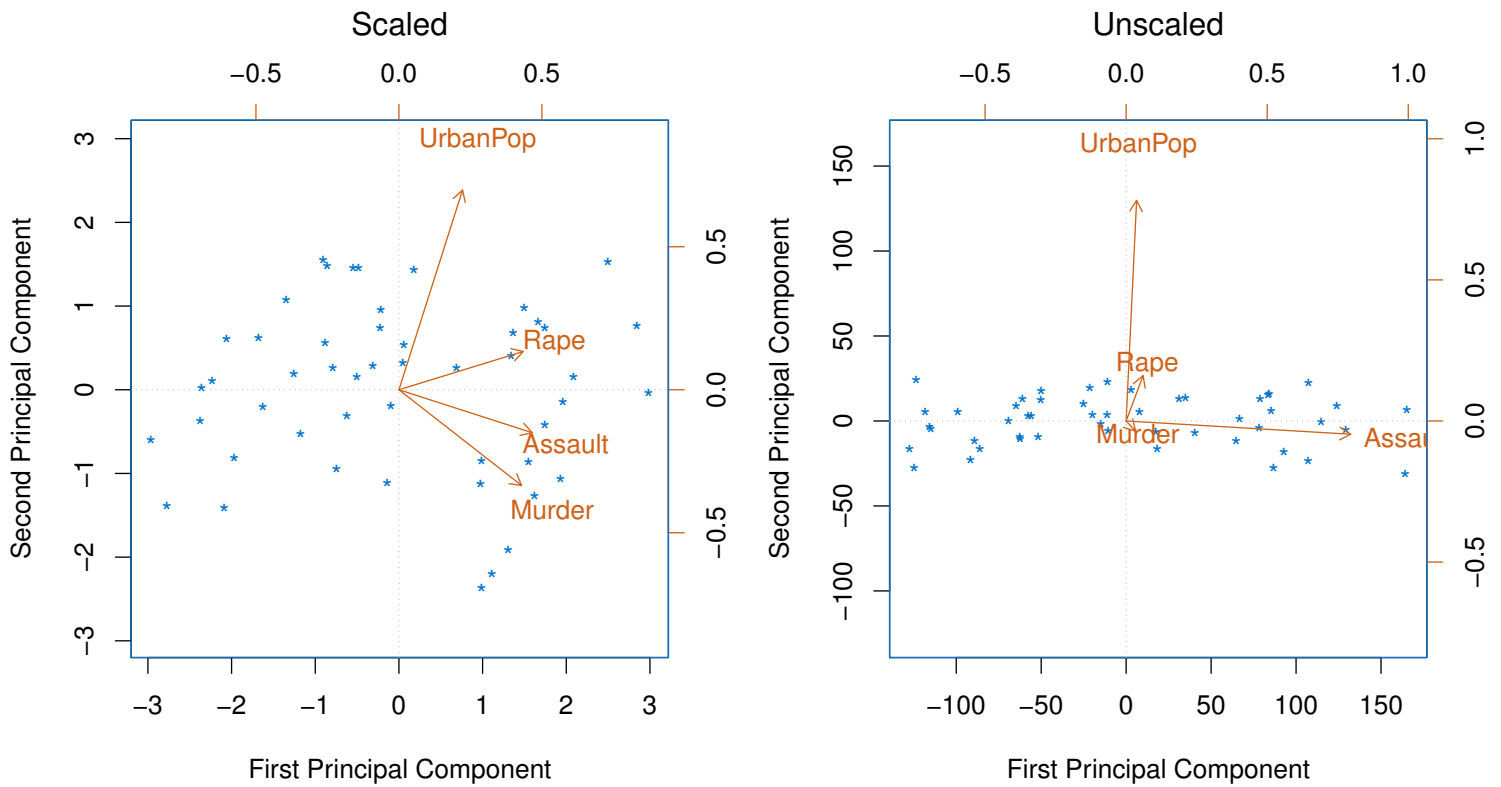


PCA find the hyperplane closest to the observations

- The first principal component loading vector has a very special property: it defines the line in p-dimensional space that is *closest* to the n observations (using average squared Euclidean distance as a measure of closeness)
- The notion of principal components as the dimensions that are closest to the n observations extends beyond just the first principal component.
- For instance, the first two principal components of a data set span the plane that is closest to the n observations, in terms of average squared Euclidean distance.

Scaling of the variables matters

- If the variables are in different units, scaling each to have standard deviation equal to one is recommended.
- If they are in the same units, you might or might not scale the variables.



Proportion Variance Explained

- To understand the strength of each component, we are interested in knowing the proportion of variance explained (PVE) by each one.
- The *total variance* present in a data set (assuming that the variables have been centered to have mean zero) is defined as

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2,$$

  and the variance explained by the mth principal component is

$$\mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2.$$

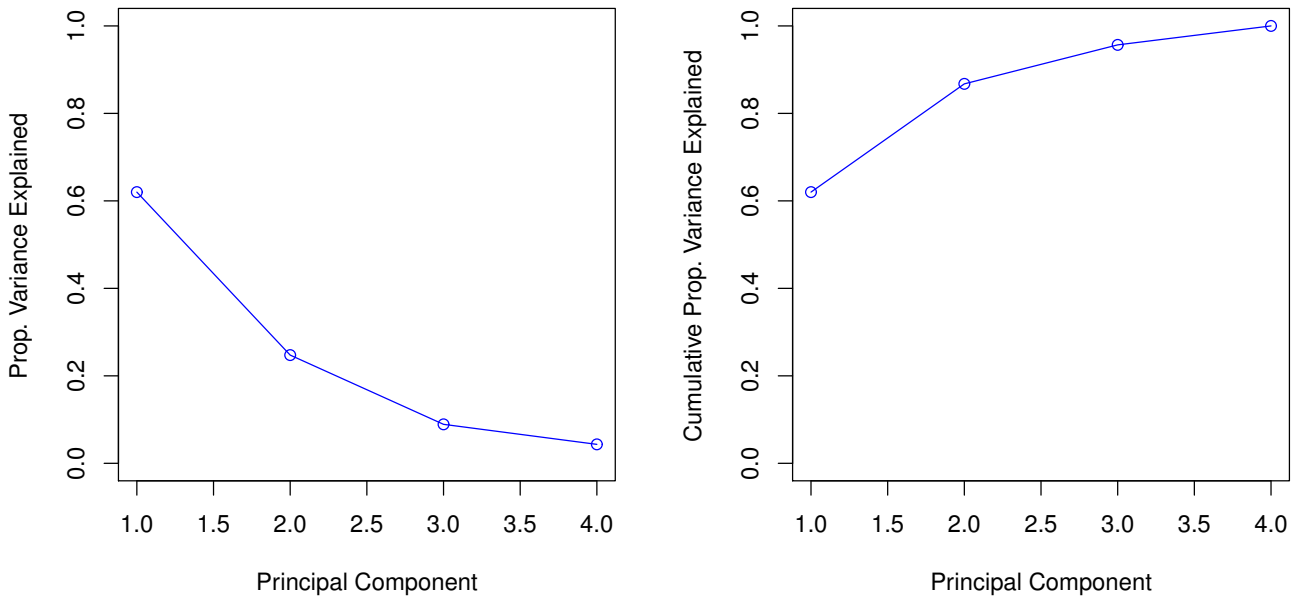
- It can be shown that $\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{m=1}^{M} \mathrm{Var}(Z_m)$, with $M = \min(n-1, p)$.

Proportion Variance Explained: continued

- Therefore, the PVE of the mth principal component is given by the positive quantity between 0 and 1

$$\frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2}.$$

- The PVEs sum to one. We sometimes display the cumulative PVEs.



How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
  - *Why not?*

How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

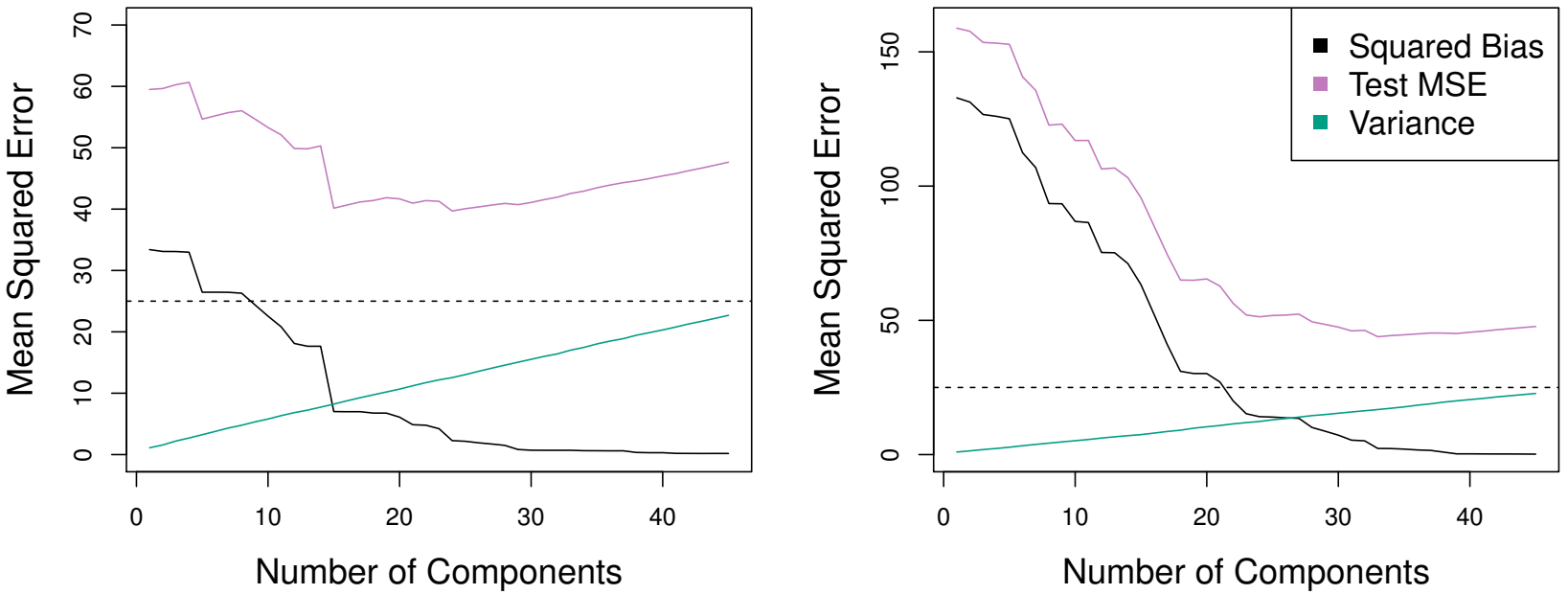
- No simple answer to this question, as cross-validation is not available for this purpose.
  - *Why not?*
  - When could we use cross-validation to select the number of components?

How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

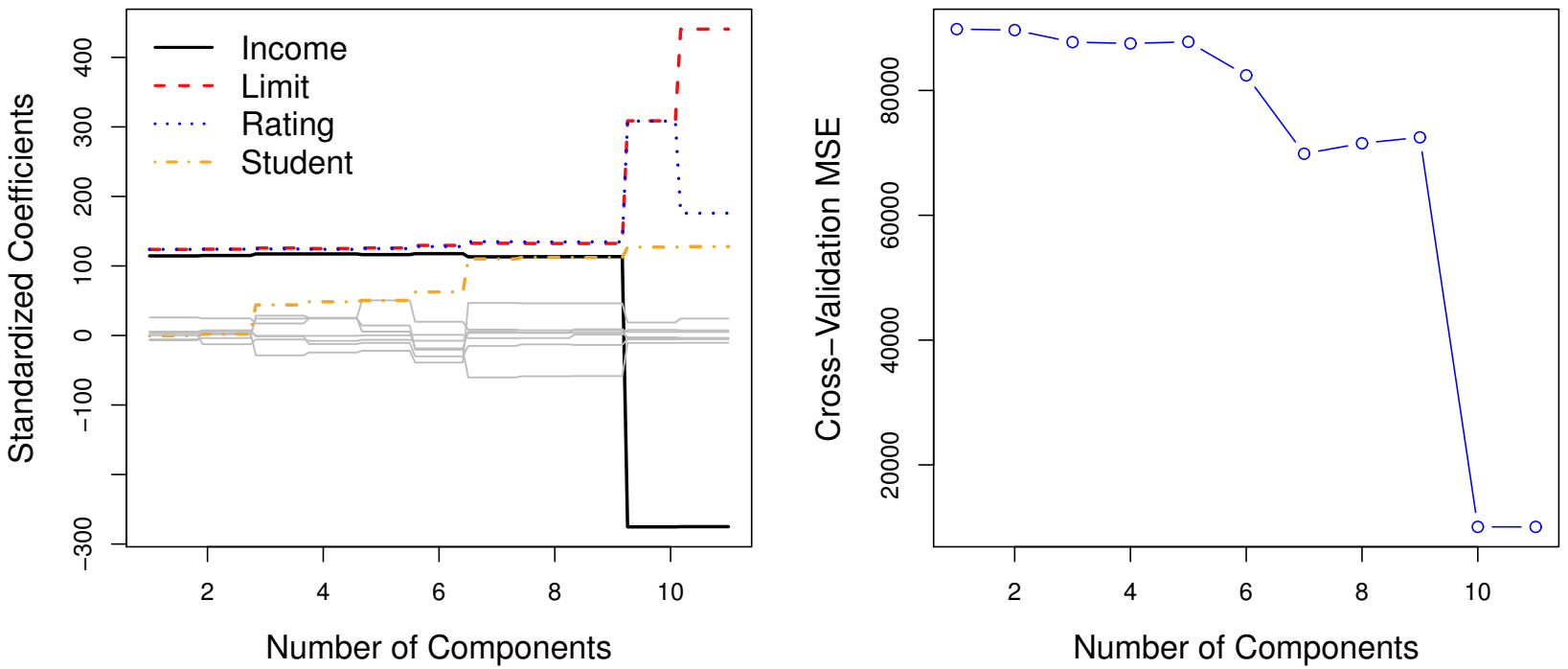
- No simple answer to this question, as cross-validation is not available for this purpose.
  - *Why not?*
  - When could we use cross-validation to select the number of components?
- the "scree plot" on the previous slide can be used as a guide: we look for an "elbow".

Application to Principal Components Regression



PCR was applied to two simulated data sets. The black, green, and purple lines correspond to squared bias, variance, and test mean squared error, respectively. Left: *Simulated data from slide 32.* Right: *Simulated data from slide 39.*

Choosing the number of directions M



Left: *PCR standardized coefficient estimates on the* `Credit` *data set for different values of* M. Right: *The* 10*-fold cross validation MSE obtained using PCR, as a function of* M.

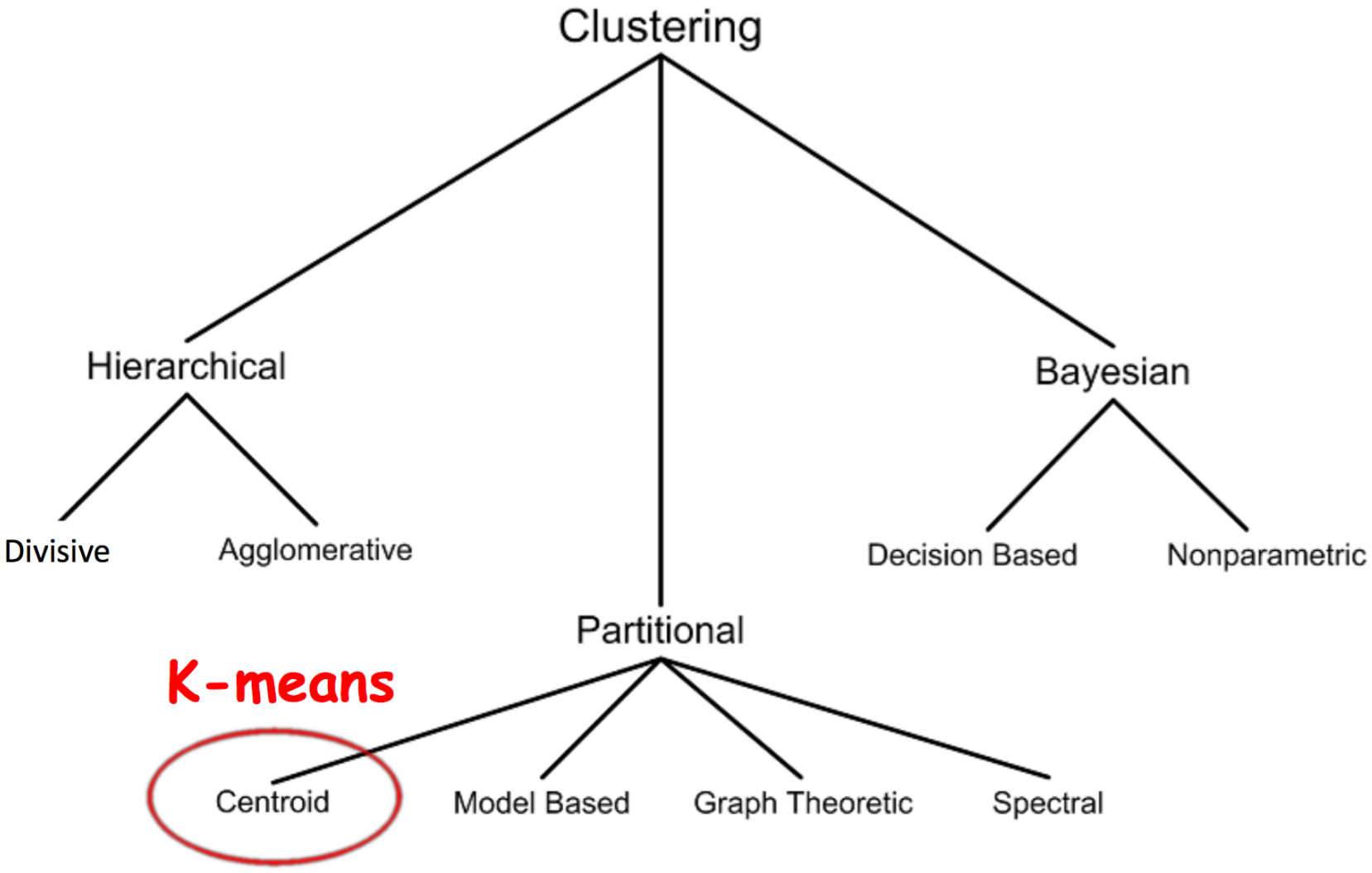
Clustering

- *Clustering* refers to a very broad set of techniques for finding *subgroups*, or *clusters*, in a data set.
- We seek a partition of the data into distinct groups so that the observations within each group are quite similar to each other, 组内相似，组间差异大
- It make this concrete, we must define what it means for two or more observations to be *similar* or *different*.
- Indeed, this is often a domain-specific consideration that must be made based on knowledge of the data being studied.

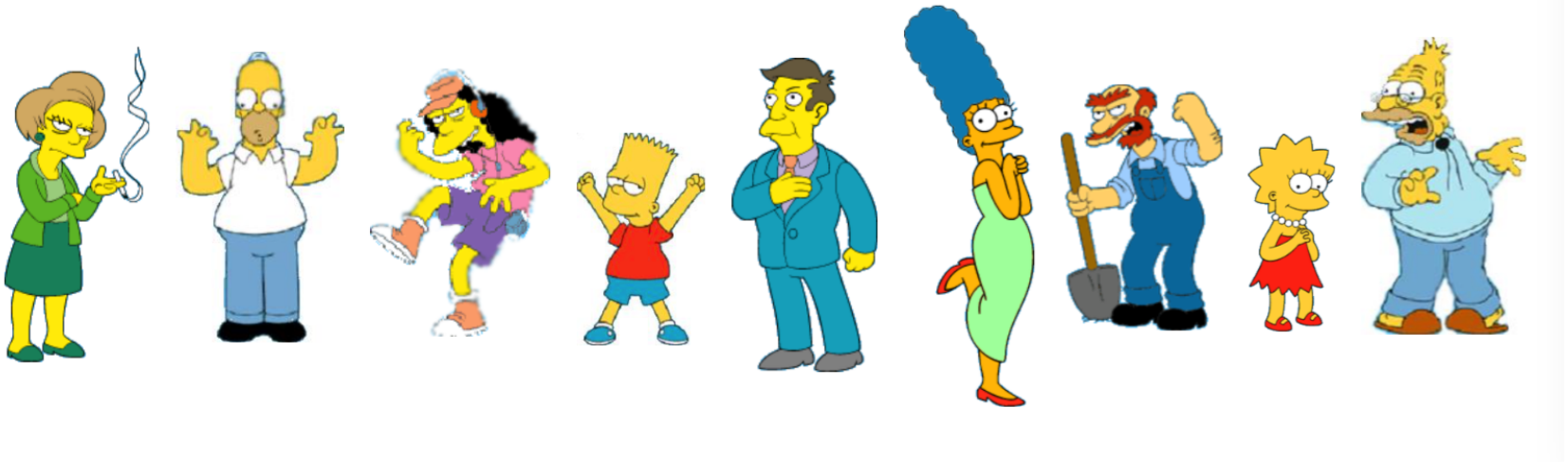
Clustering for Market Segmentation

- Suppose we have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our goal is to perform *market segmentation* by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.
- The task of performing market segmentation amounts to clustering the people in the data set.

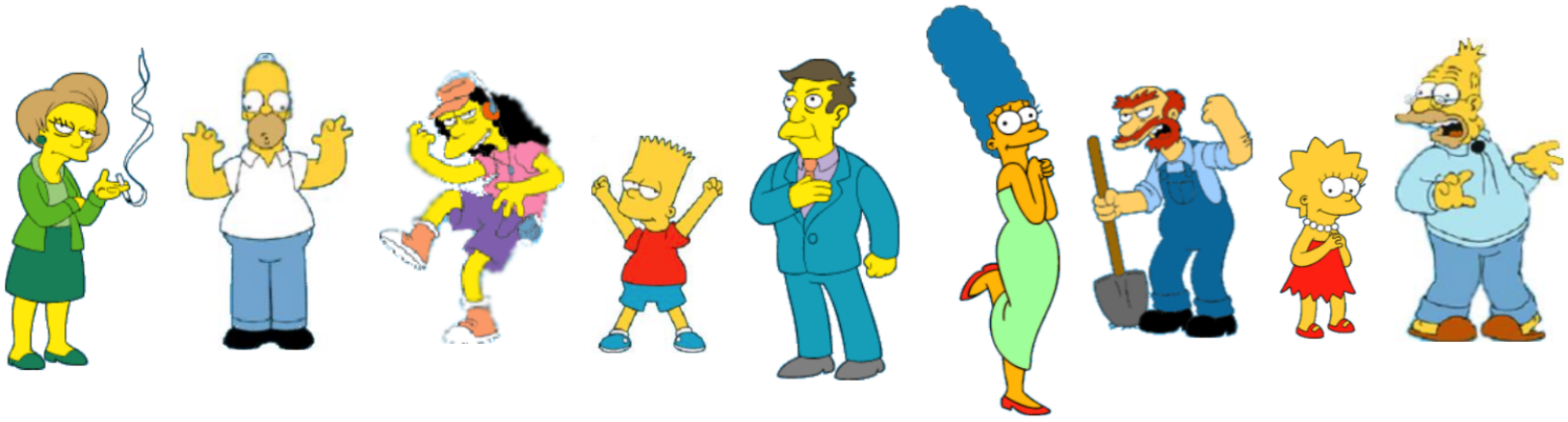
Clustering Methods



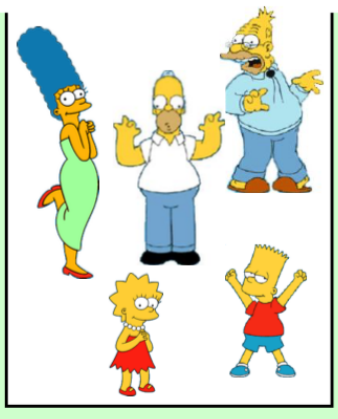
What is a natural grouping among these objects?



What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family

School Employees

Females

Males

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.

Dissimilarities Based on Attributes

Most often we have measurements x_{ij} for $i = 1, 2, \ldots, N$, on variables $j = 1, 2, \ldots, p$(also called attributes). since most of the popular clustering algorithms take a dissimilarity matrix as their input, we must first construct pairwise dissimilarities between the observations. In the most common case, we define a dissimilarity $d_j\left(x_{ij}, x_{ij}\right)$ between values of the jth attribute, and then define

$$D\left(x_i, x_{i'}\right) = \sum_{j=1}^{p} d_j\left(x_{ij}, x_{i'j}\right)$$

as the dissimilarity between objects i and i'. By far the most common choice is squared distance

$$d_j\left(x_{ij}, xi'j\right) = \left(x_{ij} - x_{i'j}\right)^2$$

However, other choices are possible, and can lead to potentially different results. For nonquantitative attributes, squared distance may not be appropriate. In addition, it is sometimes desirable to weight attributes differently.

Dissimilarities Based on Attributes

Quantitative variables. Measurements of this type of variable or attribute are represented by continuous real-valued numbers. It is natural to define the "error" between them as a monotone-increasing function of their absolute difference

$$d\left(x_i, x_{i'}\right) = l\left(\left|x_i - x_{i'}\right|\right)$$

Besides squared-error loss $\left(x_i - x_{i'}\right)^2$, a common choice is the identity (absolute error). The former places more emphasis on larger differences than smaller ones. Alternatively, clustering can be based on the correlation

$$\rho\left(x_i, x_{i'}\right) = \frac{\sum_j \left(x_{ij} - \bar{x}_i\right)\left(x_{i'j} - \bar{x}_{i'}\right)}{\sqrt{\sum_j \left(x_{ij} - \bar{x}_i\right)^2 \sum_j \left(x_{i'j} - \bar{x}_{i'}\right)^2}} \tag{14.22}$$

with $\bar{x}_i = \sum_j x_{ij}/p$. Note that this is averaged over variables, not observations. If the observations are first standardized, then $\sum_j \left(x_{ij} - x_{i'j}\right)^2 \propto 2\left(1 - \rho\left(x_i, x_{i'}\right)\right)$. Hence clustering based on correlation (similarity) is equivalent to that based on squared distance (dissimilarity).

Dissimilarities Based on Attributes

Ordinal variables. The values of this type of variable are often represented as contiguous integers, and the realizable values are considered to be an ordered set. Examples are academic grades (A, B, C, D, F), degree of preference (can't stand, dislike, OK, like, terrific). Rank data are a special kind of ordinal data. Error measures for ordinal variables are generally defined by replacing their M original values with

$$\frac{i-1/2}{M}, i = 1, \ldots, M \tag{14.23}$$

in the prescribed order of their original values. They are then treated as quantitative variables on this scale.

Combinatorial Algorithms

The most popular clustering algorithms directly assign each observation to a group or cluster without regard to a probability model describing the data. Each observation is uniquely labeled by an integer $i \in \{1, \cdots, N\}$. A prespecified number of clusters $K < N$ is postulated, and each one is labeled by an integer $k \in \{1, \ldots, K\}$. Each observation is assigned to one and only one cluster. These assignments can be characterized by a manyto-one mapping, or *encoder* $k = C(i)$, that assigns the ith observation to the kth cluster. One seeks the particular encoder $C^*(i)$ that achieves the required goal (details below), based on the dissimilarities $d(x_i, x_{i'})$ between every pair of observations. These are specified by the user as described above. Generally, the encoder $C(i)$ is explicitly delineated by giving its value (cluster assignment) for each observation i. Thus, the "parameter" of the procedure are the individual cluster assignments for each of the N observations. These are adjusted so as to *minimize* a "loss" function that characterizes the degree to which the clustering goal is *not* met.

Combinatorial Algorithms

One approach is to directly specify a mathematical loss function and attempt to minimize it through some combinatorial optimization algorithm. since the goal is to assign close points to the same cluster, a natural loss (or "energ") function would be

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} d\left(x_i, x_{i'}\right) \tag{14.28}$$

This criterion characterizes the extent to which observations assigned to the same cluster tend to be close to one another. It is sometimes referred to as the "within cluste" point scatter since

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(i')=k} d_{ii'} + \sum_{C(i')\neq k} d_{ii'}\right)$$

or

$$T = W(C) + B(C)$$

Combinatorial Algorithms

where $d_{ii'} = d(x_i, x_{i'})$. Here T is the *total* point scatter, which is a constant given the data, independent of cluster assignment. The quantity

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{ii'} \tag{14.29}$$

is the *between-cluster* point scatter. This will tend to be large when observations assigned to different clusters are far apart. Thus one has

$$W(C) = T - B(C)$$

and minimizing $W(C)$ is equivalent to *maximizing* $B(C)$.
Cluster analysis by combinatorial optimization is straightforward in principle. One simply minimizes W or equivalently maximizes B over all possible assignments of the N data points to K clusters. Unfortunately, such optimization by complete enumeration is feasible only for very small data sets. The number of distinct assignments is (Jain and Dubes, 1988)

$$S(N,K) = \frac{1}{K!}\sum_{k=1}^{K}(-1)^{K-k}\left(\begin{array}{c} K \\ k \end{array}\right)k^N \tag{14.30}$$

Combinatorial Algorithms

For example, $S(10, 4) = 34,105$ which is quite feasible. But, $S(N, K)$ grows very rapidly with increasing values of its arguments. Already $S(19, 4) \simeq 10^{10}$, and most clustering problems involve much larger data sets than $N = 19$. For this reason, practical clustering algorithms are able to examine only a very small fraction of all possible encoders $k = C(i)$. The goal is to identify a small subset that is likely to contain the optimal one, or at least a good suboptimal partition.

Details of K-means clustering

Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

不重不漏

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.
2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

For instance, if the ith observation is in the kth cluster, then $i \in C_k$.

Details of K-means clustering: continued

- The idea behind K-means clustering is that a *good* clustering is one for which the *within-cluster variation* is as small as possible.
- The within-cluster variation for cluster C_k is a measure $\text{WCV}(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Hence we want to solve the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \text{WCV}(C_k) \right\}. \tag{2}$$

- In words, this formula says that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible.

How to define within-cluster variation?

- Typically we use Euclidean distance

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2, \tag{3}$$

where $|C_k|$ denotes the number of observations in the kth cluster.

- Combining (2) and (3) gives the optimization problem that defines K-means clustering,

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}. \tag{4}$$

最优化该问题，需要搜索所有解找到全局最优解是很困难的，有K^n种方法，只能找局部最优解

K-Means Clustering Algorithm

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   - 2.1 For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   - 2.2 Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective (4) at each step. *Why?*

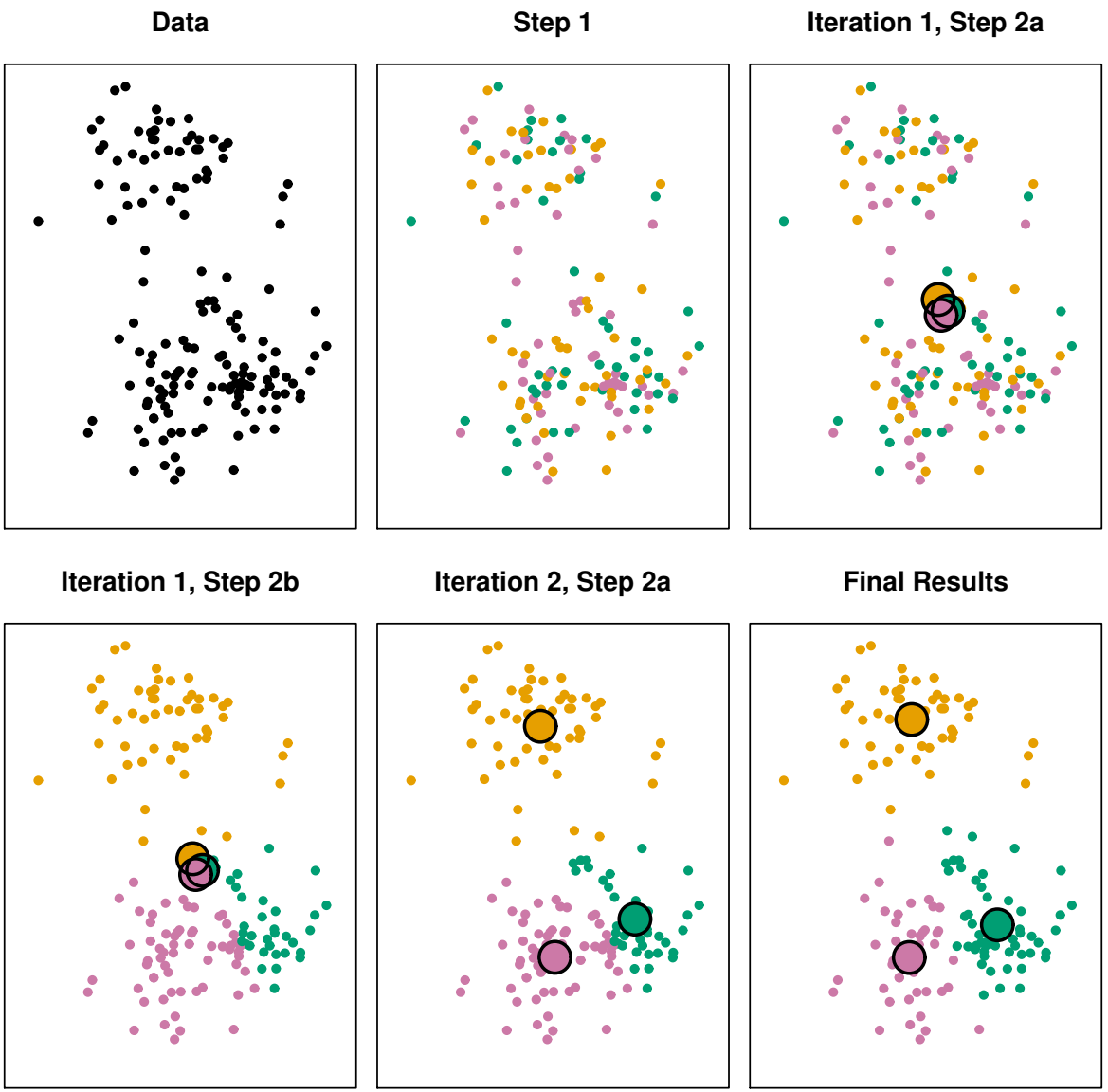
Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective (4) at each step. *Why?* Note that

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2,$$

  where $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i\in C_k} x_{ij}$ is the mean for feature j in cluster C_k.
- however it is not guaranteed to give the global minimum. *Why not?*

Example


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

由于是局部最优解
需要尝试多种初始值

Example: different starting values

Weakness of K-means

- K-means algorithm is appropriate when the dissimilarity measure is taken to be squared Euclidean distance. This requires all of the variables to be of the quantitative type.
- For categorical data, K -mode - the centroid is represented by most frequent values.
- The user needs to specify K.
- The algorithm is local optimum, isn't global optimum. It is sensitive to initical seeds.
- The algorithm is sensitive to outliers.

K-medoids

Algorithm 14.2 *K-medoids Clustering.*

1. For a given cluster assignment C find the observation in the cluster minimizing total distance to other points in that cluster:

$$i_k^* = \underset{\{i:C(i)=k\}}{\operatorname{argmin}} \sum_{C(i')=k} D(x_i, x_{i'}). \tag{14.35}$$

   Then $m_k = x_{i_k^*}$, $k = 1, 2, \ldots, K$ are the current estimates of the cluster centers.

2. Given a current set of cluster centers $\{m_1, \ldots, m_K\}$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} D(x_i, m_k). \tag{14.36}$$

3. Iterate steps 1 and 2 until the assignments do not change.

K-medoids

Solving (14.32) for each provisional cluster k requires an amount of computation proportional to the number of observations assigned to it, whereas for solving (14.35) the computation increases to $O\left(N_k^2\right)$. Given a set of cluster "centers" $\{i_1, \ldots, i_K\}$, obtaining the new assignments

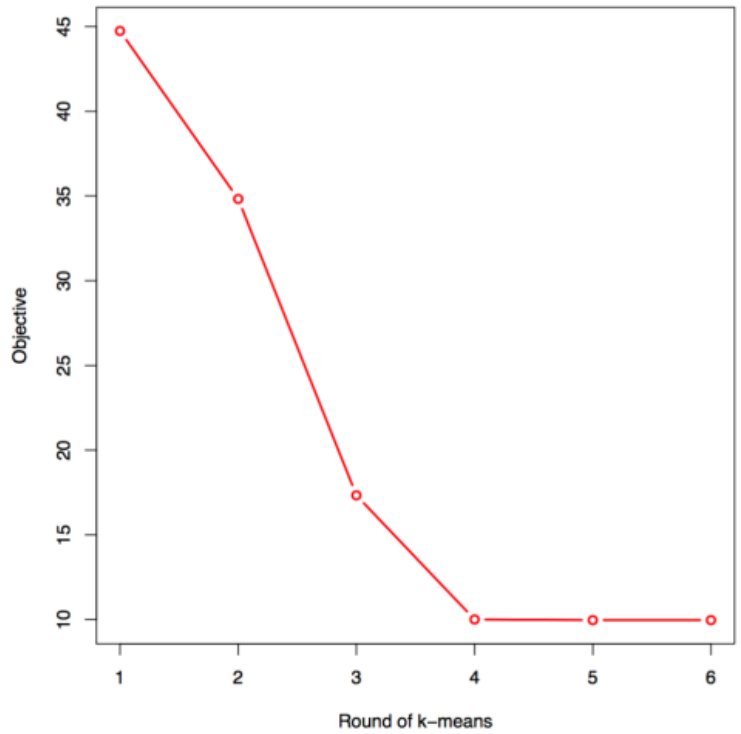
$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} d_{ii_k^*} \tag{14.37}$$

requires computation proportional to $K \cdot N$ as before. Thus, K -medoids is far more computationally intensive than K -means. Alternating between (14.35) and (14.37) represents a particular heuristic search strategy for trying to solve

$$\min_{C, \{i_k\}_1^K} \sum_{k=1}^{K} \sum_{C(i)=k} d_{ii_k}. \tag{14.38}$$

Choosing K

- Choosing K is a nagging problem in cluster analysis.
- Sometimes, the problem determines K. For example, clustering customers for K group in a business.
- Usually, we seek the natural clustering, but what does this mean?
- Plot the objective function VS K. Elbow finding.

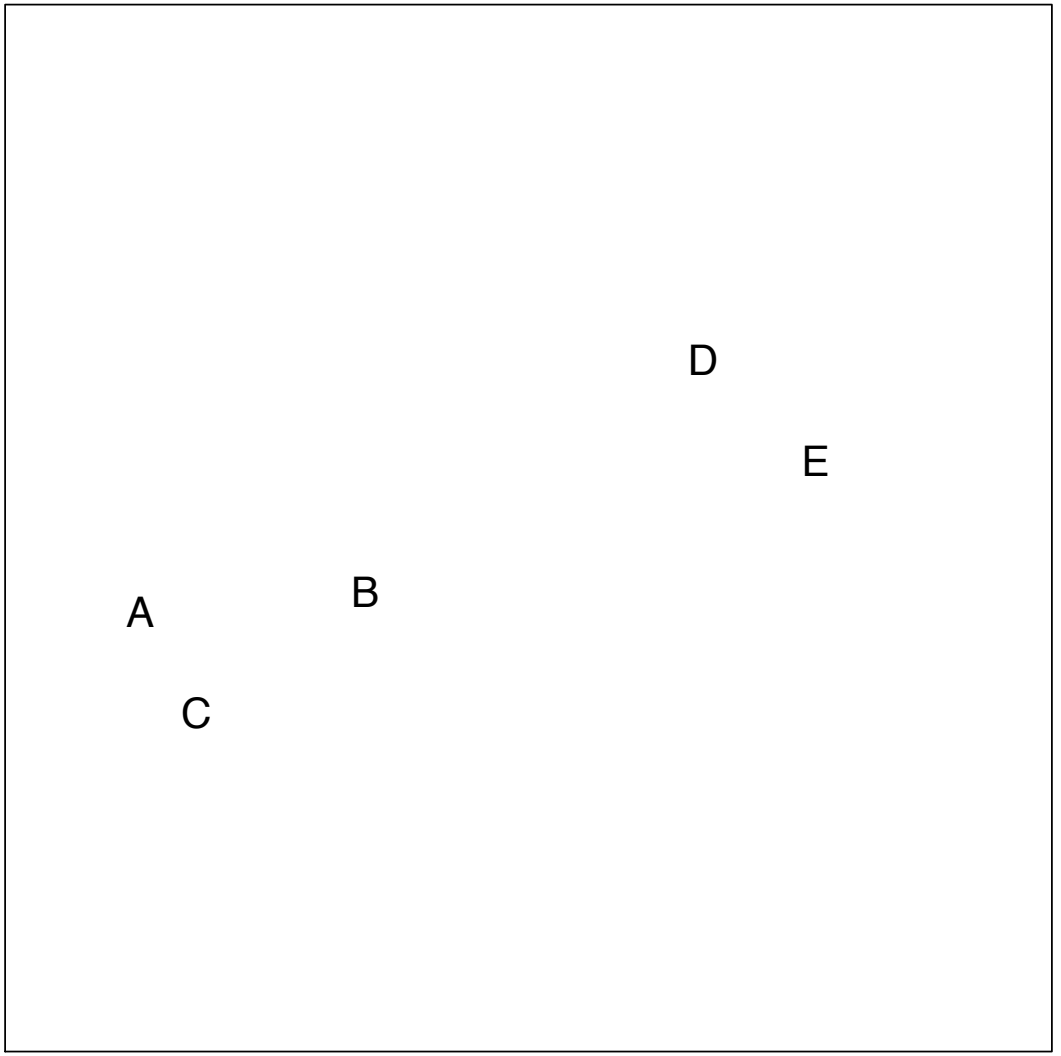


Hierarchical Clustering

- K-means clustering requires us to pre-specify the number of clusters K. This can be a disadvantage (later we discuss strategies for choosing K)
- *Hierarchical clustering* is an alternative approach which does not require that we commit to a particular choice of K.
- In this section, we describe *bottom-up* or *agglomerative* clustering. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.

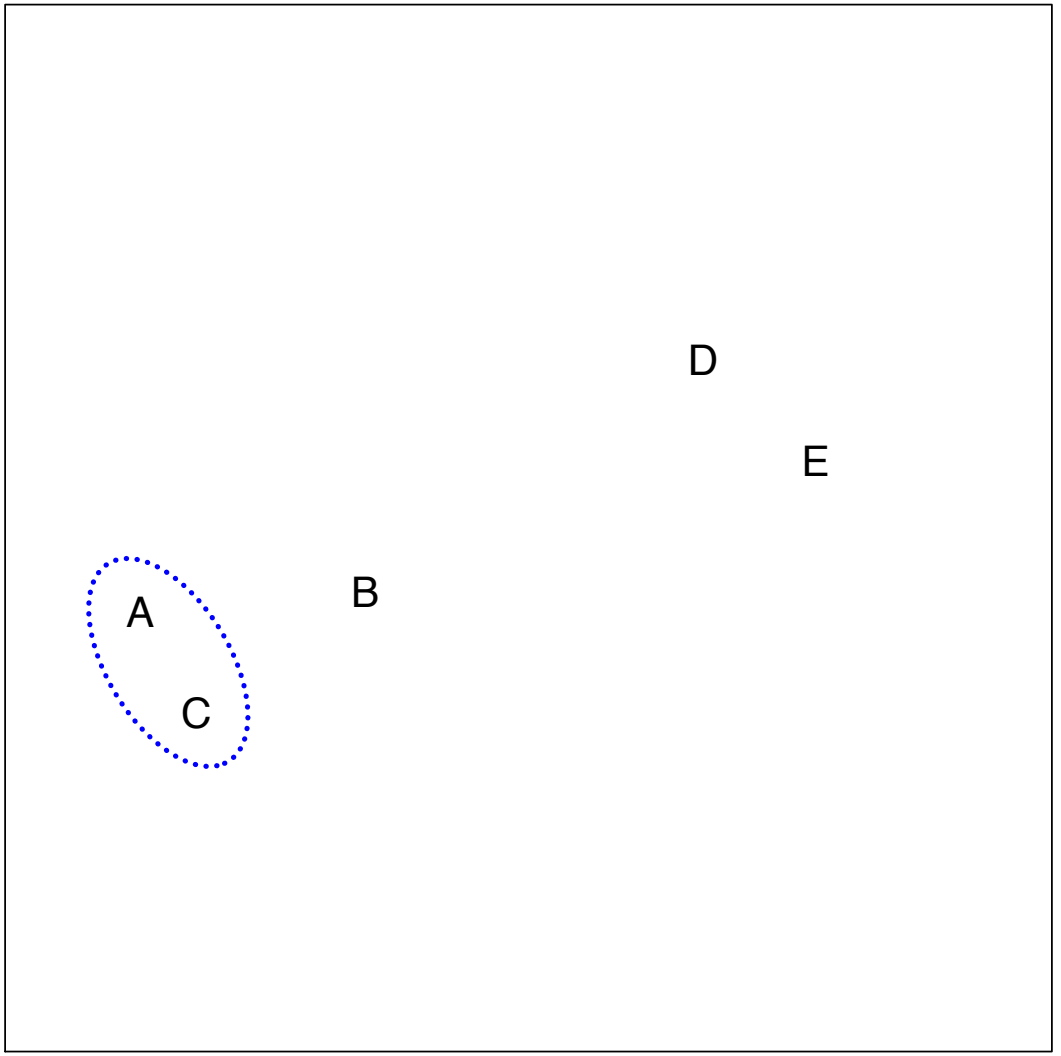
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



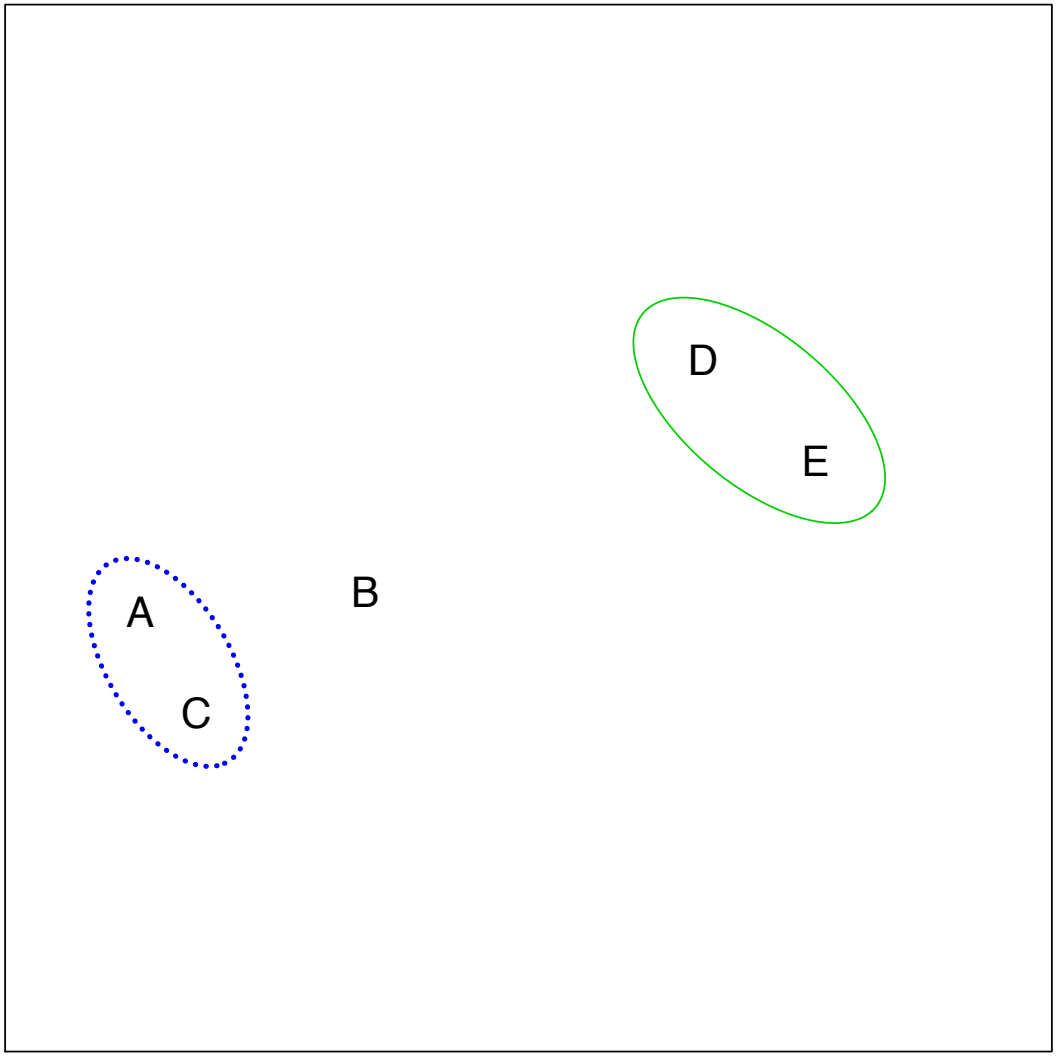
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



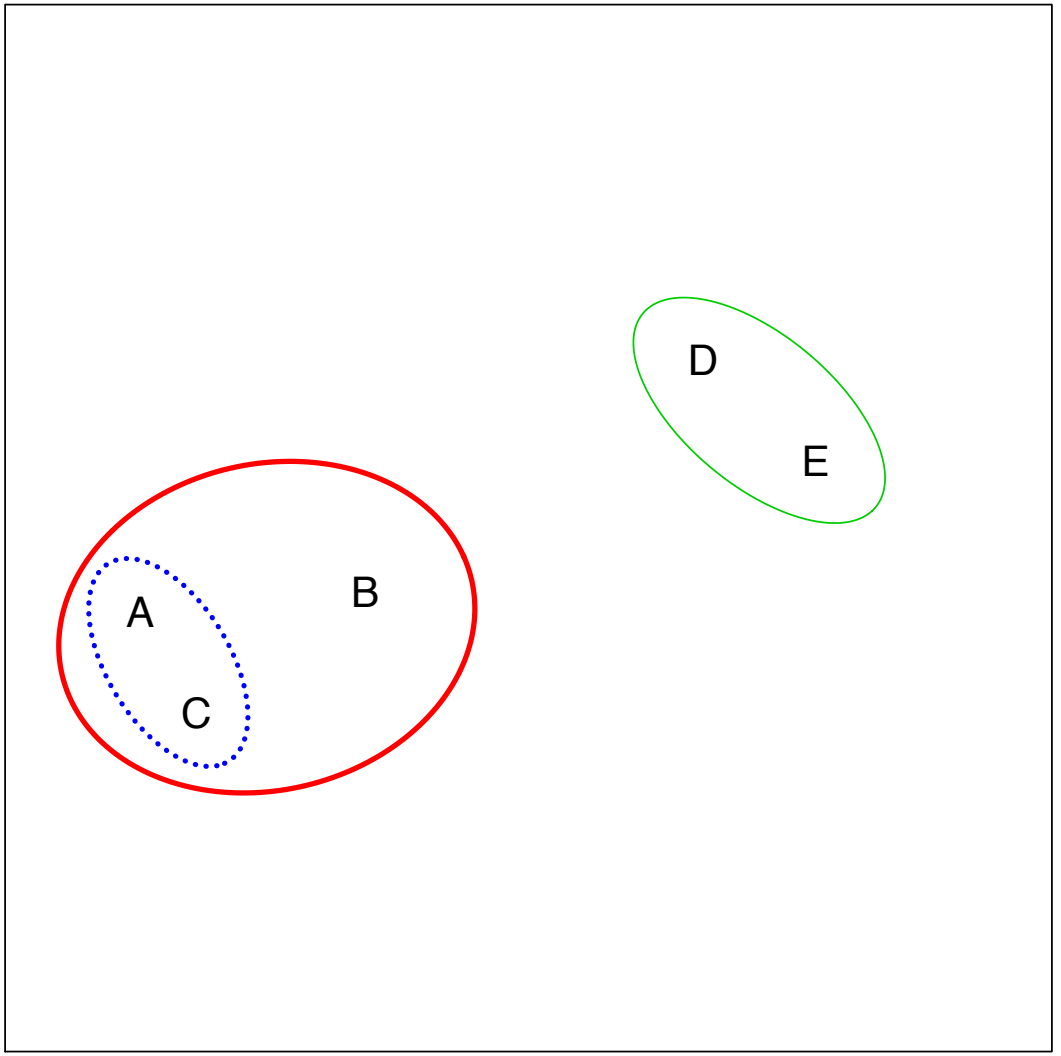
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



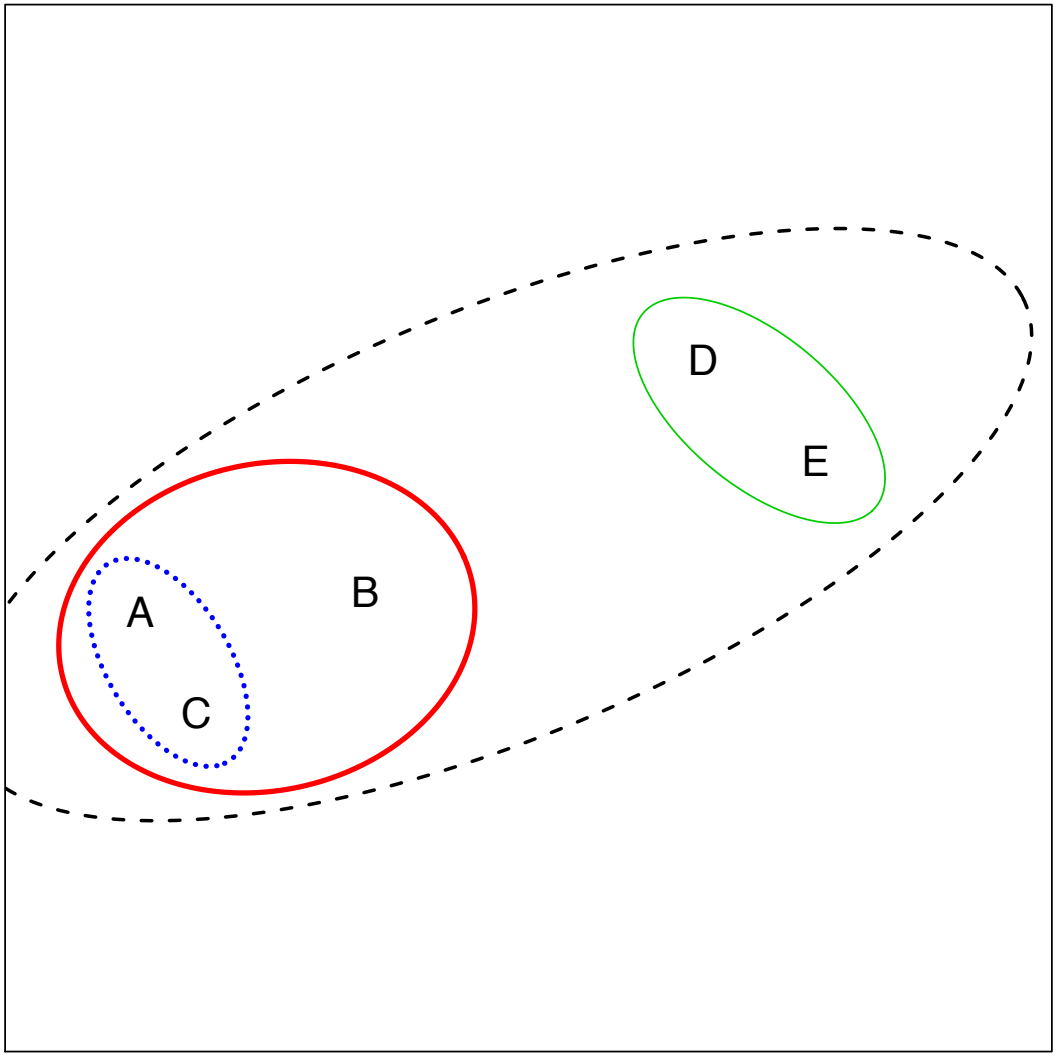
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



Hierarchical Clustering: the idea

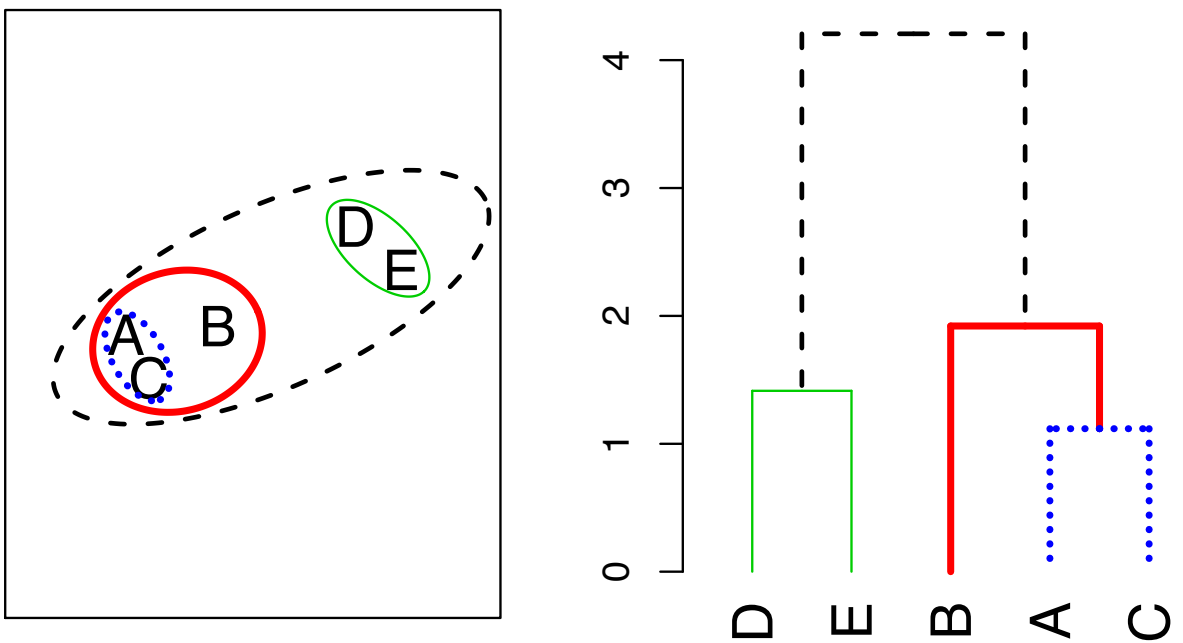
Builds a hierarchy in a "bottom-up" fashion...



Hierarchical Clustering Algorithm

The approach in words:

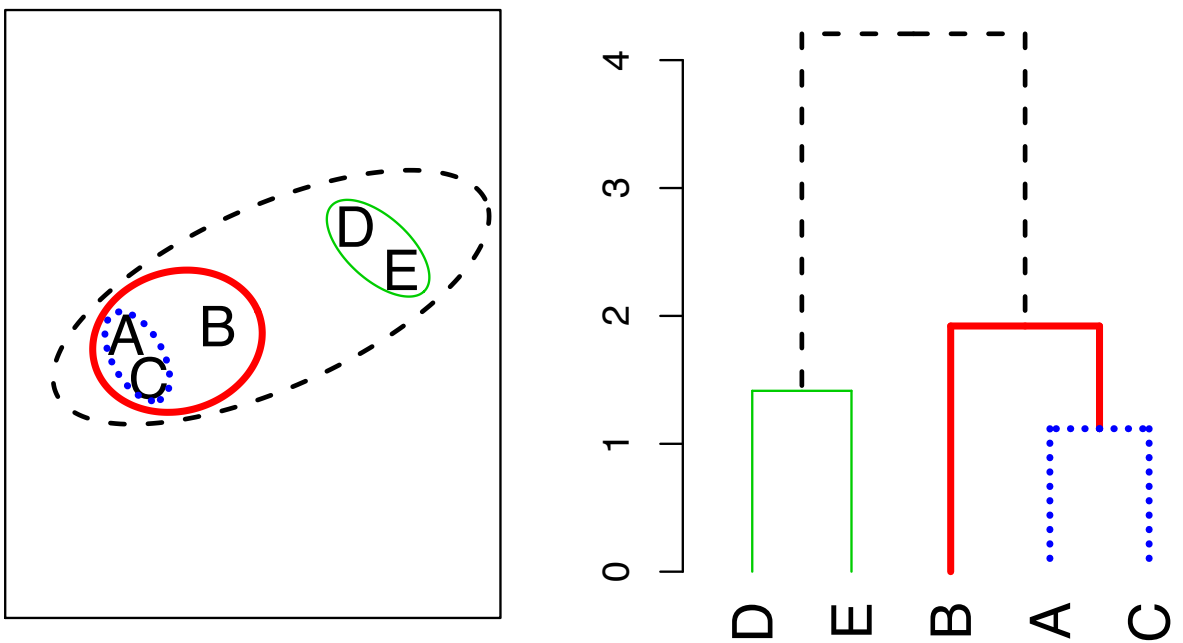
- Start with each point in its own cluster.
- Identify the closest two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.



Hierarchical Clustering Algorithm

The approach in words:

- Start with each point in its own cluster.
- Identify the closest two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.



Types of Linkage

Linkage	*Description*
Complete	Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities.
Average	Mean inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Hierarchical Clustering

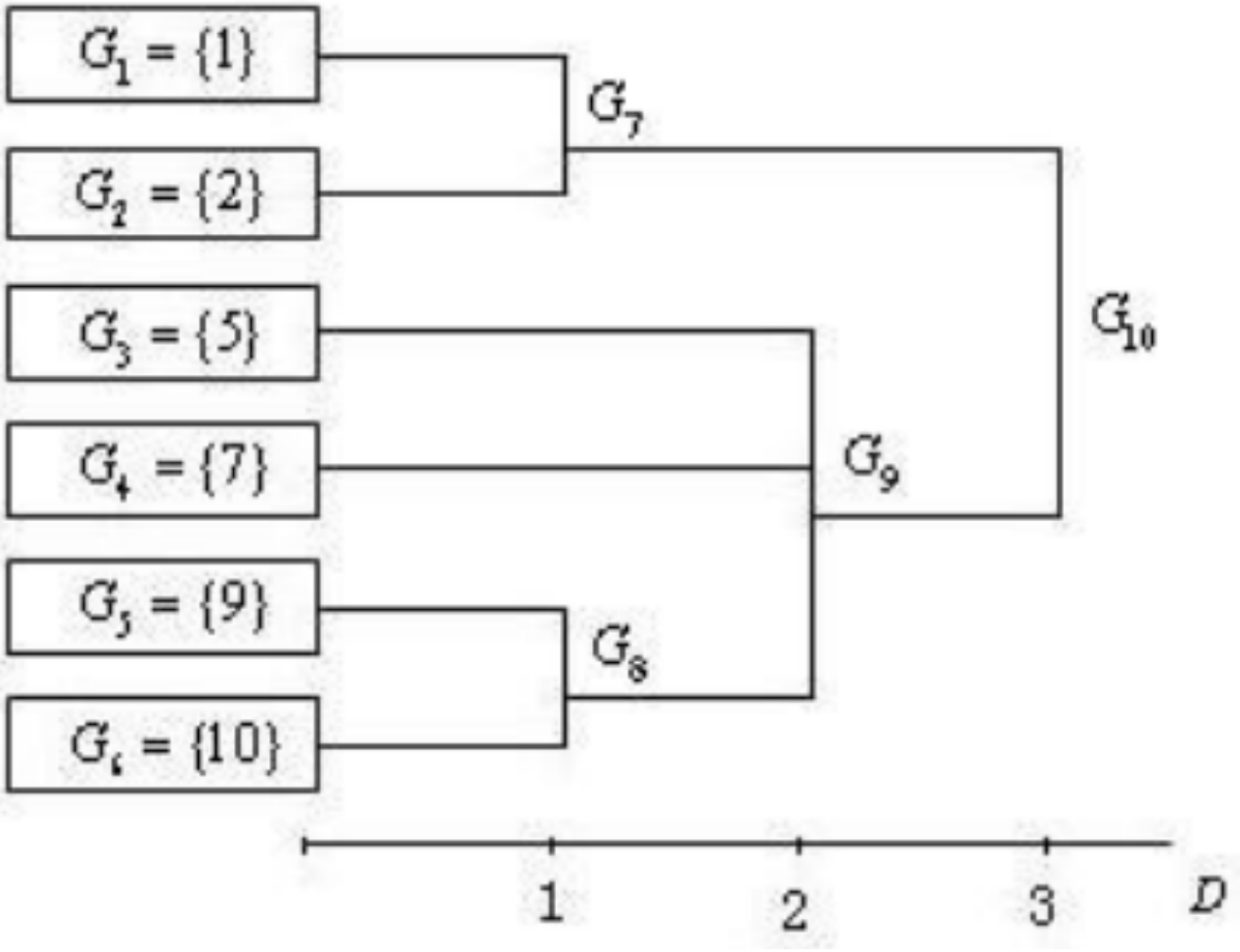
	G_1	G_2	G_3	G_4	G_5	G_6
G_1	0					
G_2	1	0				
G_3	4	3	0			
G_4	6	5	2	0		
G_5	8	7	4	2	0	
G_6	9	8	5	3	1	0

Hierarchical Clustering

	G_7	G_3	G_4	G_8
G_7	0			
G_3	3	0		
G_4	5	2	0	
G_8	7	4	2	0

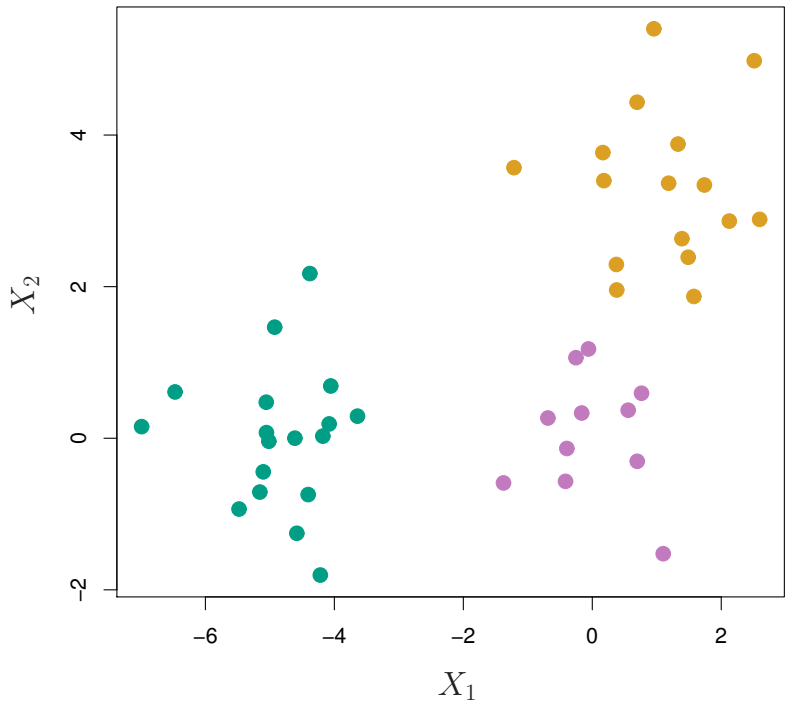
	G_7	G_9
G_7	0	
G_9	3	0

Hierarchical Clustering



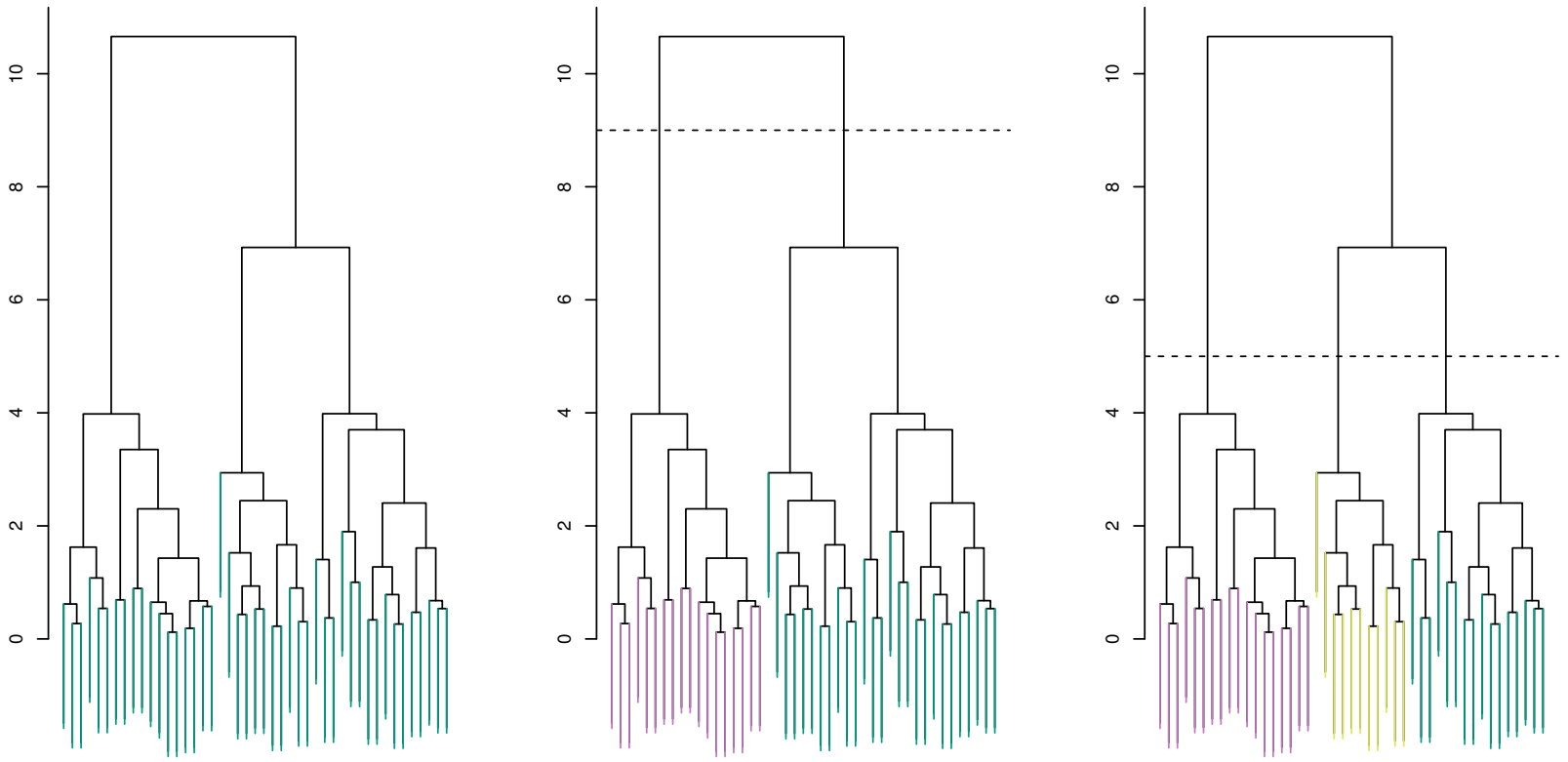
An Example

Excercise
Simulation:
K-means
Hirachical



45 observations generated in 2-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.

Application of hierarchical clustering

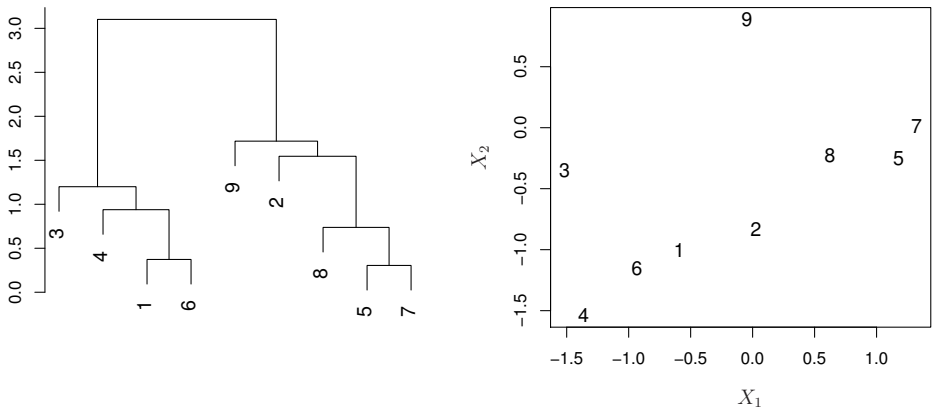


不同划分，得到不同的cluster结果

Details of previous figure

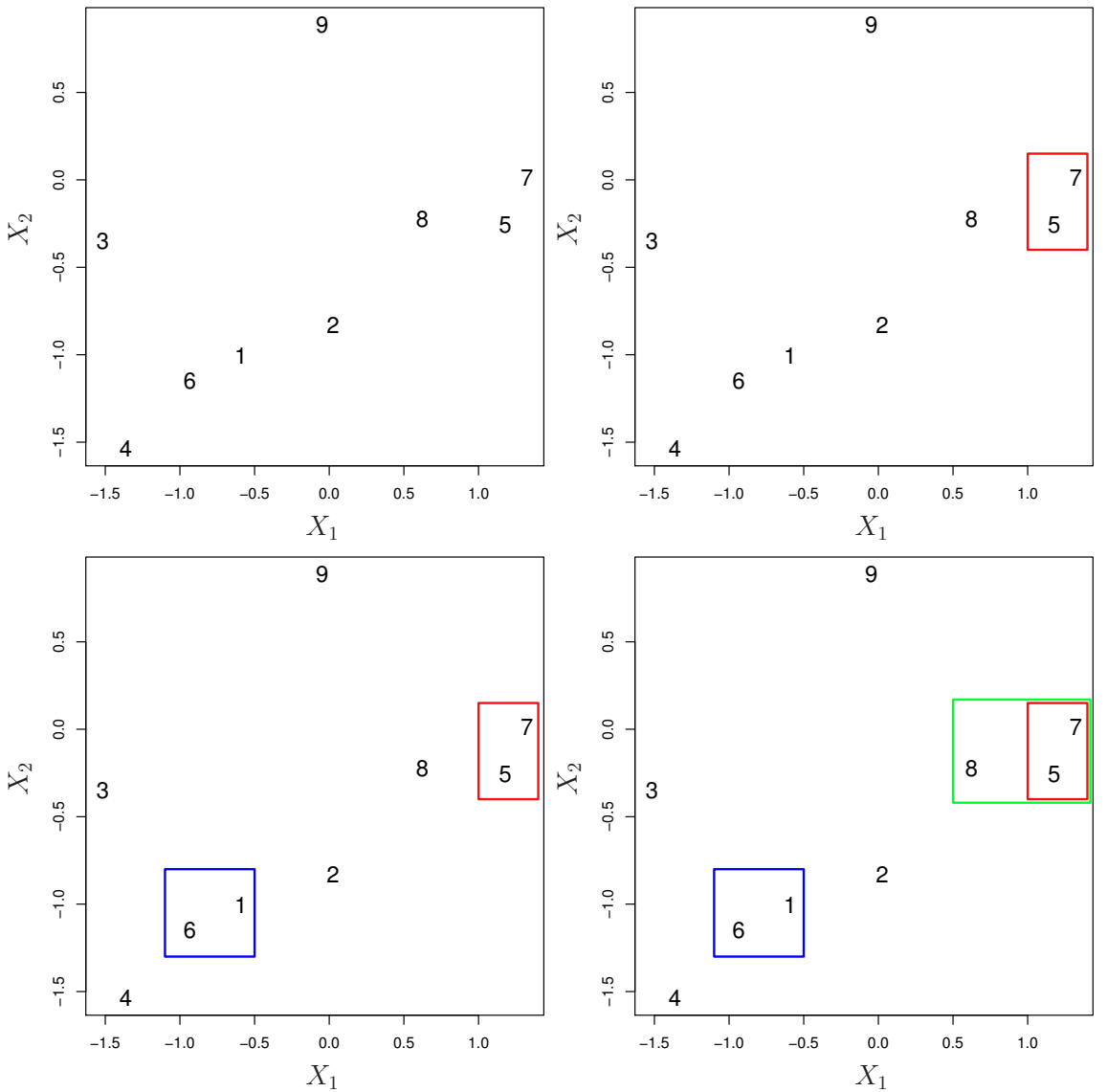
- *Left:* Dendrogram obtained from hierarchically clustering the data from previous slide, with complete linkage and Euclidean distance.
- *Center:* The dendrogram from the left-hand panel, cut at a height of 9 (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.
- *Right:* The dendrogram from the left-hand panel, now cut at a height of 5. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure

Another Example



- An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space. The raw data on the right was used to generate the dendrogram on the left.
- Observations 5 and 7 are quite similar to each other, as are observations 1 and 6.
- However, observation 9 is *no more similar to* observation 2 than it is to observations $8, 5$, and 7, even though observations 9 and 2 are close together in terms of horizontal distance.
- This is because observations $2, 8, 5$, and 7 all fuse with observation 9 at the same height, approximately 1.8.

Merges in previous example



Hierachical Clustering

Algorithm 10.2 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Hierachical Clustering

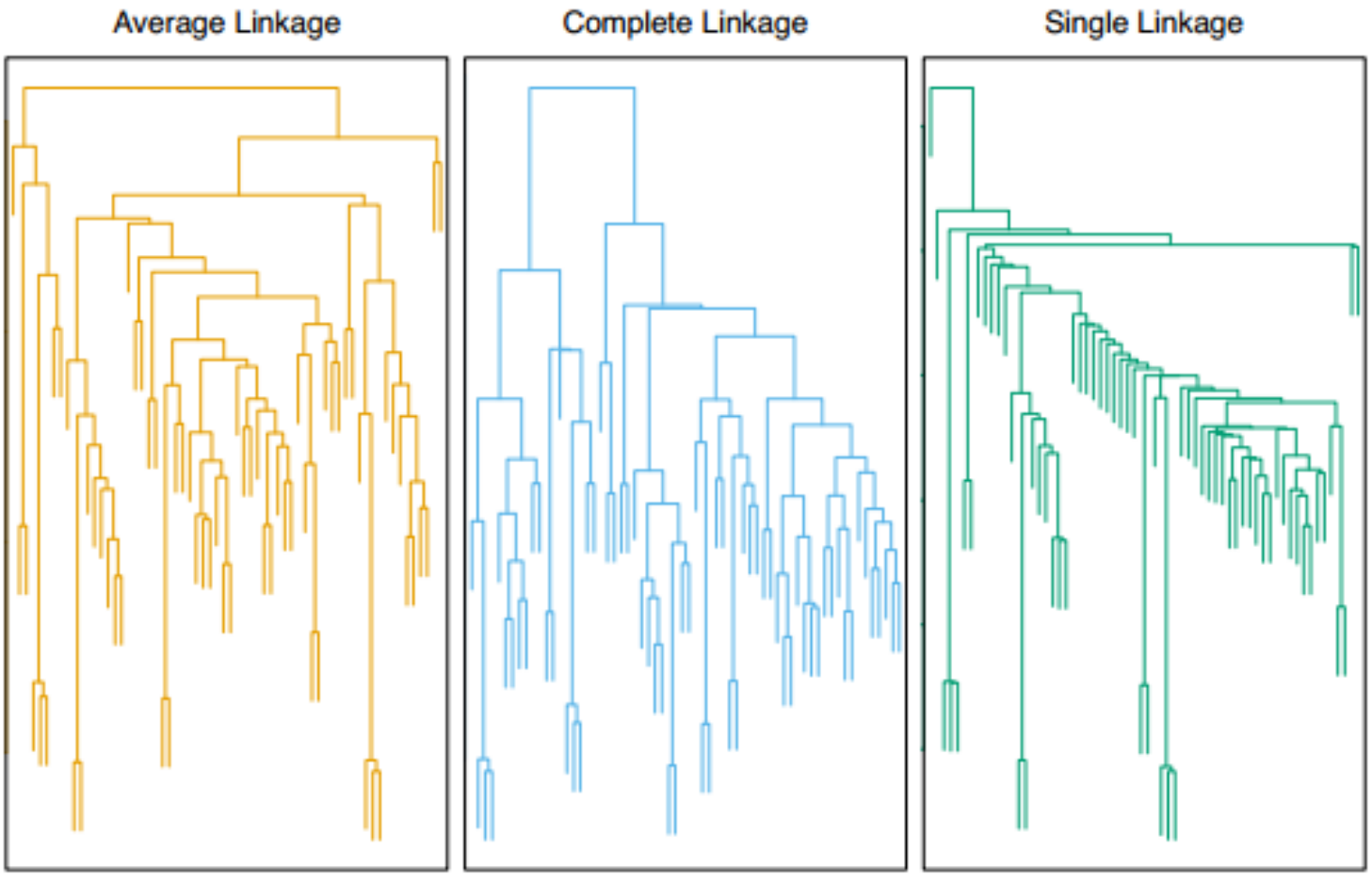


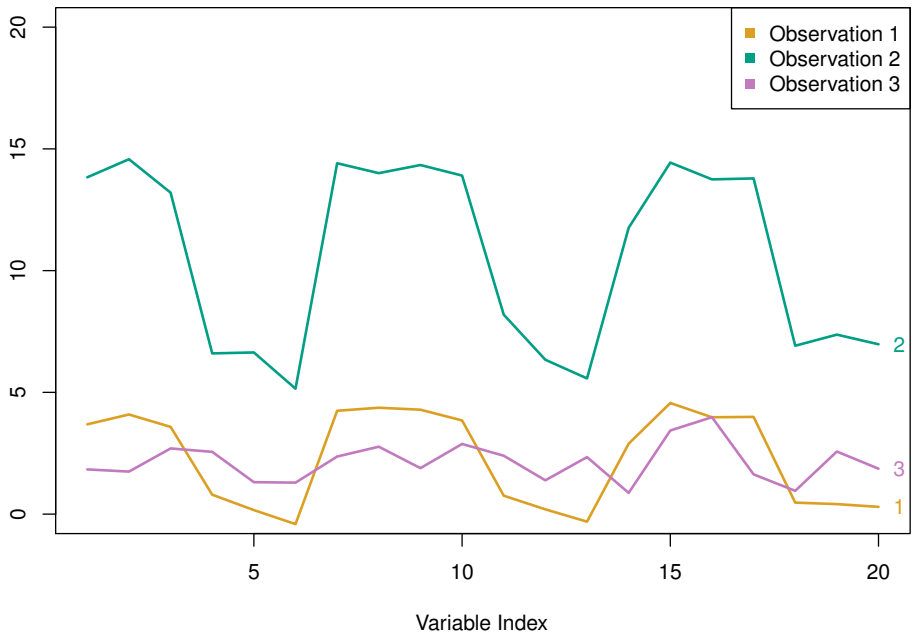
FIGURE 10.12. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

Choice of Dissimilarity Measure

- So far have used Euclidean distance.
- An alternative is *correlation-based distance* which considers two observations to be similar if their features are highly correlated.
- This is an unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations.

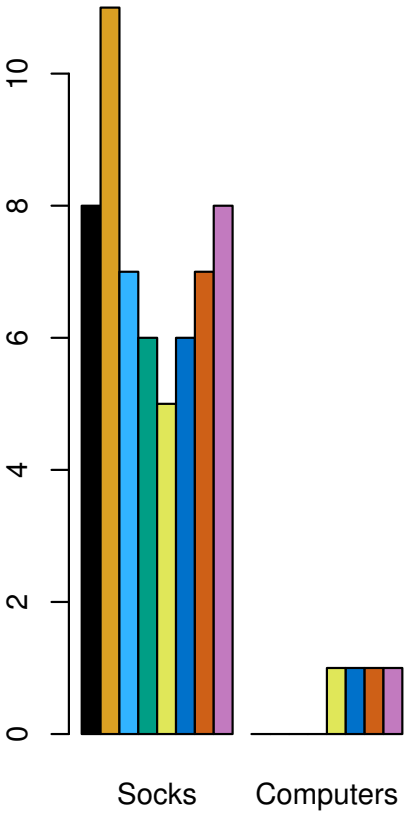
1，3欧式距离小
1，2的相关系数小

零售店客户细分
距离
相关系数?

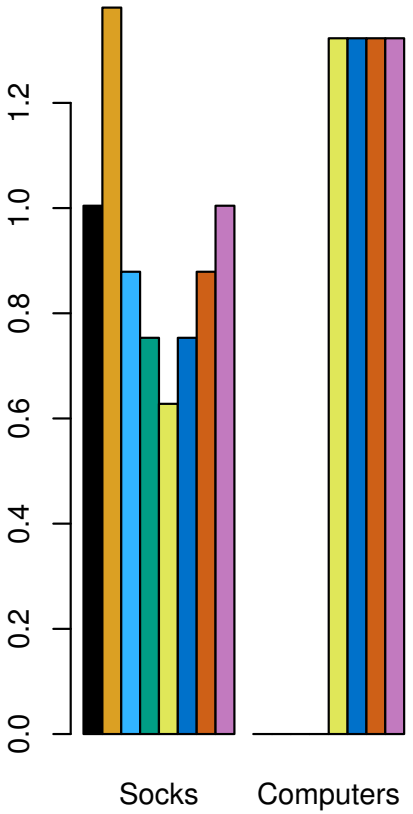


Scaling of the variables matters

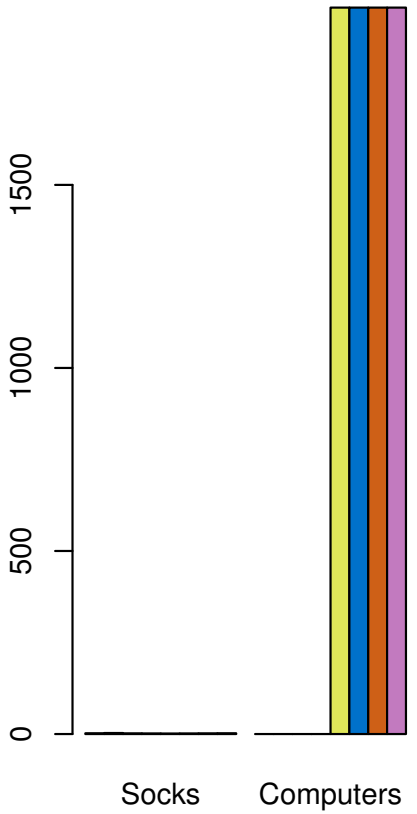
购买数量



购买数量标准化



购买金额



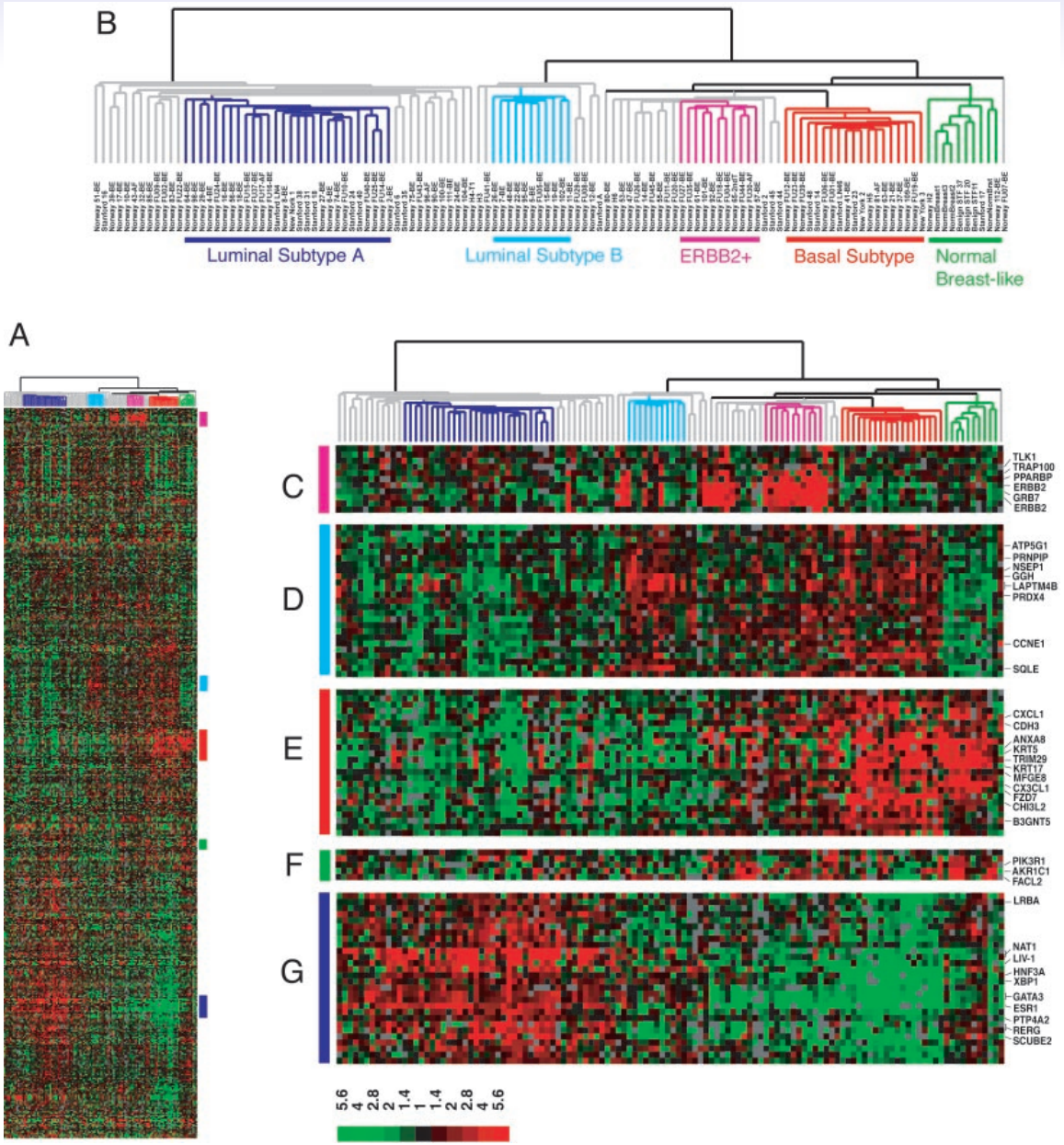
sock购买频率高于computers
是否需要标准化？标准化后变量的influence一样

Practical issues

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
- How many clusters to choose? (in both K-means or hierarchical clustering). Difficult problem. No agreed-upon method. See Elements of Statistical Learning, chapter 13 for more details.

Example: breast cancer microarray study

- "Repeated observation of breast tumor subtypes in independent gene expression data sets;" Sorlie at el, PNAS 2003
- Average linkage, correlation metric
- Clustered samples using 500 *intrinsic genes:* each woman was measured before and after chemotherapy. Intrinsic genes have smallest within/between variation.


B
Luminal Subtype A
Luminal Subtype B
ERBB2+
Basal Subtype
Normal Breast-like
A
C
D
E
F
G
TLK1
TRAP100
PPARBP
ERBB2
GRB7
ERBB2
ATP5G1
PRNPIP
NSEP1
GGH
LAPTM4B
PRDX4
CCNE1
SQLE
CXCL1
CDH3
ANXA8
KRT5
TRIM29
KRT17
MFGE8
CX3CL1
FZD7
CHI3L2
B3GNT5
PIK3R1
AKR1C1
FACL2
LRBA
NAT1
LIV-1
HNF3A
XBP1
GATA3
ESR1
PTP4A2
RERG
SCUBE2
5.6
4
2.8
2
1.4
1
1.4
2
2.8
4
5.6

Biclustering

Two types of traditional clustering methods

- Clustering samples based on all variables
- Clustering variables based on all samples

The underlying assumption

- All variables may give contribution to the classification of samples
- But there exist one problem. If only a fraction of all samples may perform similarly in a fraction of all variables?

Definition of Biclustering

In many reality datasets, we may face the problems above. Thus, we introduce Biclustering to deal with such problems.

- Biclustering is a method to cluster samples and variables simultaneously and we can get a cluster corresponding to some samples and some variables.
- This method extracts a duality between samples and variables, which can improve the accuracy of the clustering results and enhance the interpretability of the clustering results.

The Difference between Biclustering and Clustering

Here we explain the differences based on the genetic data.

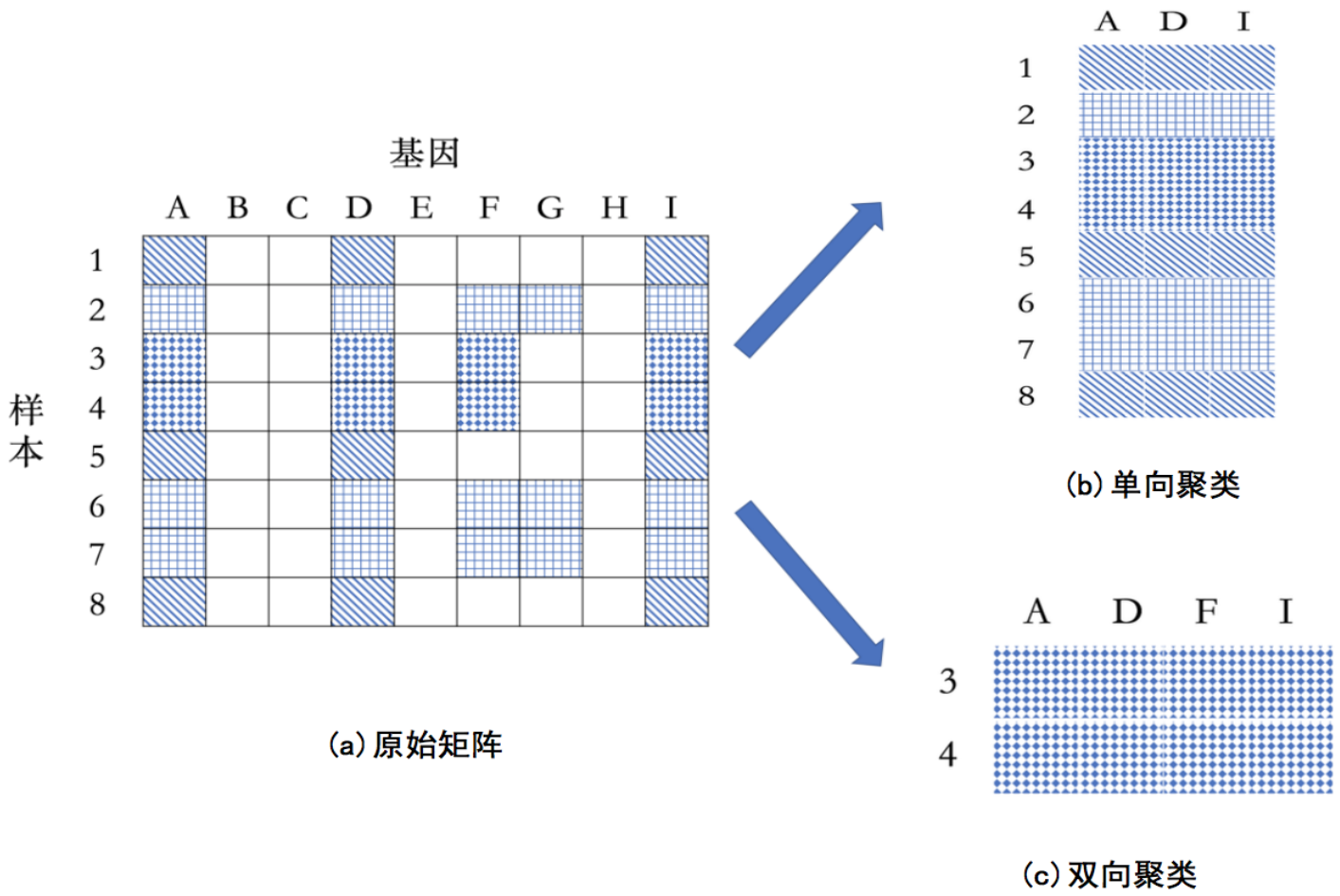


Figure 1: The Differences Between Clustering and Biclustering

The Application fileds of Biclustering

- Though many scholars proposed Biclustering mthods to make genetic analysis in the beginning, Biclustering have been applied to many fields latter, such as text clustering, collaborative filtering and so on.
- Here we take genetic analysis and text clustering as two examples to illustrate the advantages of Biclustering.

The Advantages of Biclustering in Genetic Analysis

- In one hand, one specific disease may be affected by only a few genes, not all genes. The direct clustering methods based on all genes may ignore this factor.
- In another hand, different subtypes of one specific disease may be affected by different groups of genes, but we don't know which genes the different groups may contain before.
- Consequently, Biclustering can find some patients are similar in some genes and extract this local relationship.

The Advantages of Biclustering in Text Clustering

- Text matrix is a sparse and high-dimensional matrix.
- In one hand, The computed distances between samples in sparse matrix may often be small, it's hard to cluster documents based on similarity index.
- In another hand, the documents that belong to one theme often have similarity only in a part of words.
- Consequently, Biclustering can find the duality between some documents and some words and we can infer the specific theme based on the main words.

The Division of Biclustering Methods

In fact, Biclustering methods can be roughly divided two types based on whether we should find the specific structure of submatrix.

- Structural Biclustering means that there exist a specific structure in the submatrix. We usually want to find the specific structures of datasets in genetic analysis.
- Non-structural Biclustering means that there exist no specific structure in the submatrix. We usually depend the similarity between samples and variables to find the duality in text clustering and collaborative filtering.

The Structure of Biclustering Methods

- Let $\mathcal{B}$ be a bicluster consisting of a set I of n genes and a set J of m samples, in which a_{ij} denotes the gene i under the sample j.

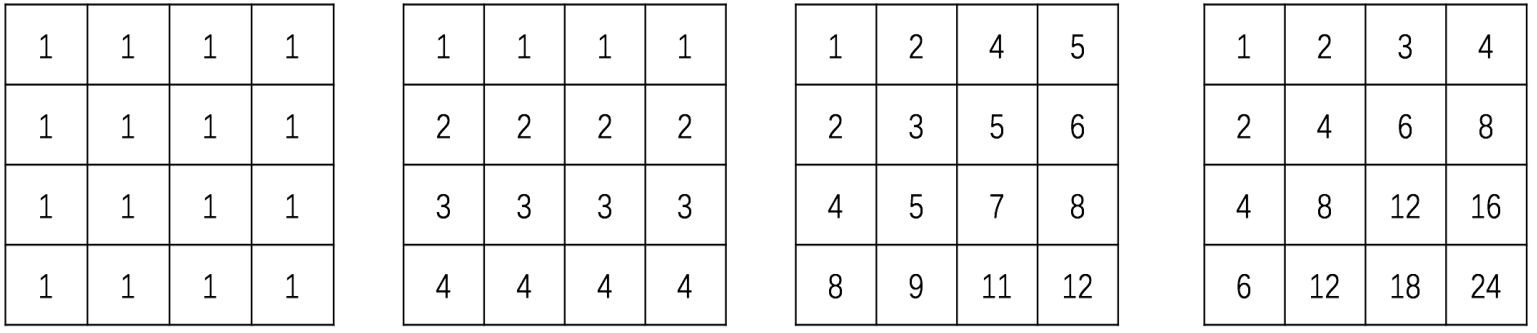
	sample 1	...	sample j	...	sample m
Gene 1	a_{11}	...	a_{1j}	...	a_{1m}
Gene ...	...	...	...	...	...
Gene i	a_{i1}	...	a_{ij}	...	a_{im}
Gene ...	...	...	...	...	...
Gene n	a_{n1}	...	a_{nj}	...	a_{nm}

Figure 2: The Structure of Gene Data

The Structure of Biclustering

We can divide the specific structural submatrix into three types based on different structures.

- Constant values
- Constant values on rows or columns
- Coherent values on both rows and columns



1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

(a) 常数类

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

(b) 行常数类

1	2	4	5
2	3	5	6
4	5	7	8
8	9	11	12

(c) 协同效应类-累加模型

1	2	3	4
2	4	6	8
4	8	12	16
6	12	18	24

(d) 协同效应类-累乘模型

Figure 3: The Difference Structures of Submatrix

Two Biclustering Methods

Here we mainly introduce two methods in Biclustering, including Sparse Biclustering and Spectral Biclustering.

- Sparse Biclustering is a method to obtain sparse clusters by means of penalized objective function..
- Spectral Biclustering is the extension of Spectral Clustering, based on the bipartite graphs.
- Sparse Biclustering methods aims to finding the specific structures of gene datasets, thus it belongs to structural biclustering.
- Spectral Biclustering belongs to non-structural biclustering if we apply it to text clustering, while it can also belong to structural biclustering if we apply it to genetic analysis.

Sparse Biclustering

- The basis objective function is as follows

$$\text{argmin} \quad Loss + \lambda \times Penalty$$

- $Loss$ denotes the loss function and we want the loss function to become small enough.
- $Penalty$ denotes the penalty term, which results to some sparse parameters.
- In general, By inducing the penalty term, some unimportant parameters are penalized to be zero, and important parameters keep non-zero. Thus the objective function reflects the trade-off between $Loss$ and $Penalty$, the larger λ, the stronger penalty effects.

Sparse Singular Value Decomposition; SSVD

- Given a gene matrix $X_{n\times d}$, the rows of X denotes samples and the columns of X denotes genes.
- The singular value decomposition of X is as follows

$$X = UDV^\top = \sum_{k=1}^{r} s_k u_k v_k^\top$$

- r denotes the rank of X.
- $U = (u_1, \cdots, u_r)$ denotes a matrix consists of left orthogonal singular vectors.
- $V = (v_1, \cdots, v_r)$ denotes a matrix consists of right orthogonal singular vectors.
- $D = diag(s_1, \cdots, s_r)$ is a diagonal matrix and the elements in the diagonal line are non-zero singular values $(s_1, \cdots, s_r)$.

SSVD

- Though the SVD can decompose a matrix into the sum of r rank-1 matrices, here we just focus on the first K largest matrices, these matrices are called the first K layers:

$$X \approx X^{(K)} = \sum_{k=1}^{K} s_k u_k v_k^\top$$

- In the SSVD algorithm, authors extract the rank-1 matrix layer by layer. Thus we focus on how to extract the first layer in later parts, and we can let X minus the first layer as the basic matrix for extracting the second layer.
- In fact, if we use no penalty, the first layer is the matrix respect to the largest singular value but we can not get sparse layer.
- Note that, we can get a series submatrixes of coherent values by multiplication.

SSVD

- Here we take penalties on the left and right singular vectors to make the unimportant elements to be zero
- The restored matrix respect to first layer may only have some elements which are non-zero.
- After realignment to the restored matrix, we can get the first bicluster.
- Therefore the final objective function is as follows:

$$\text{argmin} \quad ||X - suv^{\top}||_F^2 + s\lambda_u \sum_{i=1}^{n} w_{1,i}|u_i| + s\lambda_v \sum_{j=1}^{d} w_{2,j}|v_j|$$

- we use adaptive lasso penalty and w_1 and w_2 are weights.
- We can fix u to optimize v and fix v to optimize x until convergence.

SSVD with Lung Cancer Data

Note that, there exist sparse elements in each layer matrix and we can find positive or negative effects of different groups of genes corresponding different subtypes of lung cancer.

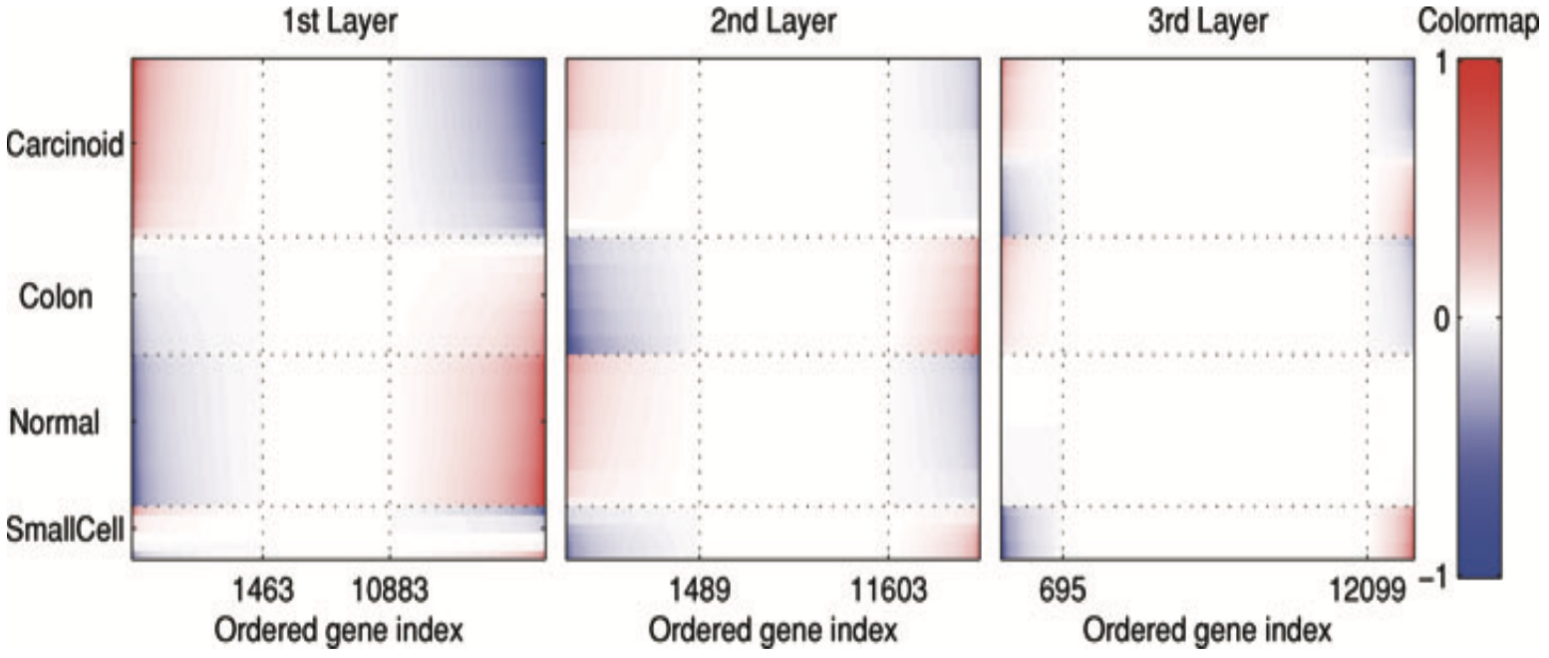


Figure 4: SSVD with Lung Cancer Data

Sparse Two-way K-Means; STKM

- Based on the idea of Sparse K-Means, we can extend it to Biclustering.
- Given matrix $X_{n\times p}$, the rows of X denotes samples and the columns of X denotes variables.
- Suppose the n samples belong to K uncross clusters $(C_1, \cdots, C_K)$, the p variables belong to R uncross clusters $(D_1, \cdots, D_R)$.
- Suppose the elements in X are independent and $X_{ij} \sim N(\mu_{kr}, \sigma^2)$

STKM

- The objective function is as follows

$$\text{argmin} \quad \{\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{i\in C_k}\sum_{j\in D_r}(X_{ij}-\mu_{kr})^2\}$$

- We can find that if $R = p$, the problems turn to divide samples into K clusters by K-Means. If $K = n$, the problems turn to divide variables into R clusters by K-Means.
- Thus we can get $K \times R$ clusters finally, each of which consists of samples in C_k and variables in D_r.
- Note that, we can get some submatrixes of constant values.

STKM

- Under such objective function, we may get some clusters, the means of which are near the overall means μ. But we want to get some clusters different from the overall means.
- We can centralization the elements in X to make the means of all elements to be zero.
- To improve the interpretability of the results and reduce the variance between clusters, we should shrinkage the means within clusters which is near zero to be exact zero.
- Finally we induce the following objective function:

$$\text{argmin} \quad \{\frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{i\in C_k}\sum_{j\in D_r}(X_{ij}-\mu_{kr})^2+\lambda\sum_{k=1}^{K}\sum_{r=1}^{R}|\mu_{kr}|\}$$

STKM with Simulated Data

Note that, suppose the hidden submatrix are filled with constant values, we can use STKM to find more accurate matrix structure than independent K-means clustering.

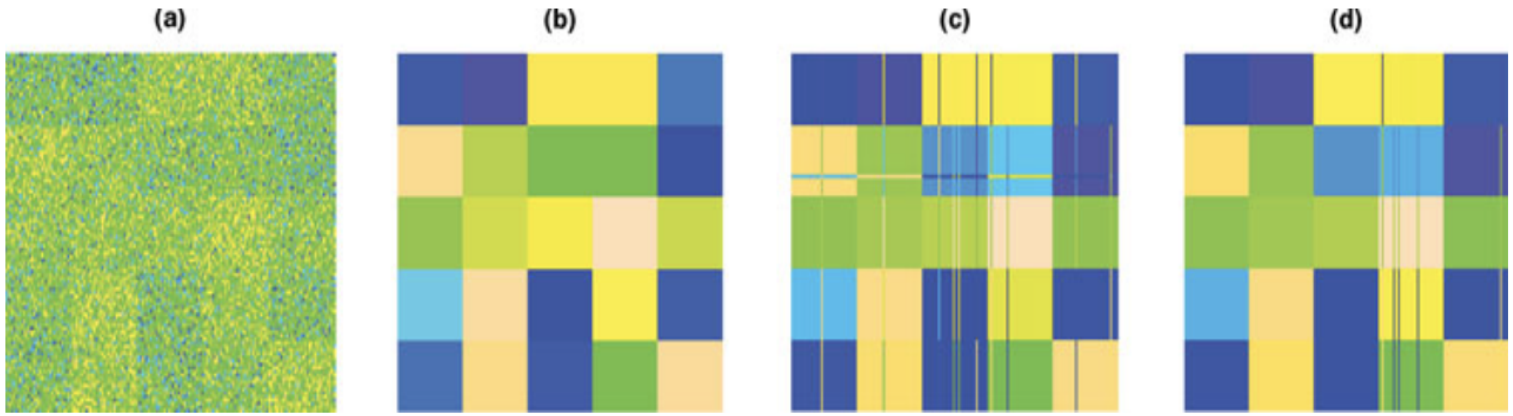


Figure 5: STKM with Simulated Data

STKM with Lung Cancer Data

The following estimated mean matrix is similar to the three image plots obtained using SSVD in lung cancer data aforehand .

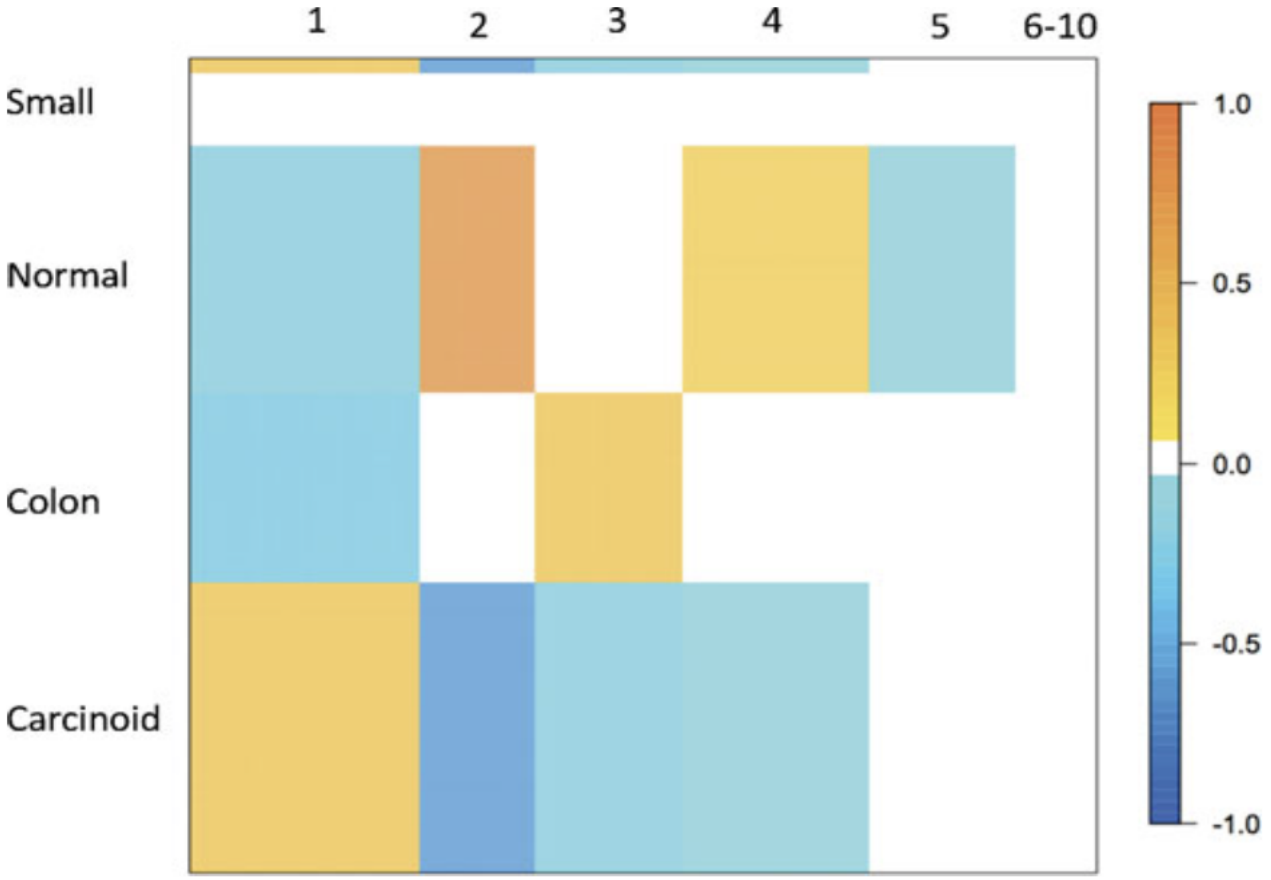


Figure 6: STKM with Lung Cancer Data

Convex Clustering; CC

- K-means clustering and Hierarchical clustering are two widely used clustering methods.
- Due to the instability of these two methods, Lindsten et al. (2011) and Hocking et al. (2011) used a convex penalty, such as L_1 norm, to replace the hidden L_0 norm in these two methods.
- Because of the convex relaxation for these two methods, we call such new clustering method, convex clustering.

Convex Clustering; CC

- Chi and Lange (2015) consider the following objective function for convex clustering.

$$\underset{\mathbf{A}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{X}-\mathbf{A}||_2^2 + \gamma \sum_{i<j} w_{i,j}||\mathbf{A}_{i\cdot} - \mathbf{A}_{j\cdot}||_q$$

- where $\mathbf{A}$ is the approximated matrix which consists of constant structures. $\mathbf{A}_{i\cdot}$ denotes the ith row of $\mathbf{A}$ and $||\cdot||_q$ is the L_q norm of a vector. $w_{i,j}$ is the non-negative weight between the ith row and the jth row.

Sparse Convex Clustering; SCC

- Wang et al. (2018) add a group sparsity-induced penalty to take variable selection and get clustering results simultaneously so that we can deal with high-dimensional data.

$$\underset{\mathbf{A}}{\operatorname{argmin}} \frac{1}{2}\sum_{j=1}^{p}||\mathbf{x}_j-\mathbf{a}_j||_2^2+\gamma_1\sum_{i<j}w_{i,j}||\mathbf{A}_{i\cdot}-\mathbf{A}_{j\cdot}||_2+\gamma_2\sum_{j=1}^{p}u_j||\mathbf{a}_j||_2$$

- where u_j is the non-negative weight on $\mathbf{a}_j$ and γ_2 is a tuning parameter to take variable selection.
- We can get important variables by shrinking the unimportant variables by the second penalty.

SCC with Simulated Data

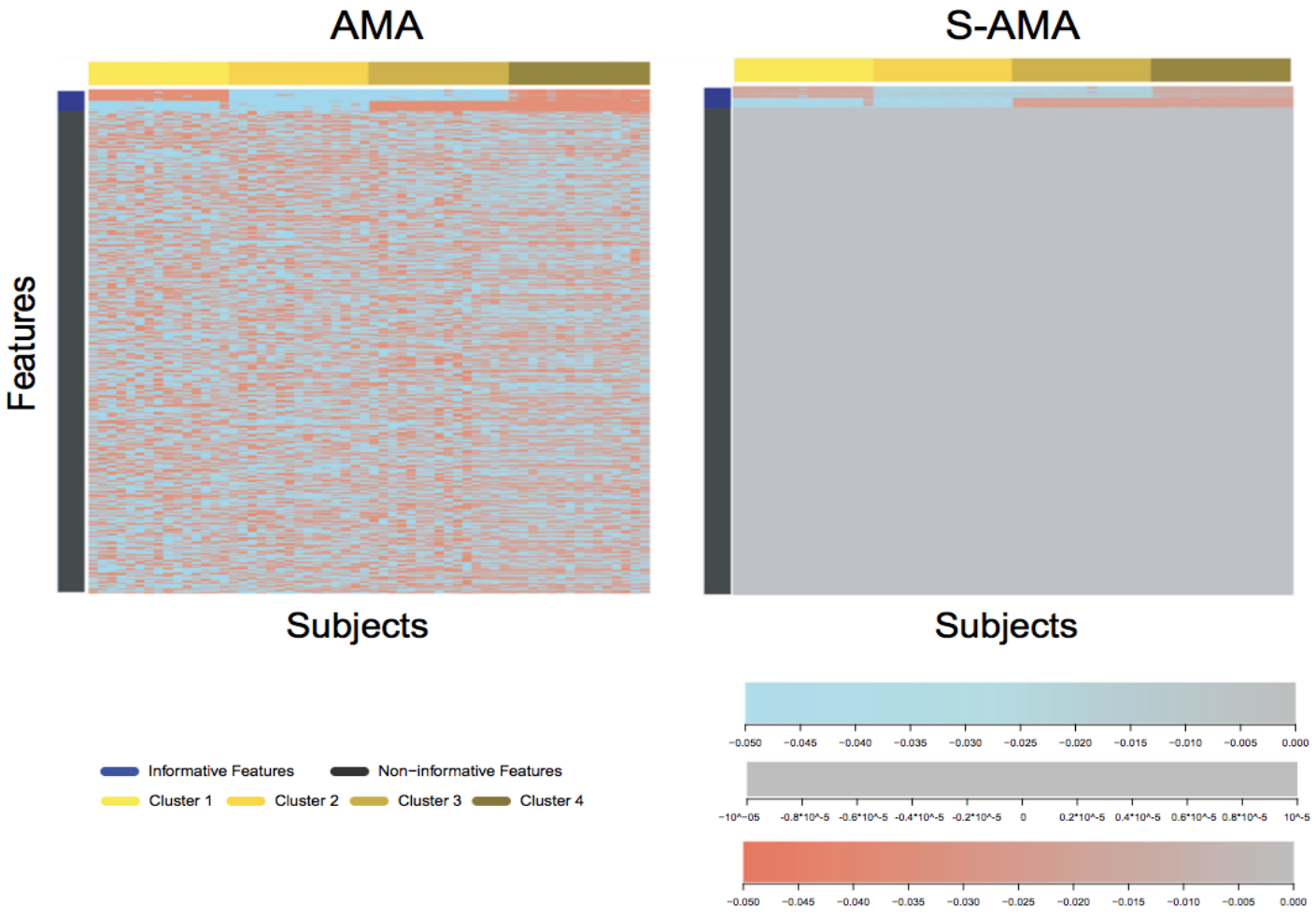


Figure 7: SCC with Simulated Data

Convex Biclustering; CBC

- The convex clustering can be extended to convex biclustering by adding the similar penalty on the pair-wise columns of the matrix.
- Different from SSVD, we want to penalize the difference value between different row vectors and the difference value between different column vectors simultaneously.
- Because we consider the simultaneous penalty on the rows and columns difference, we can finally get $K \times R$ constant submatrix.

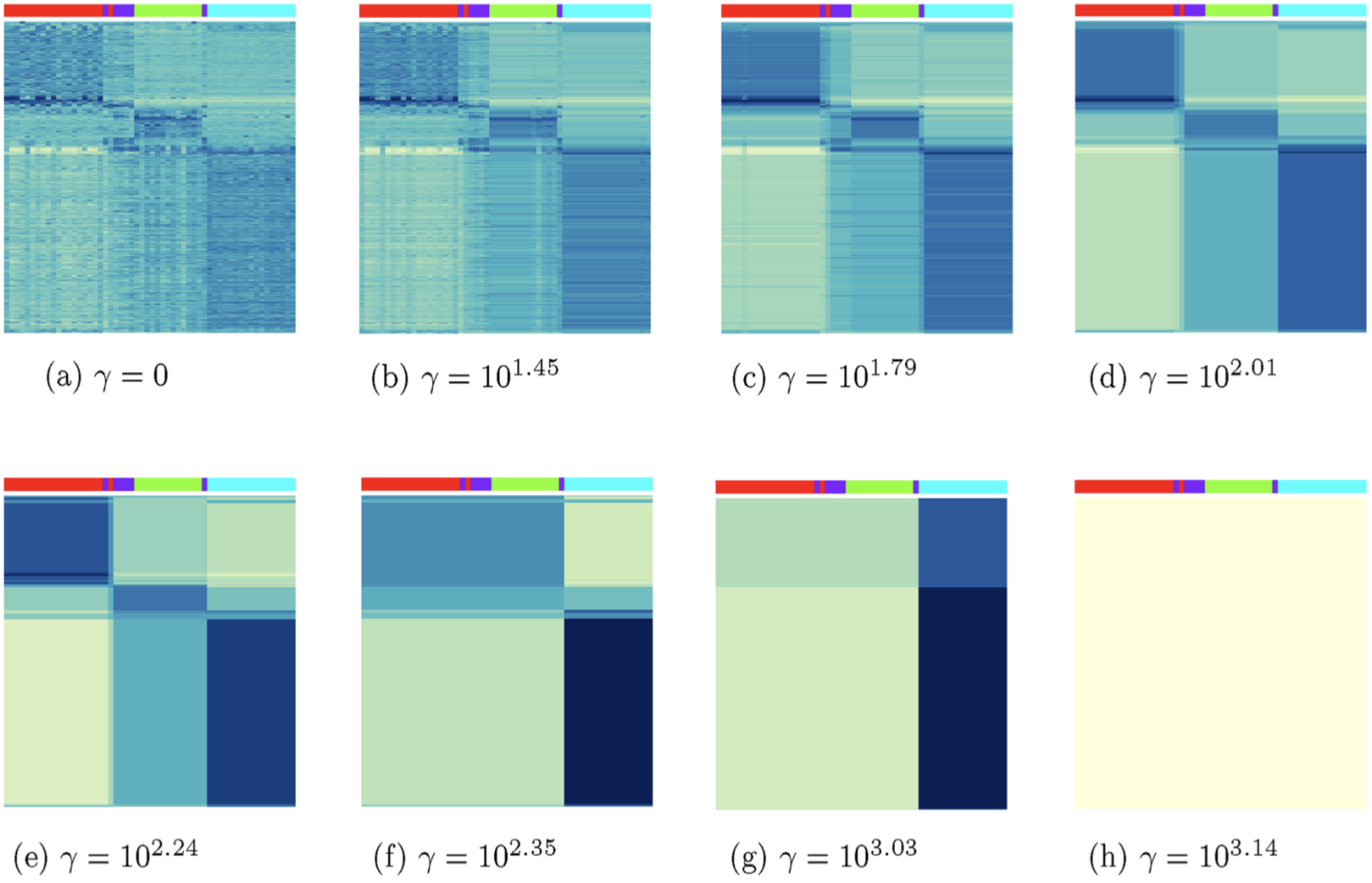
Convex Biclustering; CBC

- The detail objective function is as follows

$$F_\gamma(U) = \frac{1}{2}||X - U||_F^2 + \gamma[\Omega_W(U) + \Omega_W(U^\top)]$$

- Where $\Omega_W(U) = \sum_{i \leq j} w_{ij}||U_{.i} - U_{.j}||_2$, w_{ij} denotes the weights and $U_{.i}(U_{i.})$ denotes the ith column(row).
- If we delete one penalty from above two penalties, we may get a Convex Clustering problem.
- Authors propose to use Sparse Gaussian Kernel Weights as weights and use DLPA algorithm to turn the Convex Biclustering into Convex Clustering. Then we can use ADMM or AMA algorithm to get the final clusters.
- Note that, we can get some submatrixes of constant values.

CBC with Lung Cancer Data



(a) $\gamma = 0$ (b) $\gamma = 10^{1.45}$ (c) $\gamma = 10^{1.79}$ (d) $\gamma = 10^{2.01}$

(e) $\gamma = 10^{2.24}$ (f) $\gamma = 10^{2.35}$ (g) $\gamma = 10^{3.03}$ (h) $\gamma = 10^{3.14}$

Figure 8: CBC with Lung Cancer Data

Spectral Clustering

- At first, we introduce spectral clustering to help readers understand spectral biclustering in the latter content.
- Spectral Clustering is a widely used clustering method, which regards a clustering problem as a graph partition problem.
- Consider a similarity graph $G = (V, E)$. Each vertex v_i in this graph represents a data point x_i.
- Two vertices are connected if the similarity s_{ij} between the corresponding data points x_i and x_j is positive or larger than a certain threshold, and the edge is weighted by s_{ij}.

Spectral Clustering

- We want to find a partition of the graph such that the edges between different groups have very low weights (which means that points in different clusters are dissimilar from each other)
- The edges within a group have high weights (which means that points within the same cluster are similar to each other).

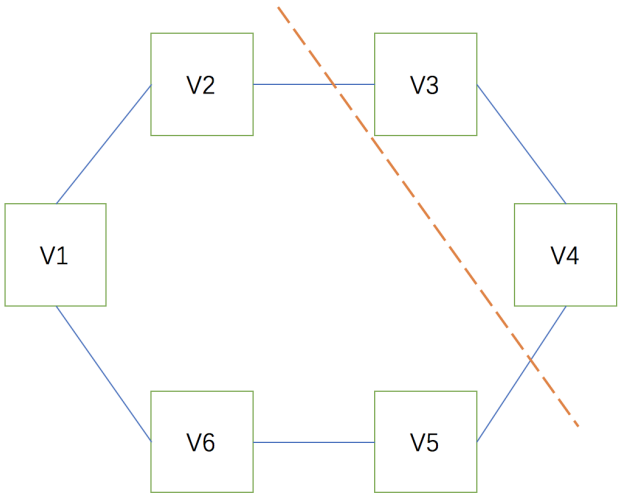


Figure 9: Graph Cut of Spectral Clustering

Some definitions in Spectral Clustering

- Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, \cdots, v_n\}$.
- We assume that the graph G is weighted, that is each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geq 0$.
- The weighted adjacency matrix of the graph is the matrix $W = (w_{ij})_{i,j=1\cdots,n}$. If $w_{ij} = 0$, this means that the vertices v_i and v_j are not connected by an edge.
- As G is undirected we require $w_{ij} = w_{ji}$. The degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- The degree matrix D is defined as the diagonal matrix with the degrees $d_1, \cdots, d_n$ on the diagonal.

Some definitions in Spectral Clustering

- Given a subset of vertices $A \subset U$, we denote its complement $V \setminus A$ by $\bar{A}$.
- We define the indicator vector $\mathbb{1}_A = (f_1, \cdots, f_n)^\top \in \mathbb{R}^n$ as the vector with entries $f_i = 1$ if $v_i \in A$ and $f_i = 0$ otherwise. For convenience, we introduce the shorthand notation $i \in A$ for the set of indices $\{i | v_i \in A\}$.
- For two not necessarily disjoint sets $A, B \subset V$, we define

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- Here we consider a way of measuring the "size" of a subset $A \subset V$:

$$vol(A) = \sum_{i \in A} d_i$$

Different similarity graphs in Spectral Clustering

- There are several popular constructions to transform a given set $x_1, \cdots, x_n$ of data points with pairwise similarities s_{ij} or pairwise distances d_{ij} into a graph.
- When constructing similarity graphs the goal is to model the local neighborhood relationships between the data points.
- The regularly used similarity graphs in spectral clustering are **The ϵ-neighborhood graph**, **k-nearest neighbor graph** and **The fully connected graph**.
- Here we mainly introduce **The fully connected graph**.

The fully connected graph in Spectral Clustering

- Here we simply connect all points with positive similarity with each other, and we weight all edges by s_{ij}.
- As the graph should represent the local neighborhood relationships, this construction is only useful if the similarity function itself models local neighborhoods.
- An example for such a similarity function is the Gaussian similarity function

$$s(x_i, x_j) = \exp(-||x_i - x_j||^2/(2\sigma^2))$$

where σ controls the width of the neighborhoods.

Graph cut point of view

- Given a similarity graph with adjacency matrix W, the simplest and most direct way to construct a partition of the graph is to solve the mincut problem.
- For a given number k of subsets, the mincut approach simply consists in choosing a partition $A_1, \cdots, A_k$, which minimizes

$$\text{cut}(A_1, \cdots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

- However, in practice it often does not lead to satisfactory partitions.
- Because the solution of mincut simply separates one individual vertex from the rest of the graph.

Graph cut point of view

- One way to circumvent this problem is to explicitly request that the sets $A_1, \cdots, A_k$ are reasonably large.
- The two most common objective functions to encode this are RatioCut (Hagen and Kahng, 1992) and the normalized cut Ncut (Shi and Malik, 2000).
- Here we mainly introduce Ncut

$$\text{Ncut}(A_1, \cdots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}$$

- Note that this objective function try to achieve the "balance" between the minimizing the edge weights between different groups and the size of groups

Graph Laplacian Matrix

- The unnormalized graph Laplacian matrix is defined as

$$L = D - W$$

- The normalized graph Laplacian matrix is defined as

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

- We denote the first matrix by L_{sym} as it is a symmetric matrix, and the second one by L_{rw} as it is closely related to a random walk.

Properties of L_{sym} and L_{rw}

- For every $f \in \mathbb{R}^n$ we have

$$f^\top L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 .$$

- λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{-1/2}u$.
- λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigen-problem $Lu = \lambda Du$
- 0 is an eigenvalue of L_{rw} with the constant one vector $\mathbb{1}$ as eigenvector. 0 is an eigenvalue L_{sym} with eigenvector $D^{1/2}\mathbb{1}$
- L_{sym} and L_{rw} are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$

Approximating Ncut in $k = 2$

- At first, we consider $k = 2$ and define the cluster indicator vector f by

$$f_i = \begin{cases} \sqrt{\frac{\mathrm{vol}(\bar{A})}{\mathrm{vol}(A)}} & \text{if} \quad v_i \in A \\ -\sqrt{\frac{\mathrm{vol}(A)}{\mathrm{vol}(\bar{A})}} & \text{if} \quad v_i \in \bar{A}. \end{cases} \tag{1}$$

- We can check that $(Df)^\top \mathbb{1} = 0$, $f^\top D f = \mathrm{vol}(V)$ and $f^\top L f = \mathrm{vol}(V)\mathrm{Ncut}(A, \bar{A})$
- Thus we can write the problem of minimizing Ncut by the equivalent problem

$$\underset{\mathrm{A}}{\mathrm{argmin}}\ f^\top L f \quad \text{subject to} f \quad \text{as in (1)} \quad Df \perp \mathbb{1} \quad f^\top D f = \mathrm{vol}(V) \tag{2}$$

Approximating Ncut in $k = 2$

- We relax the problem by allowing f to take arbitrary real values

$$\underset{\mathrm{f}\in\mathbb{R}^{\mathrm{n}}}{\operatorname{argmin}}\ f^\top L f \quad \text{subject to} \quad Df \perp \mathbb{1} \quad f^\top D f = \mathrm{vol}(V) \tag{3}$$

- Now we substitute $g = D^{1/2} f$ and the problem can be reformulated as

$$\underset{\mathrm{g}\in\mathbb{R}^{\mathrm{n}}}{\operatorname{argmin}}\ g^\top D^{-1/2} L D^{-1/2} g \quad \text{subject to} \quad g \perp D^{1/2}\mathbb{1}, ||g||^2 = \mathrm{vol}(V) \tag{4}$$

- Note that $D^{-1/2} L D^{-1/2} = L_{sym}$, $\mathrm{vol}(V)$ is a constant and $D^{1/2}\mathbb{1}$ is the first eigenvector of L_{sym}.

Approximating Ncut in $k = 2$

- Problem (4) is in the form of the standard Rayleigh-Ritz theorem and its solution g is given by the second eigenvector of L_{sym}.
- Re-substituting $f = D^{-1/2}g$ and using Proposition we described before, we can find that f is the second eigenvector of L_{rw}, or equivalently the generalized eigenvector of $Lu = \lambda Du$.

Approximating Ncut in $k > 2$

- For the case of finding $k \geq 2$ clusters, we define the indictor vectors $h_j(h_{1,j}, \cdots, h_{n,j})^\top$ by

$$h_{i,j} = \begin{cases} 1/\sqrt{(vol)(A_j)} & \text{if } v_i \in A_j \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

- Then we set the matrix H as the matrix containing those k indicator vectors as columns.
- Observe that $H^\top H = I$, ${h_i}^\top D h_i = 1$ and ${h_i}^\top L h_i = \text{cut}(A_i, \bar{A}_i)/\text{vol}(A_i)$.
- We can reformulate the problem of minimizing Ncut as

$$\underset{\mathrm{A_1},\cdots,\mathrm{A_k}}{\operatorname{argmin}} \ \text{Tr}(H^\top L H) \text{ subject to } H^\top D H = I, H \text{ as in} (5) \quad (6)$$

Approximating Ncut in $k > 2$

- Relaxing the discreteness condition and substituting $T = D^{1/2}H$, we can obtain the relaxed problem

$$\underset{\mathrm{T}\in\mathbb{R}^{\mathrm{n}\times\mathrm{k}}}{\operatorname{argmin}} \operatorname{Tr}(T^\top D^{-1/2} L D^{-1/2} T) \text{ subject to } T^\top T = I \qquad (7)$$

- This is the standard trace minimization problem which is solved by the matrix T which contains the first k eigenvectors of L_{sym} as columns.
- Re-substituting $H = D^{-1/2}T$ and using the Proposition, we see that the solution H consists of the first k eigenvectors of the matrix L_{rw} or the first k generalized eigenvectors of $Lu = \lambda Du$
- This yields the normalized spectral clustering algorithm according to Shi and Malik (2000).

Spectral Clustering Algorithm

Normalized spectral clustering according to Shi and Malik (2000)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- Compute the first k generalized eigenvectors $u_1, \ldots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Figure 10: Normalized Spectral Clustering Algorithm

Spectral Biclustering

- Dhillon (2001) firstly extended Spectral Clustering to Spectral Biclustering on document clustering problem.
- Spectral Biclustering focuses on the similarity between samples and variables.

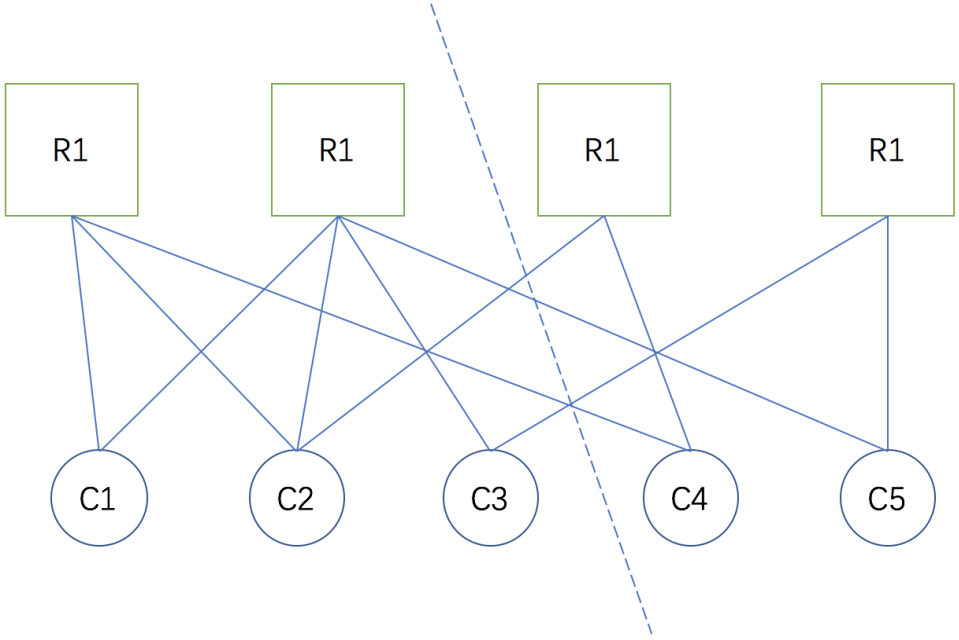


Figure 11: Graph Cut of Spectral Biclustering

Spectral Biclustering

- Consider a matrix $A_{n\times m}$ to construct a bipartite graph G.
- The row vertices $(R = 1, \cdots, n)$ denote the rows and the column vertices $(C = 1, \cdots, m)$ denote the columns.
- Only the row vertices can link with the column vertices so that there exist no link with row vertices or column vertices.
- Different from the Spectral Clustering, we want to extract the local relationship between samples and variables, thus we can design such a bipartite graph.

Laplacian Matrix

Here we construct two diagonal matrices, D_1 and D_2. For D_1, the diagonal elements are the sum of the row of matrix A. For D_2, the diagonal elements are the sum of the column of matrix A.

- We can get a new degree matrix and a new adjacent matrix

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

$$W = \begin{bmatrix} 0 & A \\ A^\top & 0 \end{bmatrix}$$

- In the same way, we can get a new Laplacian Matrix

$$L = D - W = \begin{bmatrix} D_1 & -A \\ -A^\top & D_2 \end{bmatrix}$$

The process of Spectral Biclustering

- Here we use Normalized-Cut Objective function, which is corresponding to the above Laplacian Matrix.
- Then we may ignore same technical details and find that the solution to the objective function in binary classification problem are the eigenvectors respect to the second smallest eigenvalue in $Lz = \lambda Dz$
- Based on the theorem, the minimum partition problem turns to find the specific eigenvector respect to $D^{-1}L$

The process of Spectral Biclustering

- Because of the special structure of bipartite graph, we can turn the Eigen-decomposition problem into the singular value decomposition problem of matrix $A_n = D_1^{-1/2} A D_2^{-1/2}$
- Then we can get the left and right singular vectors respect to the second largest singular values, which is u_2 and v_2.
- Finally we can obtain the solution vector, which is

$$z = \begin{bmatrix} D_1^{-1/2} u_2 \\ D_2^{-1/2} v_2 \end{bmatrix}$$

The process of Spectral Biclustering

- When the number of clusters are greater than 2, we can get the solution in the same way

$$Z = \begin{bmatrix} D_1^{-1/2}U \\ D_2^{-1/2}V \end{bmatrix}$$

- Where U and V are left singular matrix and right singular matrix respect to the largest l eigenvalues except for the largest eigenvalue.
- Then we can take K-means to Z and get the k clusters, in which some samples correspond to some variables.

The Unification of Spectral Biclustering

- Note that, we don't need to find a specific structure in this method, because the similarity is enough to extract the relationship between some documents and some words.
- While Kluger (2003) derive the similar Spectral Biclusteirng method based on genetic structure of coherent values.
- Thus we unify this kind of method as Spectral Biclustering and it can apply to both genetic filed and text mining field.

Thank you