Chapter 14: Unsupervised Learning

Christopher Bradbury, Louis Bensard, and Anthony Keys

August 2nd 2018

Recall Supervised Learning

- Supervised learning = Learning with a teacher
- "Student" use the training sample to find $\hat{y}_i$ and "teacher" provide the correct answer (for classification) or the error of the student's answer from y
- We want to determine the properties of $P(Y|X)$
- Many methods discussed in previous chapters help addressing Supervised learning

Unsupervised learning

- We want to determine properties of $P(X)$ without "teacher"
- Dimension of X is often much higher than supervised learning
- When X has low dimension ($p \leq 3$) then the density $P(X)$ can be fully estimated but when the dimensions gets higher, we need to use less accurate global models such as Gaussian mixtures
- Some methods such as Principal Component Analysis, Self-Organizing maps or Principal curves are used to reduce the dimension of X

Unsupervised learning

- For binary data, Association rules use very simplistic rule to describe regions of $P(X)$ that have high density
- Cluster Analysis and mixture modeling attempts to find modes of the density $P(X)$ instead of modeling the whole density
- Since there is no response, there is no way in unsupervised learning to assess the quality of the results. One must deeply understand the methods talked in this chapter to to trust that the result make sense. The effectiveness of some methods is a matter of opinion since they can't be verified.

Summary

1) Association rules
2) Cluster Analysis
3) Self-Organizing Maps
4) Principal Components, Curves and Surfaces
5) Non-negative matrix factorization
6) The Google PageRank Algorithm
7) Examples of Cluster Analysis

Market basket analysis

- We want to know what items are the most bought in the store, but also what items are the most bought together
- Considering the whole set of possible item to buy, the data is transformed into binary, with $z_{ik} = 1$ if item k is bought and $z_{ik} = 0$ if item k is bought
  ($i = 1, ..., N \quad ; k = 1, ..., K \quad ; N =$#observations ; $K =$# different items in the data set).
- We are interested when those values are 1 or when multiple data vector are 1 at the same time

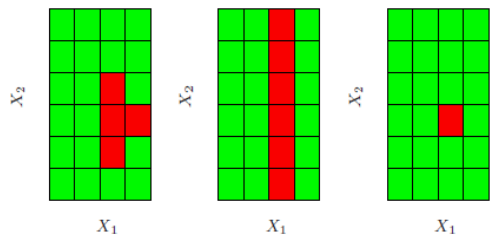


Figure 1: Association Rule (from the book)

- The goal is to find $\Omega \in \{1, ..., K\}$ for which $T(\Omega) = \frac{1}{N} \sum_{i=1}^{N} (\prod_{k \in \Omega} z_{ik})$ is large

A priori algorithm

Step 1: Choose threshold t

Step 2: Let $\Omega_1, ..., \Omega_K$ be the K single item sets ($\Omega_1 = \{bread\}, \Omega_2 = \{jam\}, ...$). Compute $T(\Omega_1), ..., T(\Omega_K)$ and keep the Ω's for which $T(\Omega_k) > t$

Step 3: With the remaining sets of Step 2, build two-item sets and apply Step 2 to those sets ($\Omega_1 = \{bread, jam\}, \Omega_2 = \{jam, butter\}, ...$)

Step 4: With the remaining sets of Step 3, build three-item sets and apply Step 2 to those sets

The algorithm goes on and on until there are no more sets that have support greater than t

Cluster Analysis

- Grouping a collection of elements into subsets or "clusters"
- Objects need to be grouped such that each element in a cluster is closer or more related to the elements inside that cluster than to the element inside another cluster (Variance within $<$ Variance Between)
- Different ways to describe an object thus different clustering methods are needed
- Some clustering method can group elements based on distance measurements (K-means), other can arrange clusters into a natural hierarchy (hierarchical clustering)
- Before applying any clustering technique, one has to define the dissimilarity measure between two objects

Proximity Matrices

- There is two ways of gathering data, first we could just gather raw data inside a matrix X (what we are used to). We would then feed it into a clustering algorithm that will work with computing distances between data points for instance
- Sometimes the data can be a matrix D of size $N \times N$ representing the similarity (or dissimilarity) between pairs of objects.
- Most clustering algorithms expect a symmetric dissimilarity matrix in input, thus converting into a similarity matrix is necessary. If D is not symmetric it must be replace by the symmetric matrix $(D + D^T)/2$

Dissimilarities based on attributes

- Most of the time we have data in the form of the matrix $X = x_{ij}$ with $i = 1, ..., N$ and $j = 1, ...p$, (p Attributes).
- We want to feed a matrix of dissimilarities to the algorithm, thus we need to convert the data matrix X into a matrix of dissimilarity $D(x_i, x_i') = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$
- We define a dissimilarity function $d(x_{ij}, x_{i'j})$ that will provide a number to quantify the dissimilarity of any given pair of data point. We often use $d(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$. The resulting matrix will be a proximity matrix D as seen above.
- Note that squared distance may not appropriate to quantify dissimilarity between data point for categorical data, also assigning weights to attributes might provide a more accurate proximity matrix

Dissimilarities for continuous attributes

- Squared error: $d(x_i, x_{i'}) = (x_i - x_{i'})^2$
- Absolute error: $d(x_i, x_{i'}) = |x_i - x_{i'}|$
- Correlation error: $d(x_i, x_{i'}) = \rho(x_i, x_{i'})$

Dissimilarities for ordinal attributes

- Suppose the attributes can take values $i = 1, ..., M$ such as those values are ordered (ex: For grades A,B,C,D,F, A=1,B=2,...)
- We replace each i value by $\frac{i-1/2}{M}$ and then we treat the data just as continuous attributes

Dissimilarities for categorical attributes

- Assume the attribute has M possible classes, the difference between pairs must be explicitly described inside a matrix of size $M \times M$.
- We usually fill out the matrix with 1's for $i \neq j$ and 0 for $i = j$ and we can add some unequal difference between classes to emphasize some dissimilarities more than others.

Object Dissimilarity

- Now that we know how to obtain the p-individual attribute dissimilarity, we want to group then into a Dissimilarity Matrix $D(x_i, x_i')$ that quantifies the the difference between two objects x_i and x_i'.
- Most of the time: $D(x_i, x_i') = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j})$, with $\sum_{j=1}^{p} w_j = 1$
- The weight corresponding to each attribute is proportional to the variance of this variable over the entire data set.
- The error weights must be assigned carefully and need to be tailored from case-to-case. Some attributes contribute more to the overall dissimilarity of the groups than others. This picture shows how setting all the weight to $w_j = \frac{1}{2 * Var(X_j)}$ to standarsize the features, using the K-means method clearly failed the grouping clustering.

Object Dissimilarity

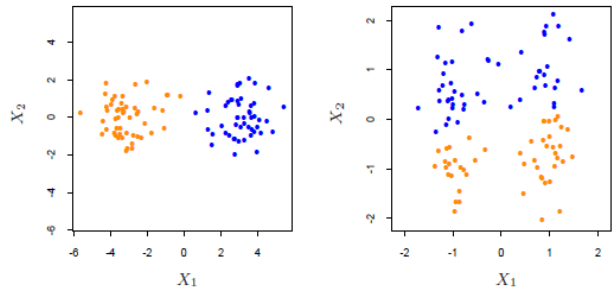


Figure 2: Standardization with weights ill-used (from the book)

- "Specifying an appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm"
- The appropriate dissimilarity measure needs to be determined with the help of scientists working on your project and could fail any clustering algorithm if not taken in consideration.

Clustering Algorithms

Goal: To partition data into cluster where the within cluster variance/scatter is the smallest and the between cluster variance/scatter is the largest.

The book defines 3 subgroups for clustering algorithms.

- Combinatorial
- Mixture Modeling
- Mode Seekers

The focus of the remainder of this chapter is Combinatorial Algorithms, but examples of mixture modeling can be found in Section 6.8 and mode seekers in section 9.3

Combinatorial Clustering Algorithms

Combinatorial Algorithms assign data to clusters without reference to the underlying probability of the data.

- Observations: x_i where $i \in \{1, 2, ..., N\}$
- Number of Clusters: K where $K < N$
- Cluster Assignment: $C(x_i) = k$ where $k \in \{1, 2, ..., K\}$

generally to preform these algorithms one wants find an optimal $C^*(x_i)$ such that a "loss function is minimized. Loss function is the degree to which the clustering has failed.

Combinatorial Clustering Algorithms

In some approaches the loss function is directly minimized and in the case of the book they use the natural loss function.

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

The $d(x_i, x_{i'})$ is denoted the dissimilarity function and will be dependent on the algorithm used. This is called the within cluster point scatter. In class we used a within cluster variance function.

Combinatorial Clustering Algorithms

The total point scatter is defined as,

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d(x_i, x_{i'}) =$$
$$\frac{1}{2}\sum_{k=1}^{K}\sum_{C(x_i)=k}(\sum_{C(x_{i'})=k} d(x_i, x_{i'}) + \sum_{C(x_{i'})\neq k} d(x_i, x_{i'}))$$

Which is defined as $T = W(C) + B(C)$ Where $W(C)$, within cluster, and $B(C)$, between cluster, point scatter. As we can see the job of minimizing $W(C)$ is equivalent to maximizing $B(C)$. An Iterative descent algorithm is needed in order to minimize $W(C)$

K-means

K-means is, according to the book, one of the most popular clustering method. Within the context of the previous slide the dissimilarity function for K-means is just Euclidean distance.

$$d(x_i, x_{i'}) = \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2$$

The algorithm now aims to minimize the following,

$$\begin{aligned} W(C) &= \frac{1}{2}\sum_{K=1}^{k}\sum_{C(x_i)=k}\sum_{K=1}^{k}\sum_{C(x_{i'})=k}||x_i - x_{i'}||^2 \\ &= \sum_{k=1}^{k} N_k \sum_{C(x_i)=k}||x_i - \bar{x}_k||^2 \end{aligned}$$

Where $N_k = \sum_{i=1}^{N} I(C(x_i) = k)$ and $x_k = (\bar{x}_{1k}, ..., \bar{x}_{pk})$ which are the mean vector of each cluster.

K-means

An Introduction to Statistical Learning has the notation that follows with the notation used in class. For every cluster the difference between observations should be minimized. This measure of within cluster difference is defined as.

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

K-means

K-means Clustering Algorithm (ELS pg 510)

---

1. For a given cluster assignment C, the total cluster variance is minimized with respect to $\{m_1, m_2, ..., m_k\}$ yielding the means of the currently assigned clusters.

2. Given a current set of means $\{m_1, m_2, ..., m_k\}$, is minimized by assigning each observation to the closest cluster mean. That is ,

$$C(x_i) = argmin_{1 \leq k \leq K} ||x_i - m_k||^2$$

3. Steps 1 and 2 are iterated until the assignments do not change.

K-means

Notes on the Algorithm

- As discussed in class the stopping criteria should be a global measure, namely the sum of all within cluster variance $\sum_{\forall \ell} W(C_\ell)$
- Algorithm's starting point could effect the out come due to the algorithm finding local maximums rather than global maximums.
- Repeating K-means with different random starting points helps reduce the effect the initial values have on the outcome.

K-Medoids

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.

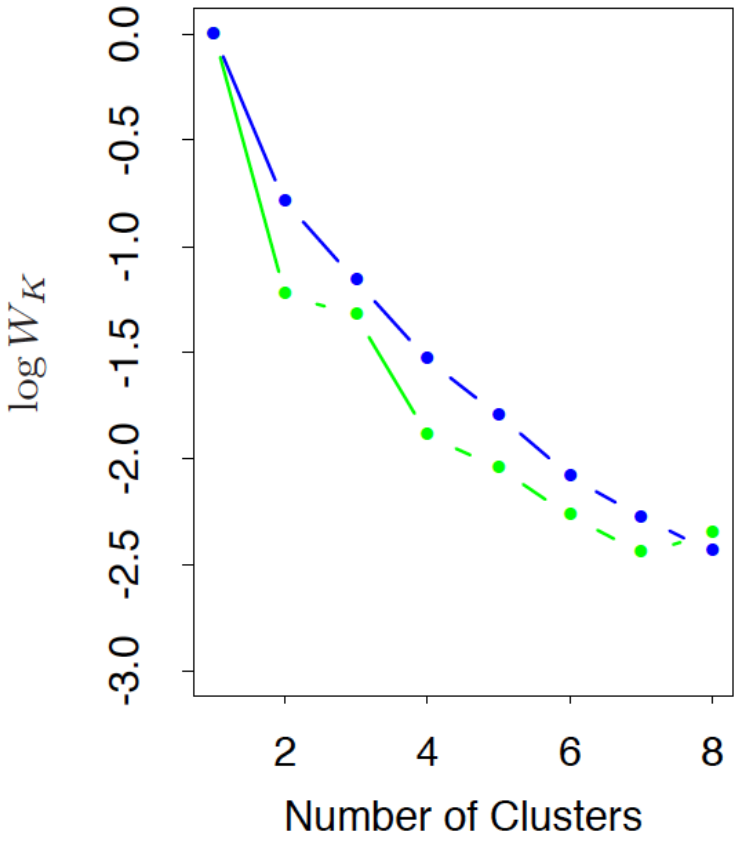
Vector Quantization

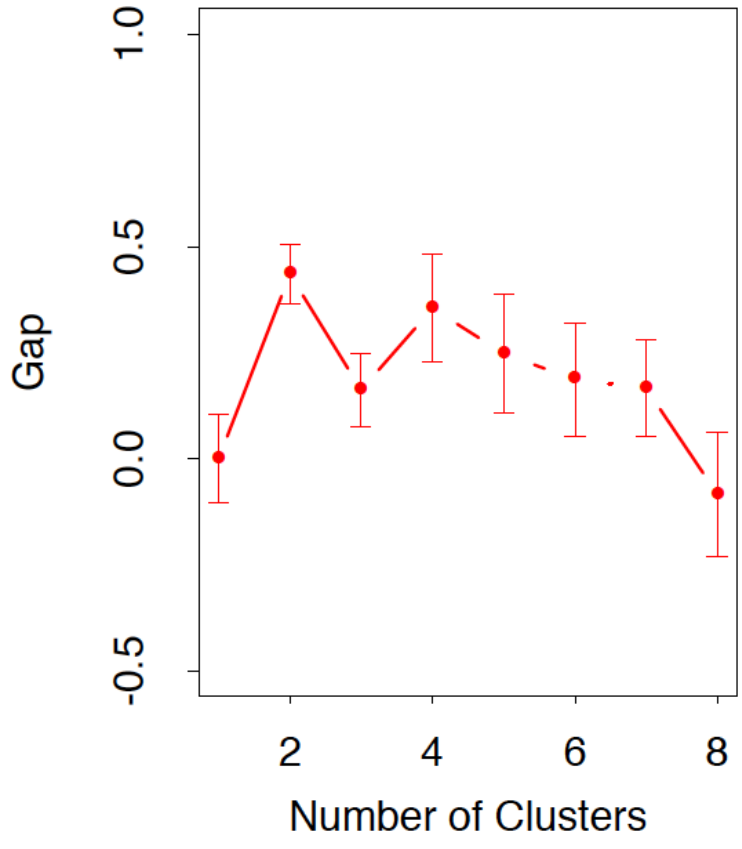
- A data compression technique
- The group of centroids is the code book.

How do we choose K?

- The Gap statistic proposed by Tibshirani et al., 2001b compares the curve logWk to the curve obtained from data uniformly distributed over a rectangle containing the data.
- The point with the largest separation between the two curves is the optimal K number of clusters
- This appears as a kink in the plots of Wk.

How do we choose K?





Hierarchical Clustering

Agglomerative: bottom-up

- Start with N clusters
- Recursively join the closest two clusters according to their dissimilarity
- The "nearest-neighbor" technique

Divisive: Top Down

- Start with 1 cluster
- Use K-means or K-medoids to make binary splits

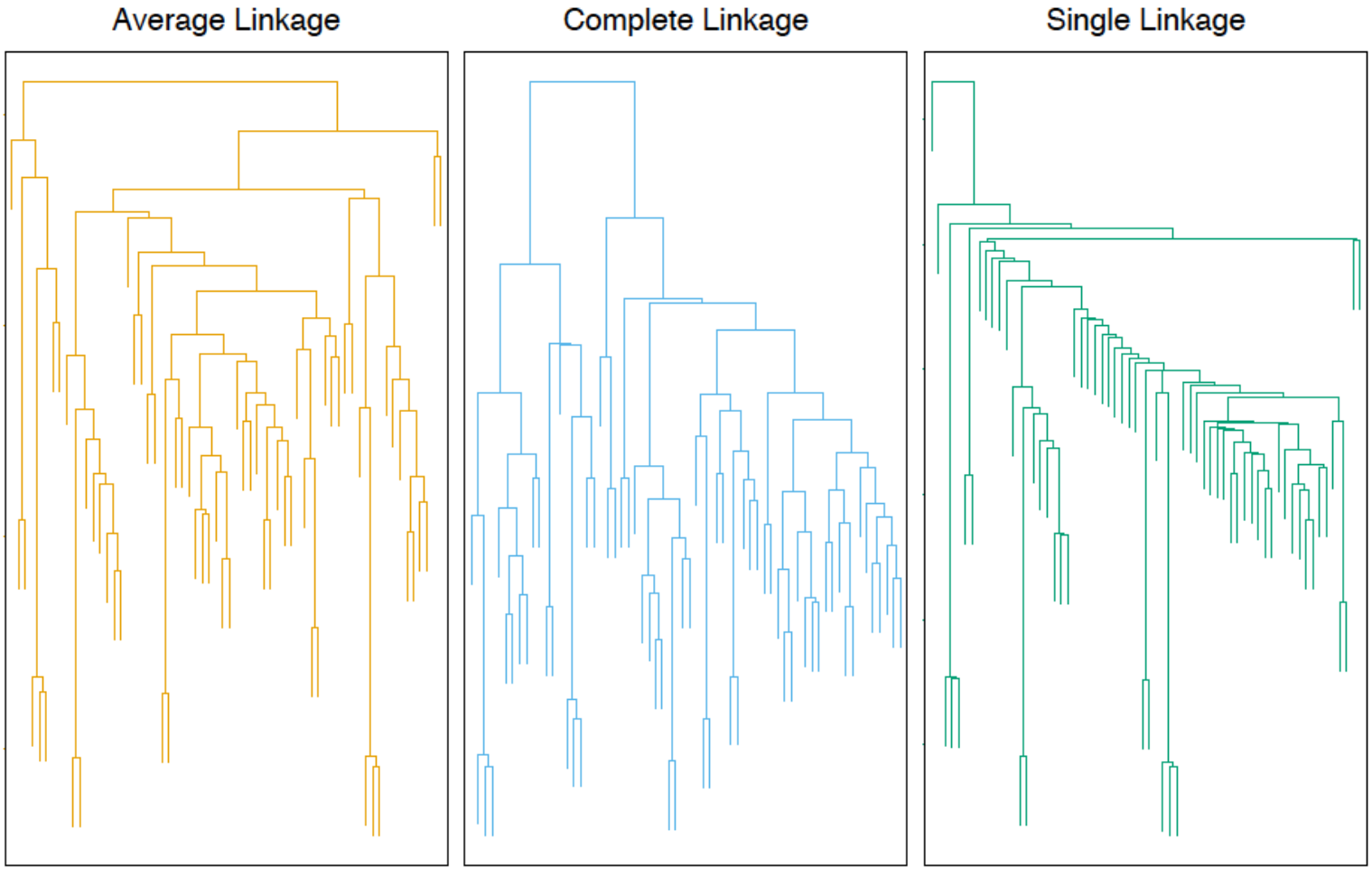
Agglomerative Linkage for Dissimilarity

Linkage	*Description*
Complete	Maximal intercluster 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 intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster 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*.

Divisive Clustering

- This approach has not been studied as much as agglomerative methods
- Start with a single cluster G
- The observation with the largest average dissimilarity becomes the first observation in the second cluster H (Macnaughton Smith et al. 1965).

Dendrograms



Dendrograms

- By their very nature, a hierarchical structure is forced on the data
- Can be interpretable
- Biologists can interpret the gene clusters in terms of biological processes

Self-Organizing Maps

Self-Organizing Maps utilize mapping higher dimensional observation into a two-dimensional grid in order to cluster the variables.

Notation

- K prototypes or points, $m_j \in R^p$, that lie on a two-dimensional rectangular grid.
- m_j are parameterized by a coordinate pair $\ell_j \in Q_1 x Q_2$, where $Q_i = 1, 2, .., q_i$ and $K = q_1 q_2$

Self-Organizing Maps

The method

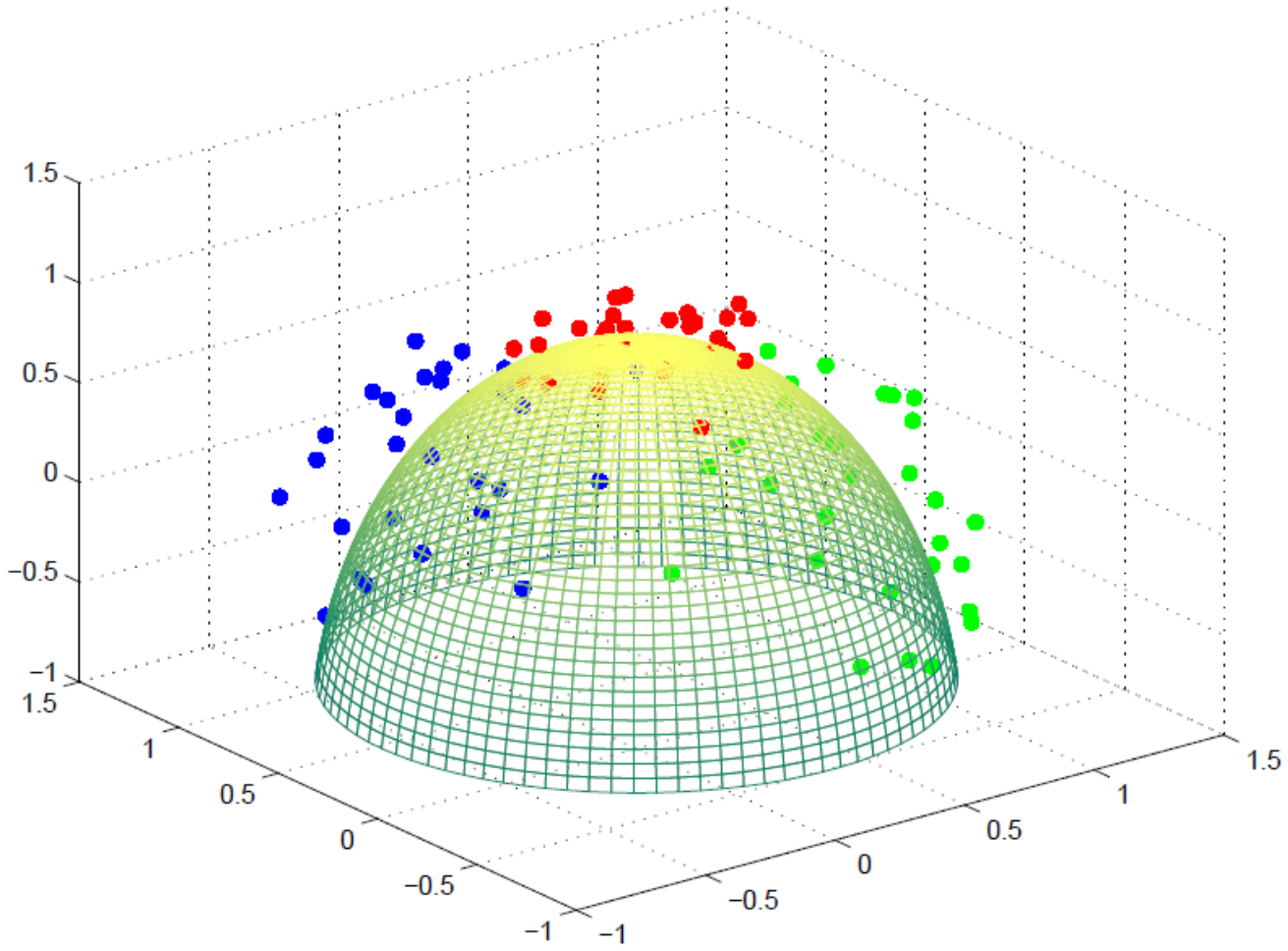
- The m_j are initialized and placed on the grid.
- for each x_i the closest m_j is found.
- For all neighbors, m_k, of m_j are moved towards x_i via the following update

$$m_k \leftarrow m_k + \alpha(x_i - m_k)$$

Neighborhood of m_j is all of the prototypes, m_k, such that ℓ_j and ℓ_k are close in distance. Usually Euclidean distance is used and threshold, r, is used to determine which prototypes are considered close. This threshold has an initial starting value of R that is decreased to 1 after a few thousand iterations. The learning rate α is also decreased over time from 1.0 to 0.0

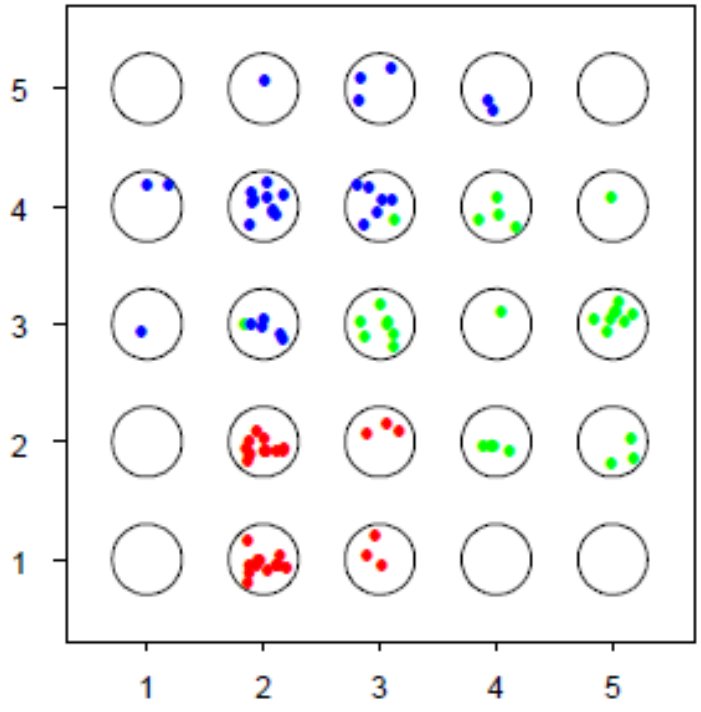
Self-Organizing Maps Brief Example

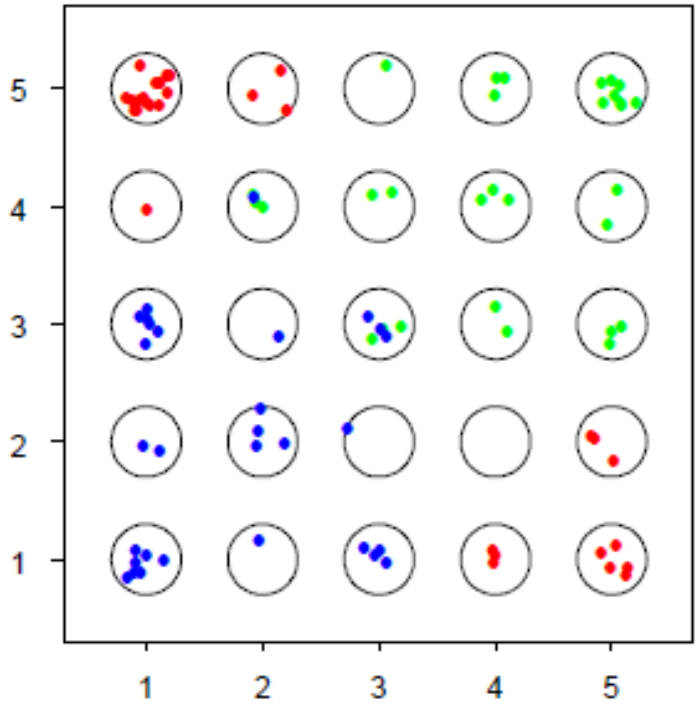
90 observations which represents points around the surface of a sphere are generated. The groups are colors red, green, and blue with they cluster residing at $(0, 1, 0)$, $(0, 0, 1)$, and $(1, 0, 0)$ respectively.



Self-Organizing Maps Brief Example

The SOM method is applied, with a 5x5 grid of prototypes. The method was applied 40 times to the 90 observations with 3600 iteration.





Principle Component Analysis

Principle Component Analysis is a method that represents the data in a lower dimension via linear combinations of the predictors.

We begin with p predictors $X_1, ..., X_p$ for n observations that have been normalized. Then the Principle Components are denoted,

$$Z_j = \phi_{j1}X_1 + ... + \phi_{jp}X_p$$

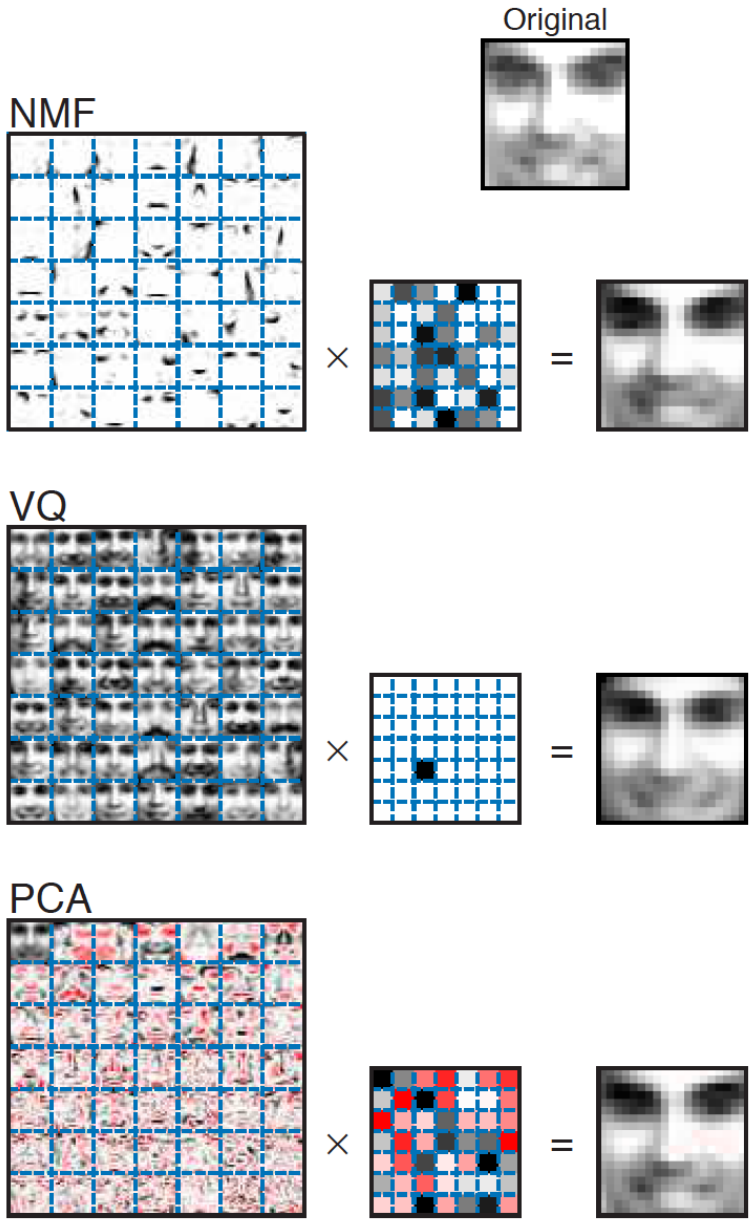
Where the Z_j are independent, $Var(Z_1) \geq Var(Z_2) \geq ...Var(Z_p)$, and $Var(Z_j) = \lambda_j$ the normalized eigen values of the covariance matrix. The $\phi_j = \{\phi_{j1}, ..., \phi jp\}$ are the loadings and the Z_j's are call the scores.

Non-negative Matrix Factorization

- Alternative to PCA
- Data and components are assumed to be nonnegative
- Approximate X by WH

$$w_{ik} \leftarrow w_{ik} \frac{\sum_{j=1}^{p} h_{kj} x_{ij} / (\mathbf{WH})_{ij}}{\sum_{j=1}^{p} h_{kj}}$$
$$h_{kj} \leftarrow h_{kj} \frac{\sum_{i=1}^{N} w_{ik} x_{ij} / (\mathbf{WH})_{ij}}{\sum_{i=1}^{N} w_{ik}} \qquad (14.74)$$

Non-negative Matrix Factorization



The Google PageRank Algorithm

- Brief description of the Original Google PageRank algorithm to rank N pages by relevance.
- The more a web page is mentioned in other web pages, the more relevant it is considered by the algorithm. A link is created whenever a website mentions another, however those links don't have the same weights! Weights will depend on the quality of the website judged by the algorithm (popularity, number of links etc.)

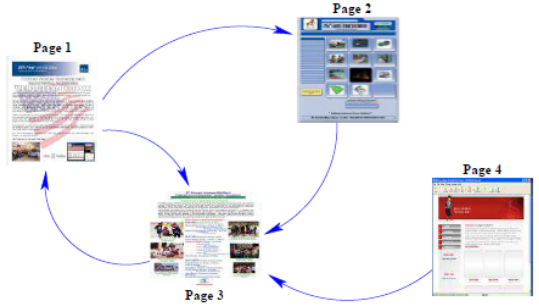


Figure 3: Example of small network of web pages (from the book)

The Google PageRank Algorithm

- "The idea is that the importance p_i of page i is the sum of the importances of pages that point to that page".
- Page relevance are recursively defined such that:

$$p_i = (1 - d) + d \sum_{j=1}^{N} \frac{L_{ij}}{c_j} \cdot p_j$$

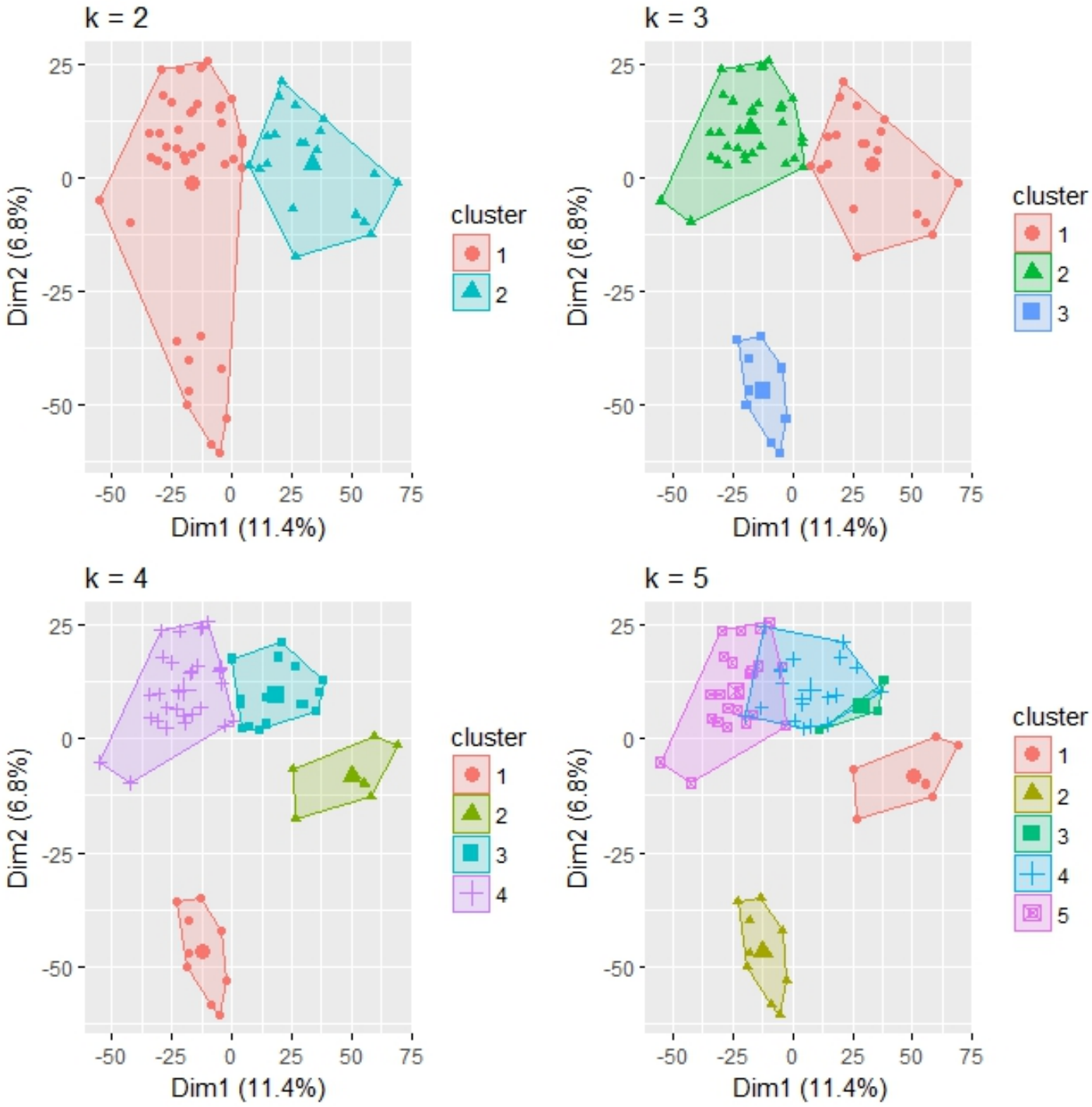
- $Lij = 1$ if page j points at page i, zero otherwise, $c_j =$ Total # of pages pointed by page j
- Different kind of clustering but still use the same basics mentioned earlier about defining how to measure dissimilarities and how to combine them with the attributes and weight those wisely.

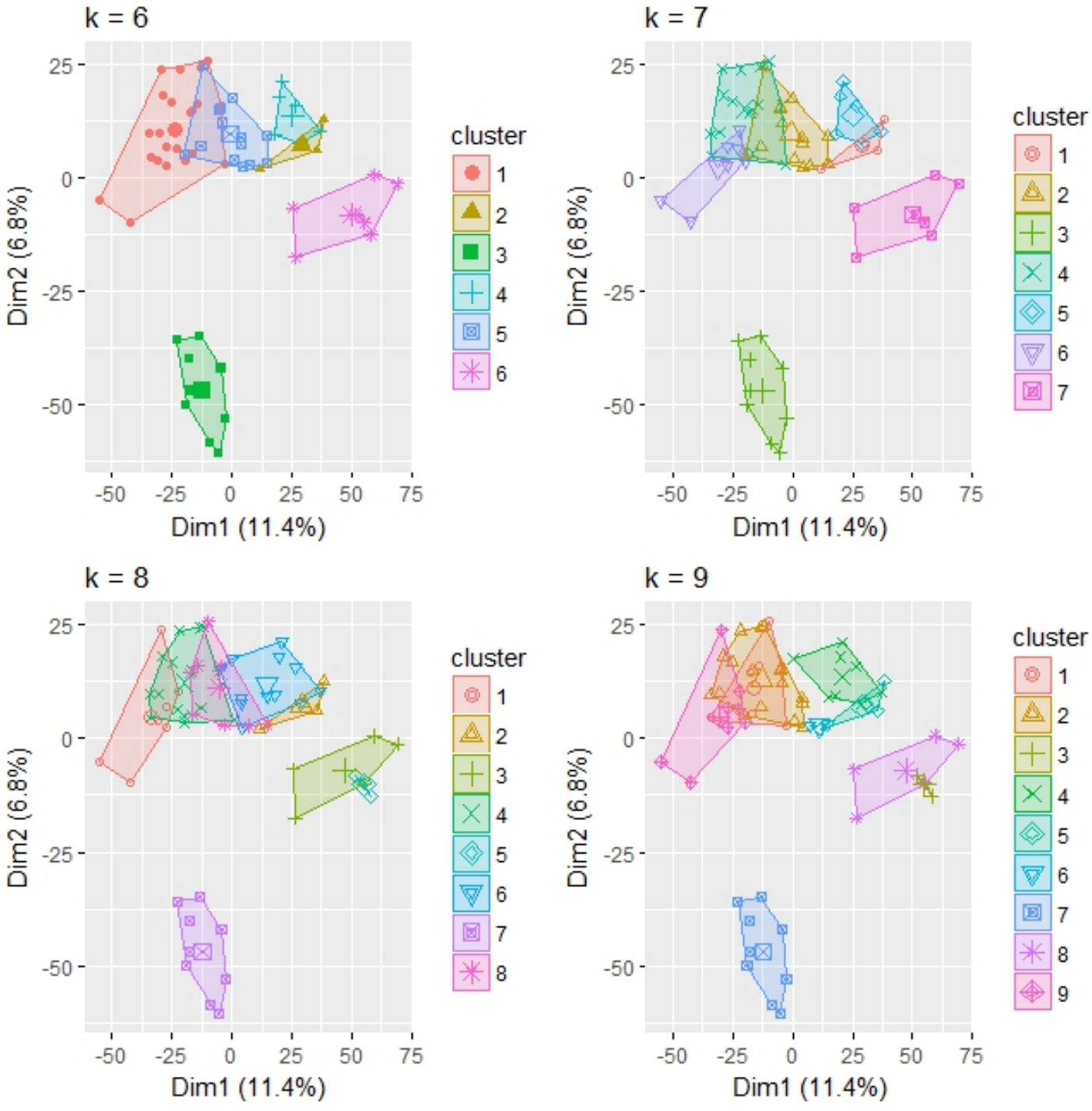
National Cancer Institute (NCI) Data

- 6830 genes expression tested on 64 cancerous cells
- K-means approach with k from 2 to 9
- Mixture models approach using Mclust()

K-means

- kmeans(data, centers $= 2$, nstart $= 15$)
- K-means is very dependent on the initial random location of the centroids, thus the algorithm will generate 15 random initial centers and pick the best set of centroids.
- We display the output produces by the K-means method for k going from 2 to 9 using the function fvizcluster() from the package f actoextra

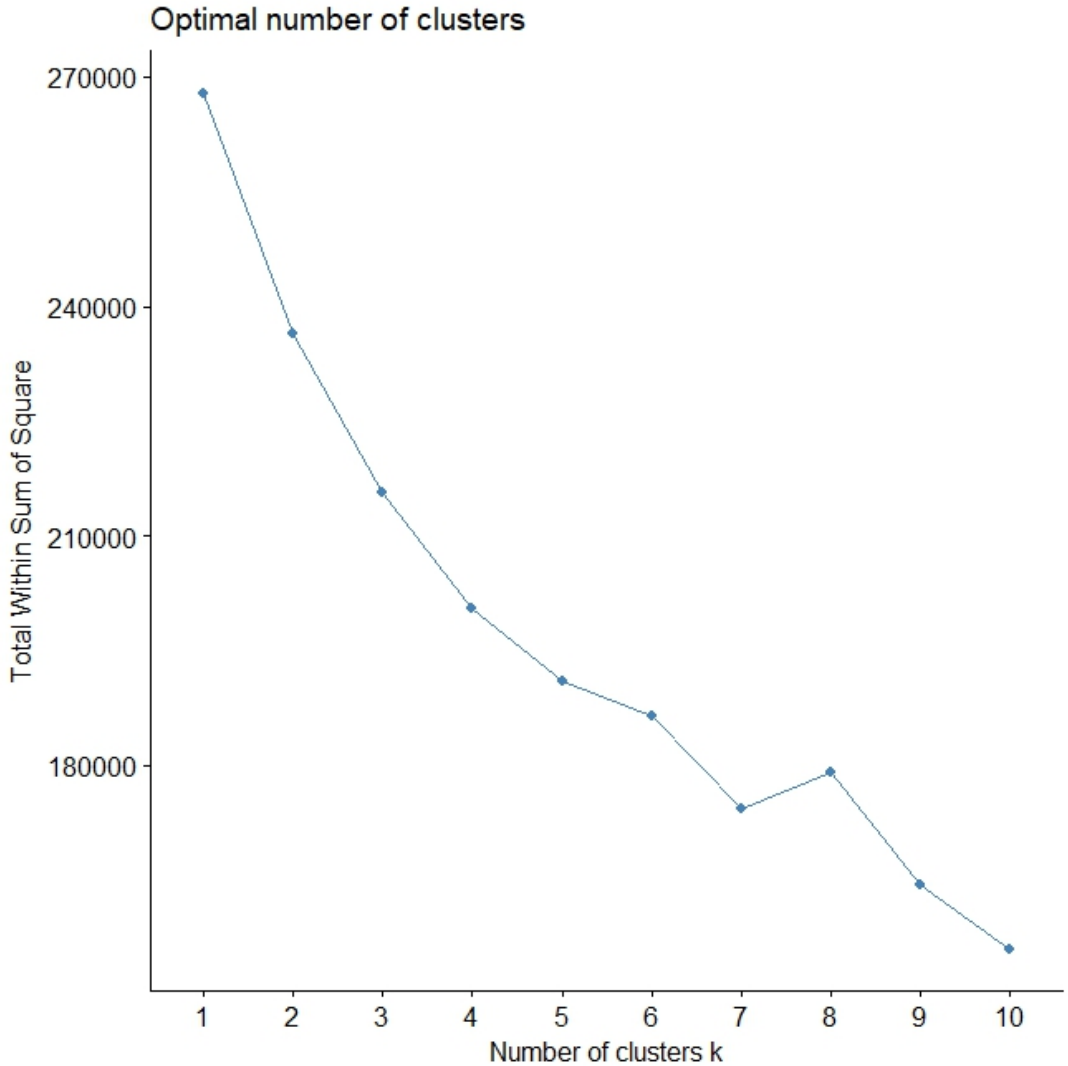

k = 2
k = 3
k = 4
k = 5
Dim1 (11.4%)
Dim2 (6.8%)
cluster
1
2
3
4
5
25
0
-25
-50
-50
-25
0
25
50
75


k = 6
k = 7
k = 8
k = 9
Dim1 (11.4%)
Dim2 (6.8%)
cluster
25
0
-25
-50
-50
-25
0
25
50
75
1
2
3
4
5
6
7
8
9

K-means

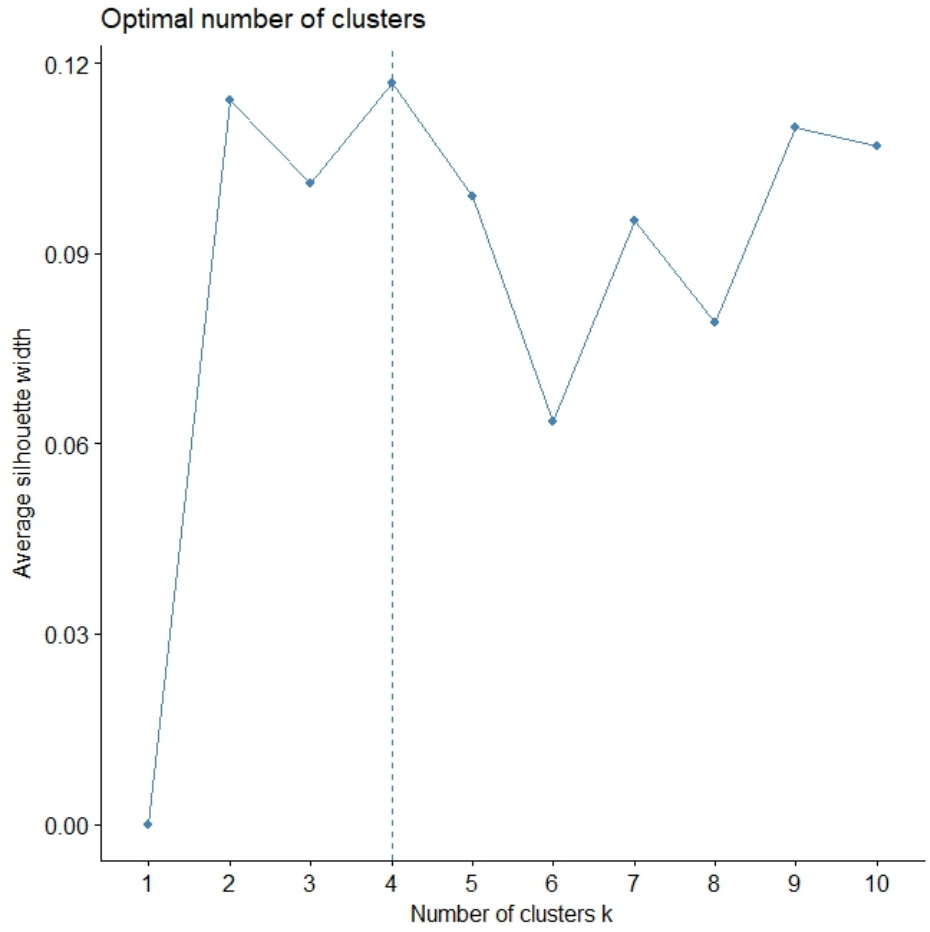
- Any guess on what the optimal k might be according to those plots? k $= 4$ seems like a good value.
- "Elbow method": The optimal k corresponds to the value where the bend of the curve $W(k)$ vs k is. Here, $k = 4, 5$ would work.

K-means



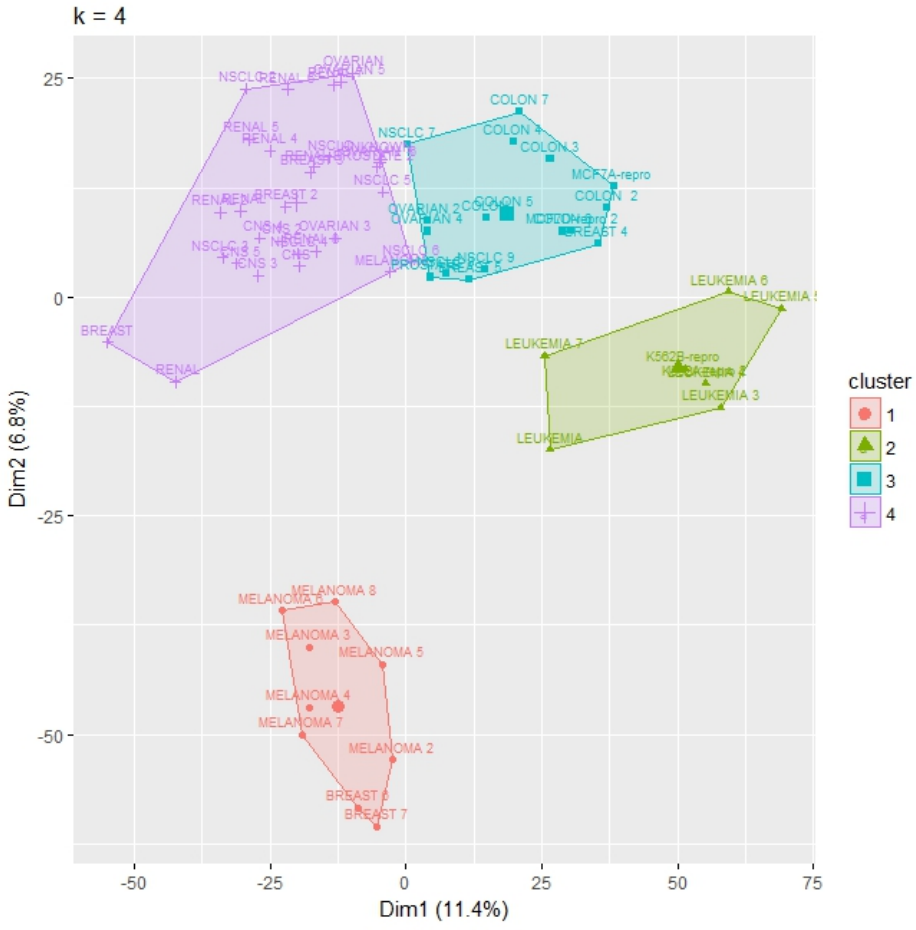
"Silhouette method": Describes graphically how well each object lies within its cluster as opposed to other clusters.

K-means



The higher the silhouette (between -1 and 1), the better. Thus once again $k = 4$ seems like the optimal number of clusters. We have enough evidence for $k = 4$ thus we can use k-means with $k = 4$

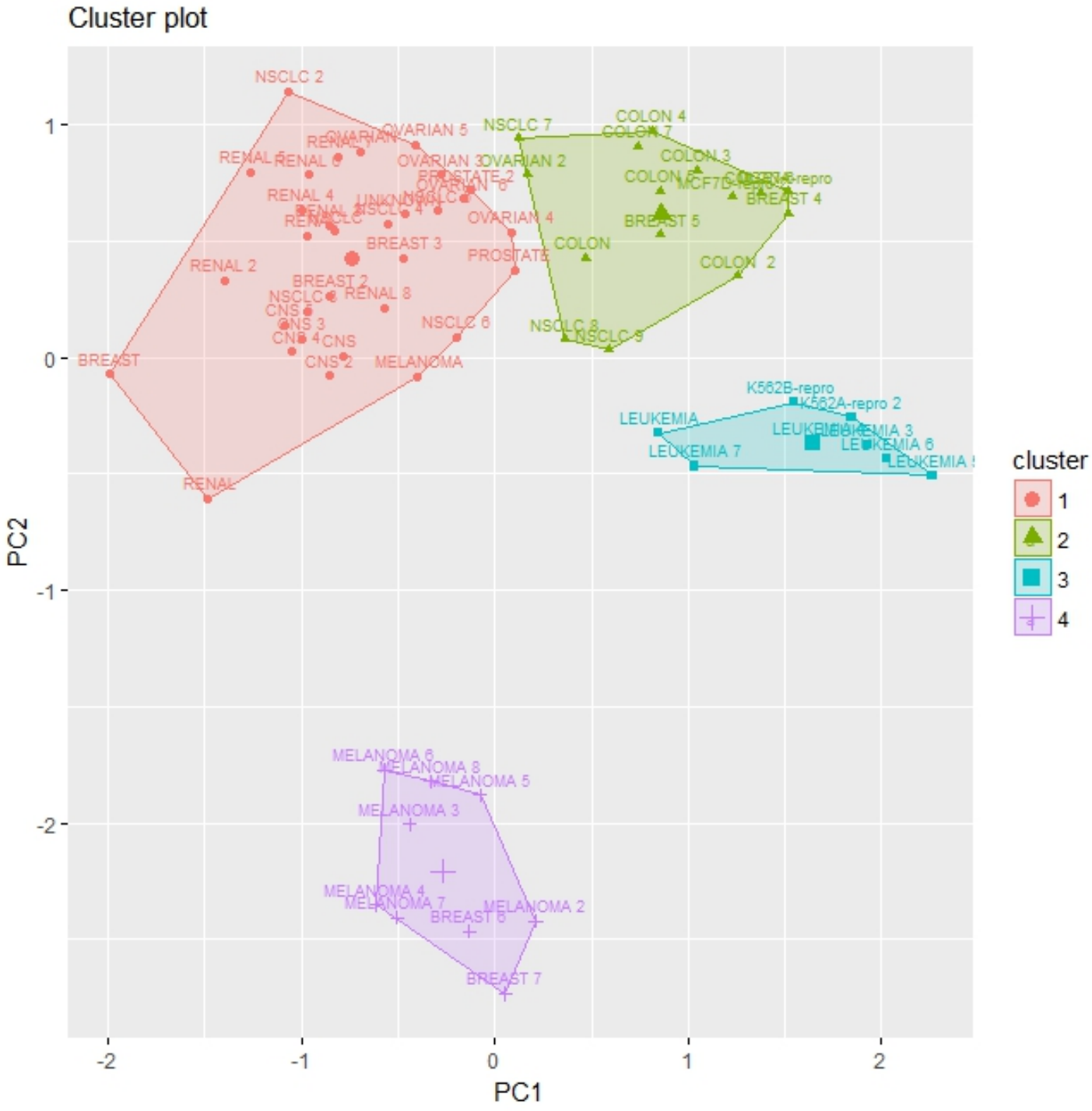
K-means



It seems that the activity of some specific genes can be associated with certain type of cancer...

Clustering based of mixture models

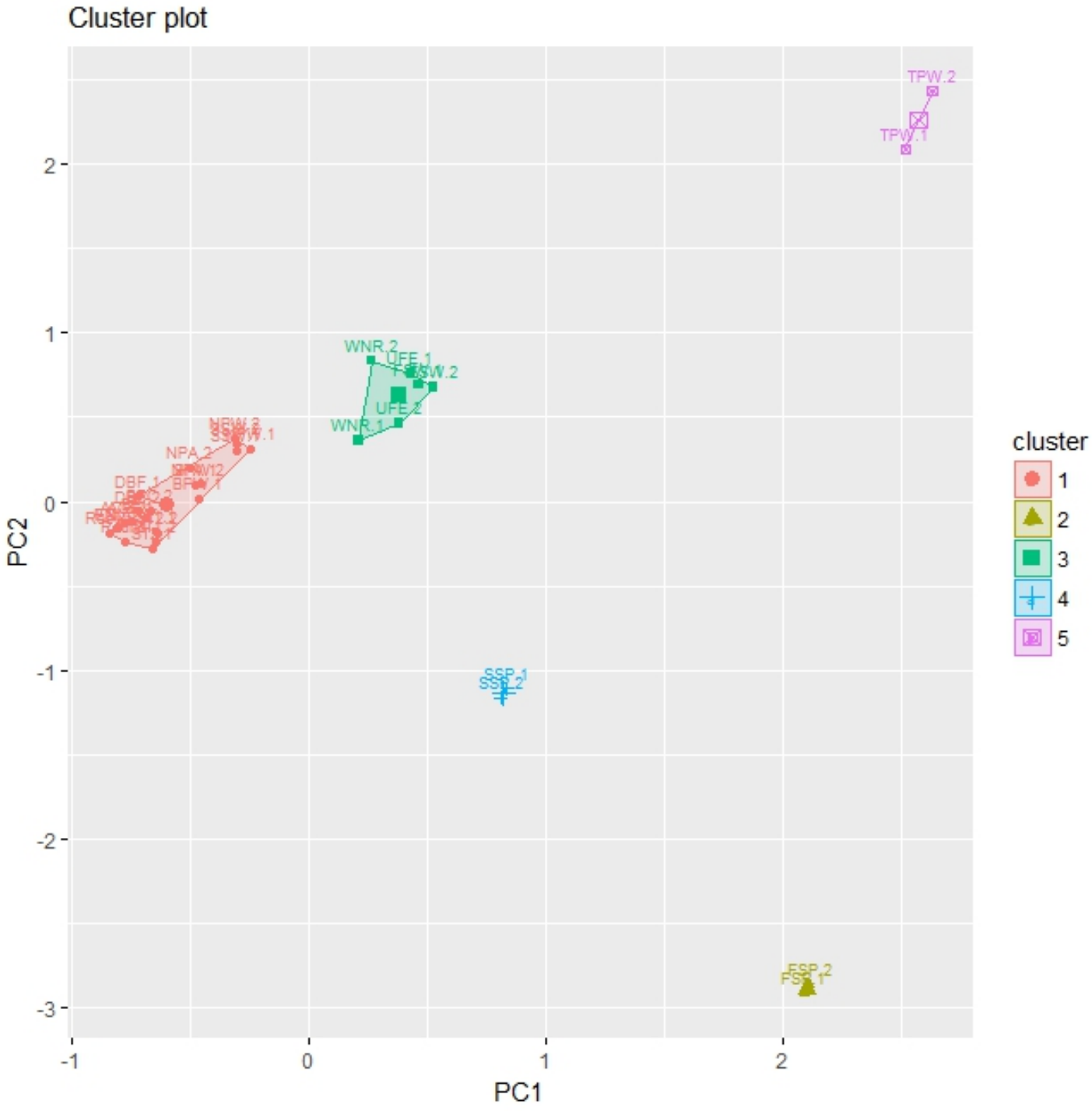
- Is Mclust() getting the same results as K-means with k $= 4$?
- Combining a Principal Component Analysis on the data and using the first two components with Mclust() gives us:
- Those results are very similar to the ones obtained with K-means, but we did not have to determine "manually" the optimal k

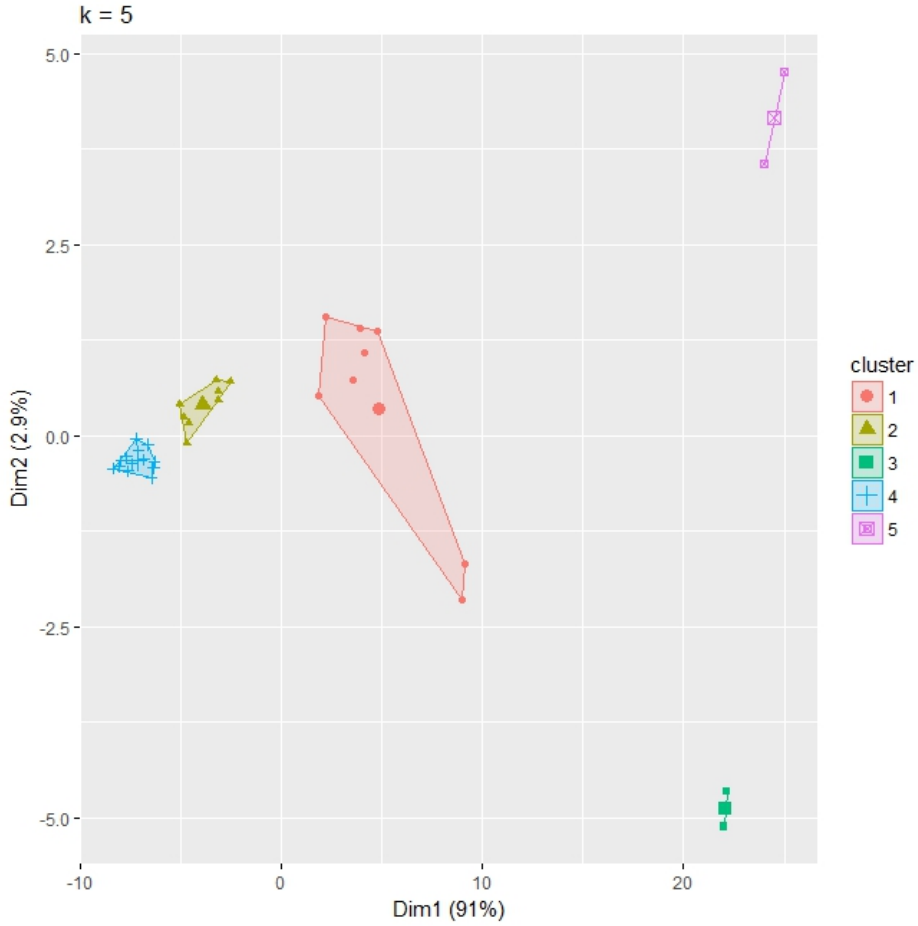

Cluster plot
NSCLC 2
OVARIAN 5
NSCLC 7
COLON 4
COLON 7
RENAL 1
RENAL 5
RENAL 6
OVARIAN 3
OVARIAN 2
COLON 3
PROSTATE 2
COLON 5
MCF7D-repro
BREAST 4
OVARIAN 6
RENAL 4
UNKNOWN
NSCLC 1
NSCLC 4
OVARIAN 4
BREAST 5
RENAL 3
NSCLC
BREAST 3
COLON
PROSTATE
COLON 2
RENAL 2
BREAST 2
NSCLC 8
RENAL 8
CNS 1
NSCLC 6
NSCLC 9
CNS 3
CNS 4
CNS
BREAST
CNS 2
MELANOMA
K562B-repro
K562A-repro 2
LEUKEMIA
LEUKEMIA 3
LEUKEMIA 6
LEUKEMIA 7
RENAL
MELANOMA 6
MELANOMA 8
MELANOMA 5
MELANOMA 3
MELANOMA 4
MELANOMA 7
MELANOMA 2
BREAST 6
BREAST 7
PC2
PC1
1
0
-1
-2
-2
-1
0
1
2
cluster
1
2
3
4

Tennis Woman French Open Data

- 76 Tennis matches from the Woman 2013 French Open, 42 statistics about the match are collected for both players, such as the number of aces or the total number of a player.

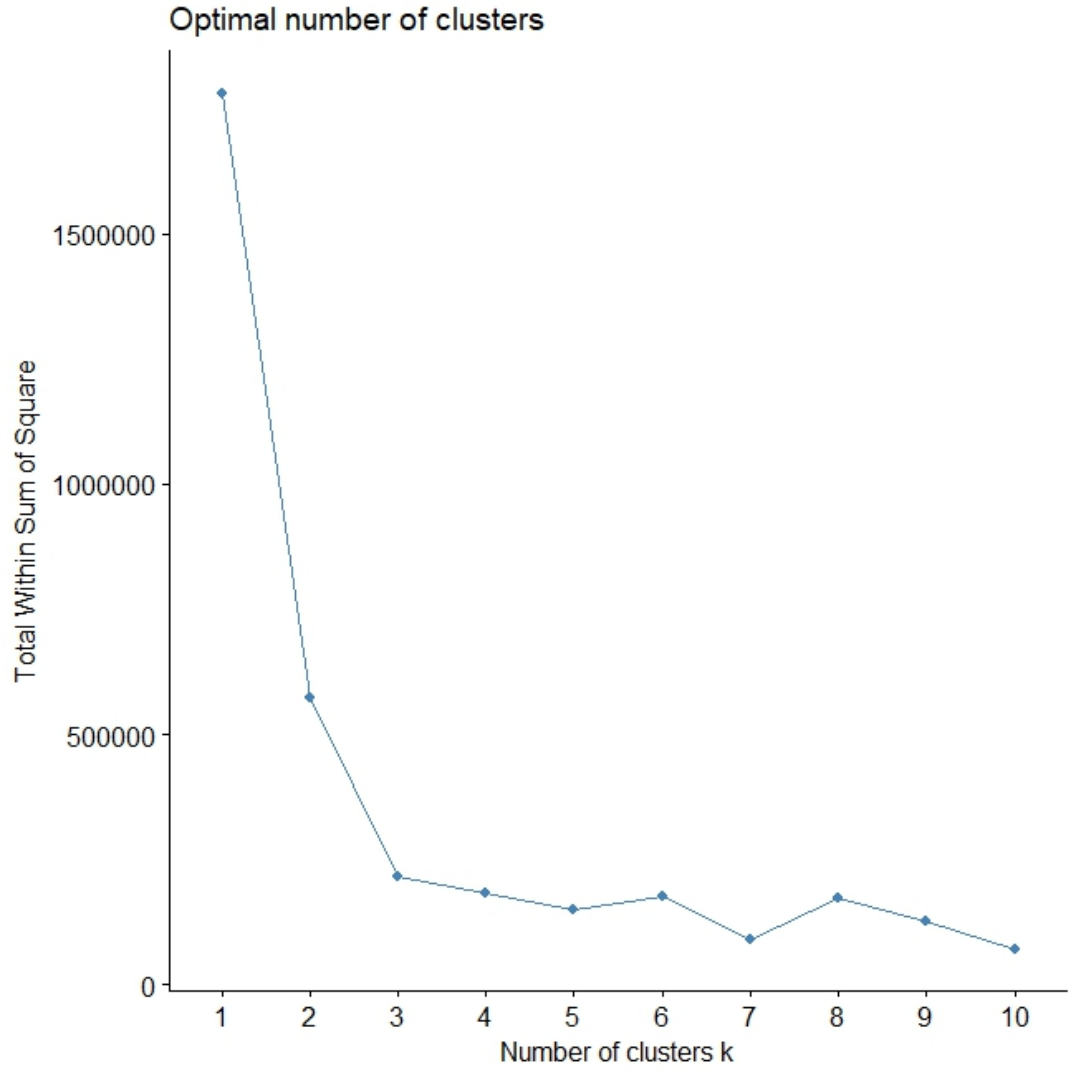
- Running a PCA on the data and applying Mclust() gives us:

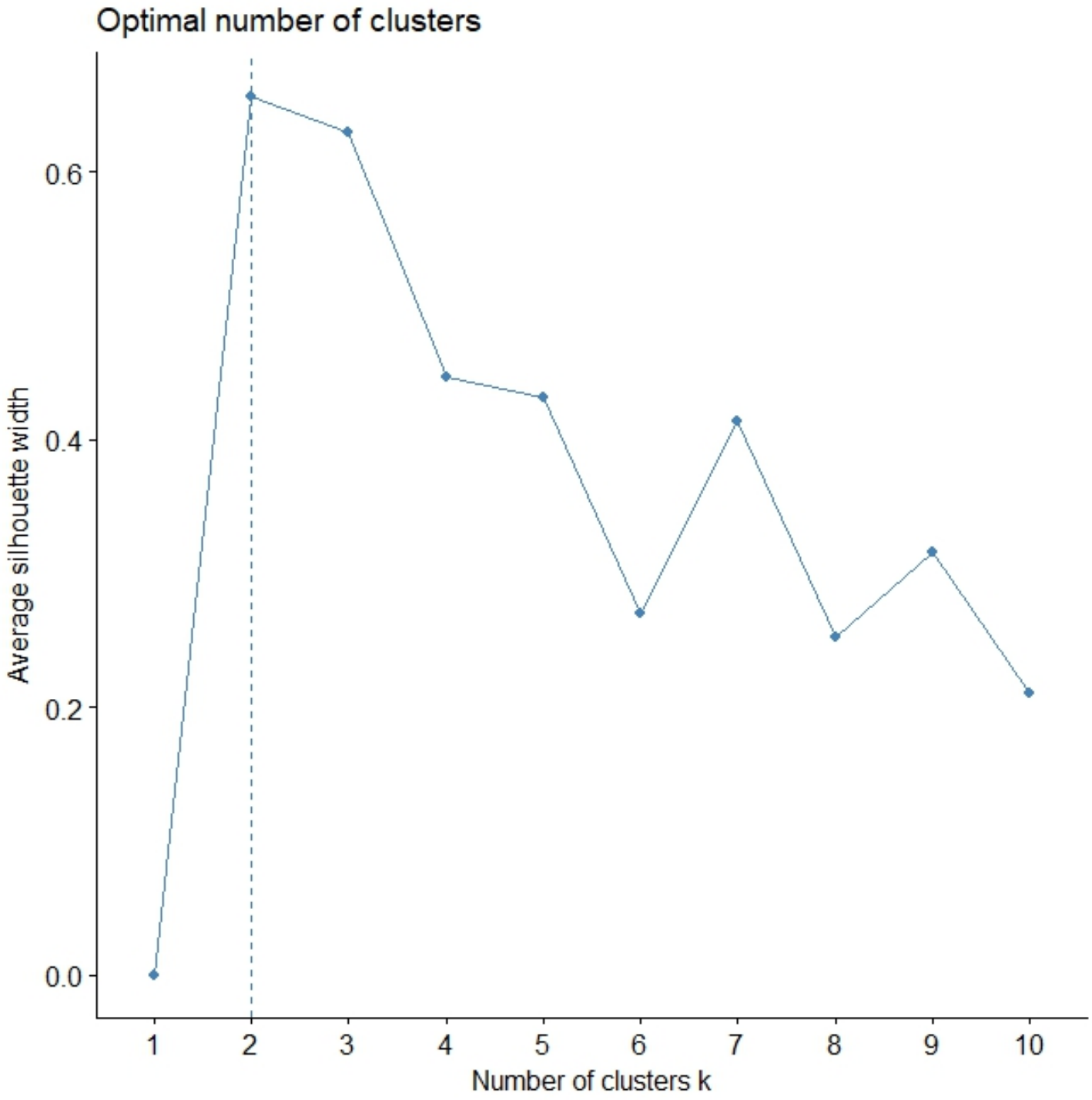

Cluster plot
TPW.2
TPW.1
WNR.2
UFE.1
UFE.2
WNR.1
NPA.2
DBF.1
SSP.1
SSP.2
FSP.2
FSP.1
2
1
0
-1
-2
-3
PC2
-1
0
1
2
PC1
cluster
1
2
3
4
5



Running K-means with 5 clusters does not give us the same clusters:

In fact, the Elbow method and Silhouette method suggest a lower number of clusters...




Optimal number of clusters
Average silhouette width
0.0
0.2
0.4
0.6
1
2
3
4
5
6
7
8
9
10
Number of clusters k

In Conclusion

We have discussed a multitude of notation and methods that are required of unsupervised learning. Recap on the methods presented were,

- Association Rules
- K-means and K-medoids
- Vector Quantization
- Hierarchical Models
- Self-Organizing Maps
- Non-negative Matrix Factorization
- Google Park Rank Algorithm

When then compared the K-means method of clustering to a mixture model method of clustering in our example.

Are there any Questions?

Reference

- Hastie, T., Tibshirani, R., & Friedman, J. H. (2017). The elements of statistical learning: Data mining, inference, and prediction. New York, NY: Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statistical learning with applications in R. New York: Springer.
- K-means Cluster Analysis. (n.d.). Retrieved from https://uc-r.github.io/kmeans_clustering .