

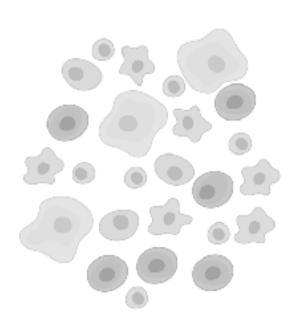
Susan Holmes Wolfgang Huber Laura Symul October 9,2019

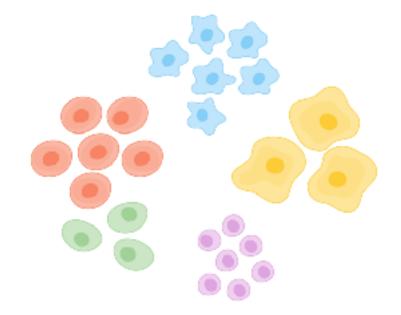
Clustering =

Finding a latent or hidden variable which is not necessarily provided/accessible

Clusters =

Groups in the population (of individuals, cells, gene expression profiles) in which the **individual elements are similar to each others** and **different from the elements of other groups**.





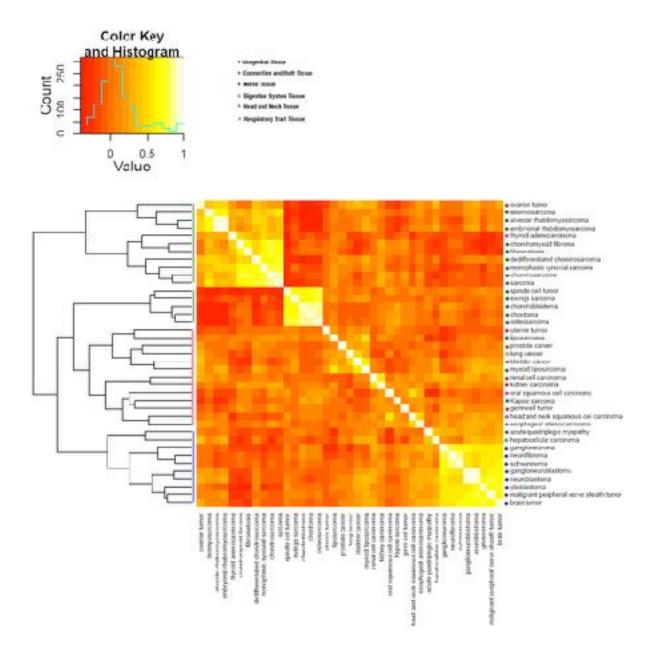
Some **measurements** (imaging, single cell RNA-seq, etc.)

Groups = cell types

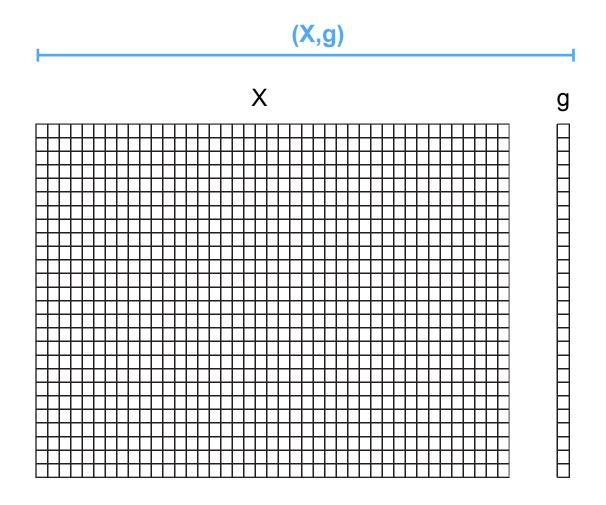
What is the central statistical concept behind any clustering?

Metric of (dis)similarity: a distance

Clusters frequently occur in biology. Cell type, tissue, tumor type, ethnicities, etc.



Data Augmentation is the aim of clustering



data augmentation methods are methods that add useful but unknown components to the data

Goals for this lecture

- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**

- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.

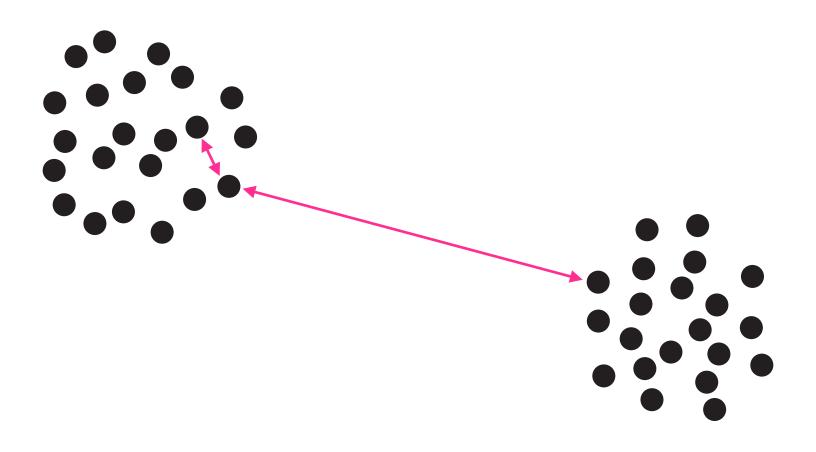
How (dis)similar are 2 elements?





(dis)similarity measure = **distance**

How (dis)similar are 2 elements?



How (dis)similar are 2 elements?

The choice or definition of **distance**

depends on the data

Euclidean distance (L2)

$$d(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_p - b_p)^2}.$$

Manhattan distance (L1)

$$d(A,B) = |a_1 - b_1| + |a_2 - b_2| + \ldots + |a_p - b_p|.$$

Maximum distance (L∞)

$$d_{\infty}(A, B) = \max_{i} |a_i - b_i|.$$

Minkowski distance (Lm)

$$d(A,B) = ((a_1-b_1)^m + (a_2-b_2)^m + \ldots + (a_p-b_p)^m)^{rac{1}{m}}.$$

Edit (Hamming) distance

Binary distance

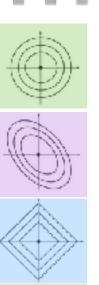
$$d_J(S,T) = 1 - J(S,T) = rac{f_{01} + f_{10}}{f_{01} + f_{10} + f_{11}}$$
 .

Correlation-based distance

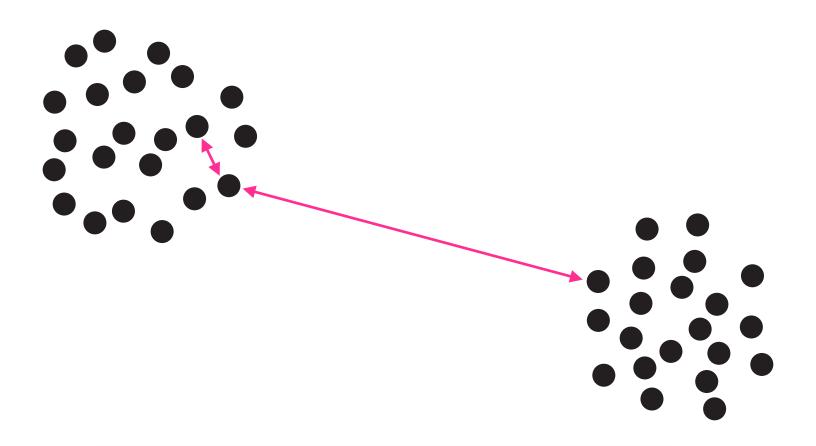
$$d(A,B) = \sqrt{2(1-\operatorname{cor}(A,B))}.$$

Weighted Euclidean distance

Mahalanobis distance



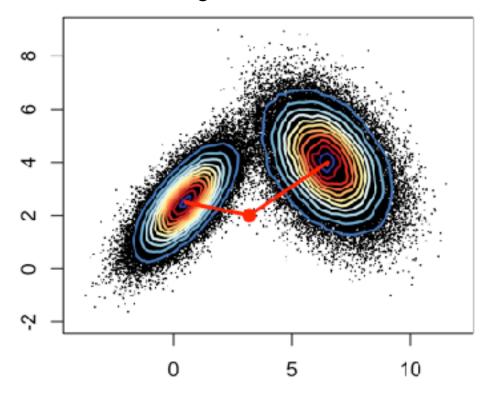
Euclidean distance



$$d(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_p - b_p)^2}.$$

Weighted Euclidean distance

Is the **red point**, a "new" datapoint, closer to the cluster on the right or the cluster on the left?



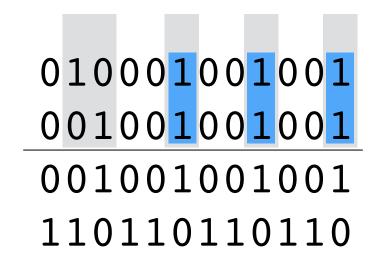
The weight on the dimensions depends on the cluster

Binary distance

The sum of elements that are the same

The number of elements

Jaccard distance



The **co-occurrence** is more important than the co-absence

1- The sum of 1's that overlap
The union of 1's

Initially developed for ecological data for the occurrence of traits or features

Edit (Hamming) distance



The number of edits that needs to be done.

This could be applied to nucleotide or amino acid sequences...

```
- mtvelpstqralvfdtwngplevrqvpvpspaddeil
mstagkvikckaavlwelkkpfsieevevappkahevr
- - - msipetqkgvifyeshgkleykdipvpkpkanell
- - manqvirckaavaweagkplsieeievappkahevr
mgtqgkvikckaaiawktgsplcieeievsppkacevr
```

...although in that case, the **different character substitutions** are usually associated with **different contributions to the distance** (to account for physical or evolutionary similarity), and **deletions and insertions** may also be allowed

Distances in R

```
mx = c(0, 0, 0, 1, 1, 1)
my = c(1, 0, 1, 1, 0, 1)
mz = c(1, 1, 1, 0, 1, 1)
mat = rbind(mx, my, mz)
```

Euclidean

```
## mx my
## my 1.732051
## mz 2.000000 1.732051
```

Binary

```
## mx my
## my 0.6000000
## mz 0.66666667 0.5000000
```

```
mut = read.csv("../data/HIVmutations.csv")
mut[1:3, 10:16]
    p32I p33F p34Q p35G p43T p46I p46L
## 1
            1
## 3
library("vegan")
mutJ = vegdist(mut, "jaccard")
mutC = sqrt(2 * (1 - cor(t(mut))))
Jaccard
mutJ
               2
                     3
##
## 2 0.800
## 3 0.750 0.889
## 4 0.900 0.778 0.846
## 5 1.000 0.800 0.889 0.900
Correlation-based
as.dist(mutC)
                  3
             2
##
## 2 1.19
## 3 1.10 1.30
```

4 1.32 1.13 1.30

5 1.45 1.19 1.30 1.32

15

Distances in R

```
phyloseq package
```

```
phyloseq: analyze microbiome
         census data using R
```

The phyloseg package is a tool to import, store, analyze, and graphically display complex phylogenetic sequencing data that has already been clustered into Operational Taxonomic Units (OTUs), especially when there is associated sample data, phylogenetic tree, and/or taxonomic assignment of the OTUs.

```
SDPCoA
[1] "dpcoa"
$JS0
[1] "jsd"
$veqdist
[1] "manhattan" "euclidean" "camberra"
                                                           "kulczynski" "jaccard"
                                             "bray"
                                                                                      "gower"
                                                                                                    "altGower"
                                                                                                                 "morisita"
[10] "horn"
                  "mountford" "raup"
                                                           "chao"
                                                                         "cao"
                                              "binomial"
$betadiver
[21] "rlb" "sim" "ql"
```

Almost 50 different distances

\$dist

[1] "maximum"

\$designdist

[1] "ANY"

> library("phyloseq")

> distanceMethodList

[1] "unifrac" "wunifrac"

\$UniFrac

"binary"

"minkowski"

If you **really** need to define a **new distance**, it must...

Definition [edit]

A **metric space** is an ordered pair (M, d) where M is a set and d is a metric on M, i.e., a function

$$d: M \times M \to \mathbb{R}$$

such that for any $x,y,z\in M$, the following holds:[2]

- 1. $d(x,y) = 0 \Leftrightarrow x = y$ identity of indiscernibles
- 2. d(x,y) = d(y,x) symmetry
- 3. $d(x,z) \leq d(x,y) + d(y,z)$ subadditivity or triangle inequality

Given the above three axioms, we also have that $d(x,y) \geq 0$ for any $x,y \in M$. This is deduced as follows: https://en.wikipedia.org/wiki/Metric_space

Goals for this lecture

- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**

- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.

k-medoids
PAM (Partitioning Around (k) Medoids)
k-means

- 1. Starts from a matrix of p features measured on a set of n observations.
- 2. Randomly pick k distinct cluster centers out of the n observations ("seeds").
- 3. **Assign each of the** remaining **observations** to the group to whose center it is the closest.
- 4. For each group, choose a new center from the observations in the group, such that the sum of the distances of group members to the center is minimal; this is called the *medoid*.
- 5. Repeat Steps 3 and 4 until the groups stabilize.



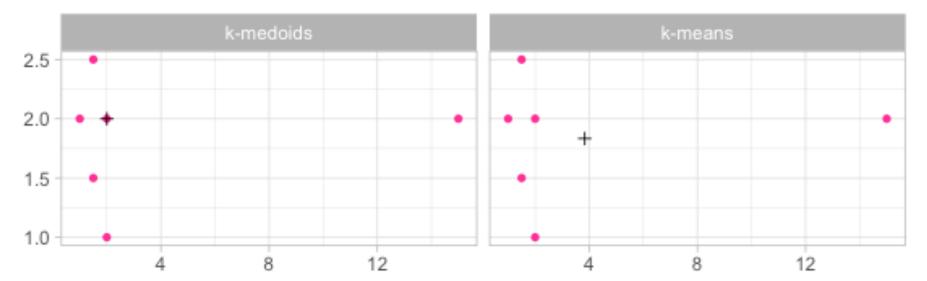
The difference between k-medoids and k-means is at step 4, when choosing the new centers

k-medoids

The new centers are selected among the observations

k-means

The new centers are **computed** as the **arithmetic mean** between the observations of each group



```
Xmat=matrix(runif(200),ncol=2)
nk=3
cents=matrix(runif(2*nk),ncol=2)

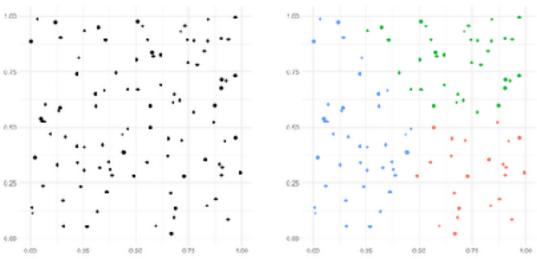
out=kmeans(Xmat,centers = nk) # give the # of clusters
out=kmeans(Xmat,centers = cents) # give the seeds (initial center of clusters)

X = data.frame(x = Xmat[,1], y = Xmat[,2], cluster = factor(out$cluster))

g = ggplot(X, aes(x = x, y = y)) + xlab("") + ylab("") + guides(col = FALSE)

g + geom_point()

g + geom_point(aes(col = cluster))
```



out1=kmeans(Xmat,cents,iter.max=1)
out2=kmeans(Xmat,cents,iter.max=2)
out3=kmeans(Xmat,cents,iter.max=3)

- 1. Starts from a matrix of p features measured on a set of n observations.
- 2. Randomly pick k distinct cluster centers out of the n observations ("seeds").
- 3. **Assign each of the** remaining **observations** to the group to whose center it is the closest.
- 4. For each group, choose a new center from the observations in the group, such that the sum of the distances of group members to the center is minimal; this is called the *medoid*.
- 5. Repeat Steps 3 and 4 until the groups stabilize.

Is the method robust? *i.e.* does it give always the same result?

How can we improve that?



Strong Forms & Tight Clusters

Dynamical Clusters

Repeats the process many times with **different seeds** (initial centers) to build `strong forms' which are groups of observations that end up in the same classes for most possible initial configurations.

Diday and Brito, 1989

Resampling

Repeats the process many times on subsamples of the dataset to create `tight clusters' which are groups of observations that are almost always grouped together.

Tseng and Wong, 2015

Data preprocessing matters

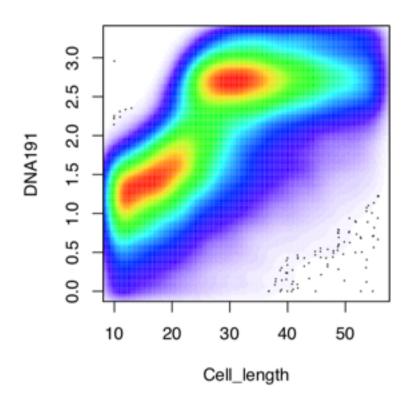
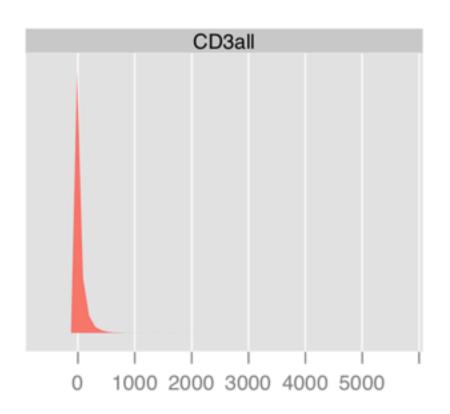


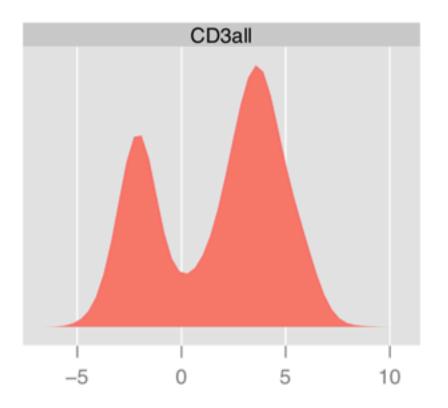
Figure 5.11: Cell measurements that show clear clustering in two dimensions.

Different scales

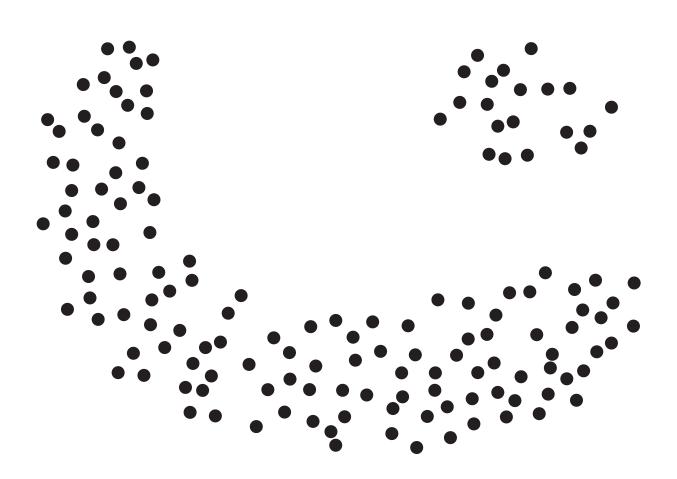
Raw

asinh





Density-based clustering



Density-based clustering

2 parameters:

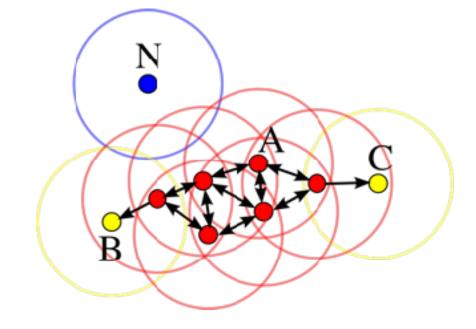
Eps (ε) : the maximum distance between 2 points so that they are considered as "reachable" from one another

minPts: the minimum number of points for a cluster to be created.

It classifies each data-point into

- core points
- directly reachable points
- noise / outliers

Algorithm: **DBSCAN D**ensity-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise



Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu,1996

How does density-based clustering (dbscan) work?

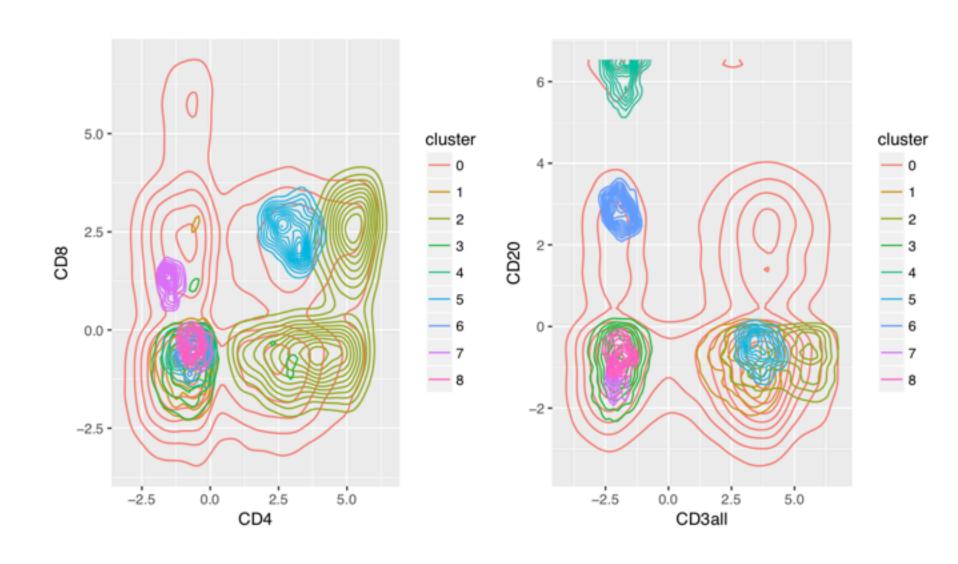
The dbscan method clusters points in dense regions according to the **density-connectedness** criterion. It looks at small neighborhood spheres of radius ϵ to see if points are connected.

The building block of dbscan is the concept of density-reachability: a point q is directly **density-reachable** from a point p if it is not further away than a given threshold ϵ , and if p is surrounded by sufficiently many points such that one may consider p (and q) be part of a dense region. We say that q is **density-reachable** from p if there is a sequence of points p_1, \ldots, p_n with $p_1 = p$ and $p_n = q$, so that each p_{i+1} is directly density-reachable from p_i .

A **cluster** is then a subset of points that satisfy the following properties:

- 1. All points within the cluster are mutually density-connected.
- If a point is density-connected to any point of the cluster, it is part of the cluster as well.
- 3. Groups of points must have at least MinPts points to count as a cluster.

Density-based clustering

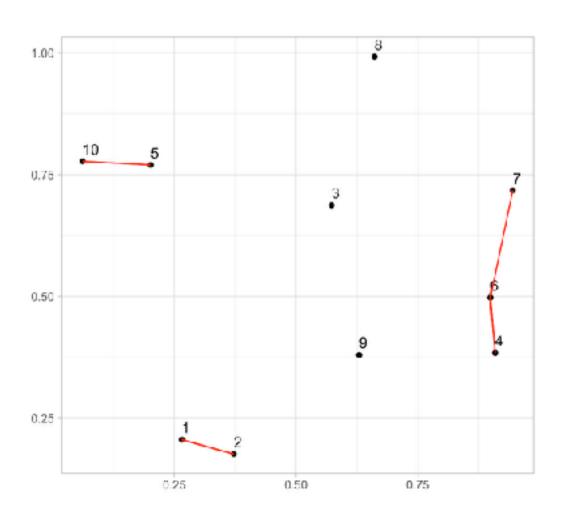


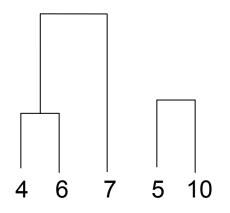
Goals for this lecture

- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**

- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.

Hierarchical Clustering (bottom-up)





Hierarchical Clustering (bottom-up)

```
set.seed(1)
Xmat=matrix(runif(20),ncol=2)
d = dist(Xmat)
```

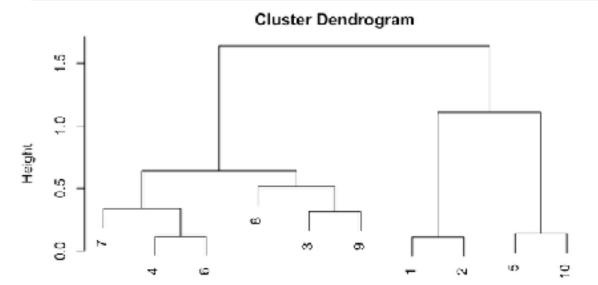
```
1.00 8 0.75 0.50 0.75
```

```
round(d,3)
                 3
 0.111
 0.571 0.549
 0.667
       0.575 0.452
 0.567
       0.617 0.380 0.805
       0.617
             0.377
       0.788
                   0.336 0.745 0.225
             0.373
       0.865
             0.317 0.656 0.510
                                0.548 0.395
                          0.578
                    0.279
                                0.294
             0.519 0.933 0.140 0.882 0.885
       0.676
                                             0.636 0.693
```

Hierarchical Clustering (bottom-up)

```
set.seed(1)
Xmat=matrix(runif(20),ncol=2)
d = dist(Xmat)
```

```
clust = hclust(d, method = "ward.D")
plot(clust)
```

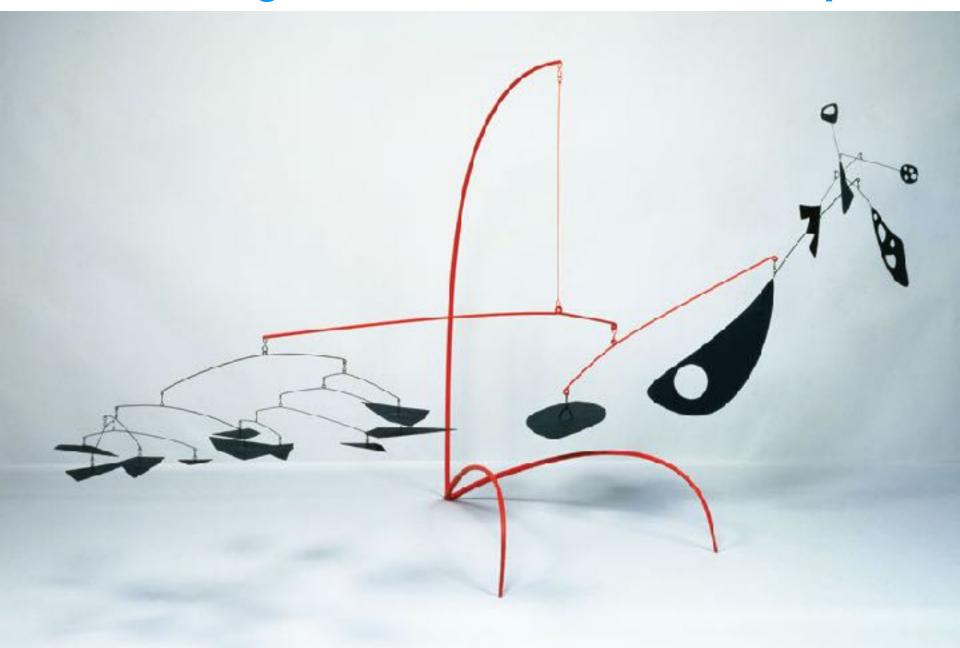


Goals for this lecture

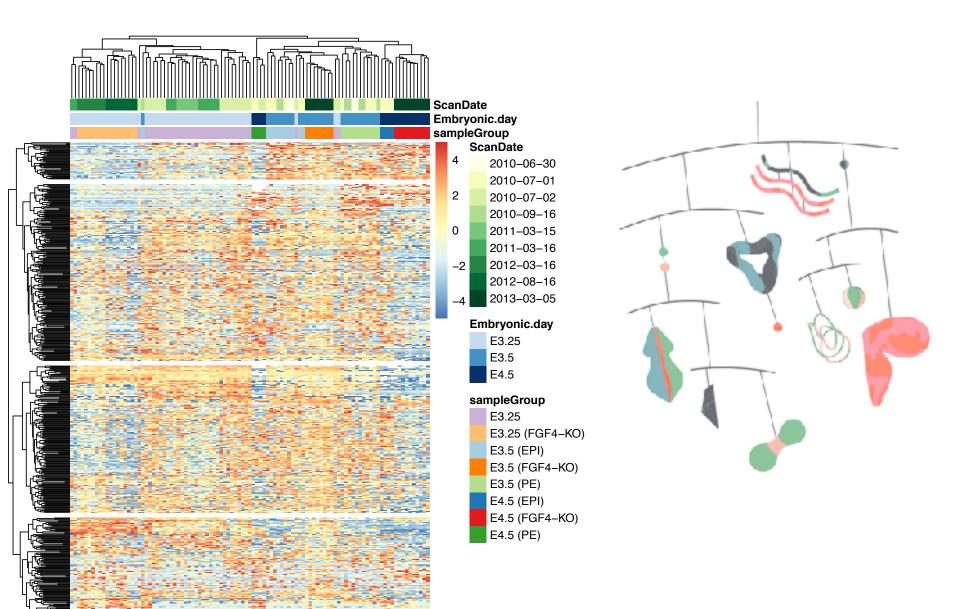
- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**

- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.

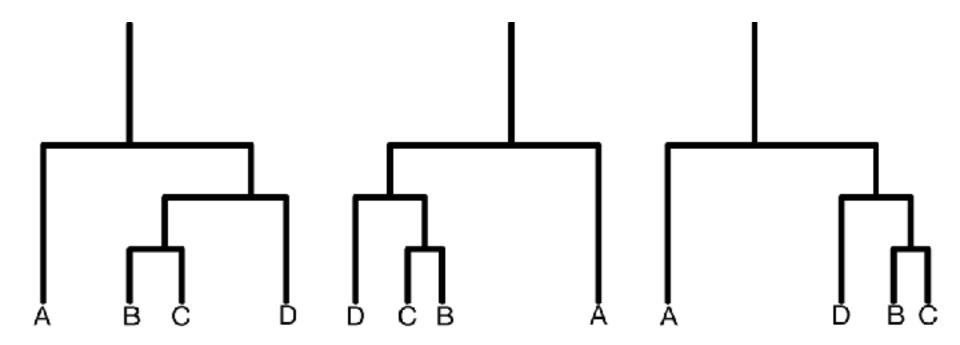
The ordering of the branches is not unique



The order of dendrogram branches is not unique



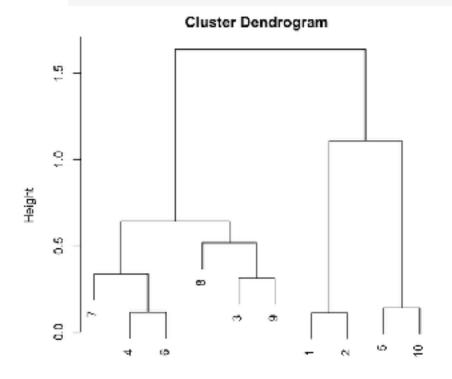
The ordering of the branches is not unique

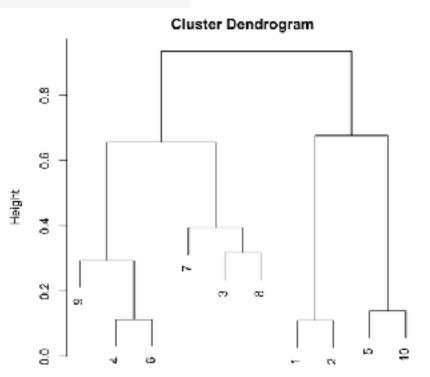


What matters is the "order of the agglomerations" and the length between leaves

Different options to agglomerate

```
clust = hclust(d, method = "ward.D")
plot(clust)
```



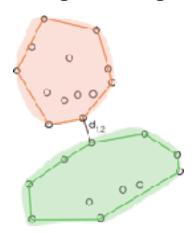


d hcluat (*, "ward.D")

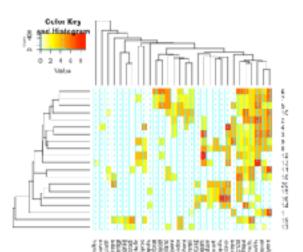
holust (*, "complete")

Different options to agglomerate

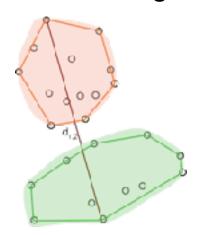
Single linkage



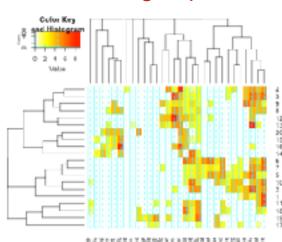
Good for recognizing the number of clusters
But "combs"



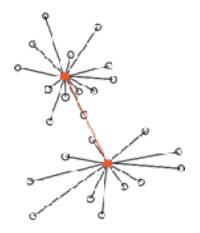
Maximal linkage



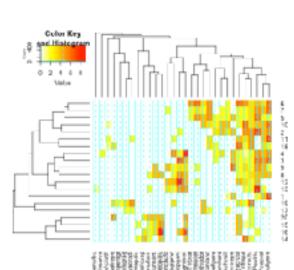
Compact classes
But one observation
can alter groups



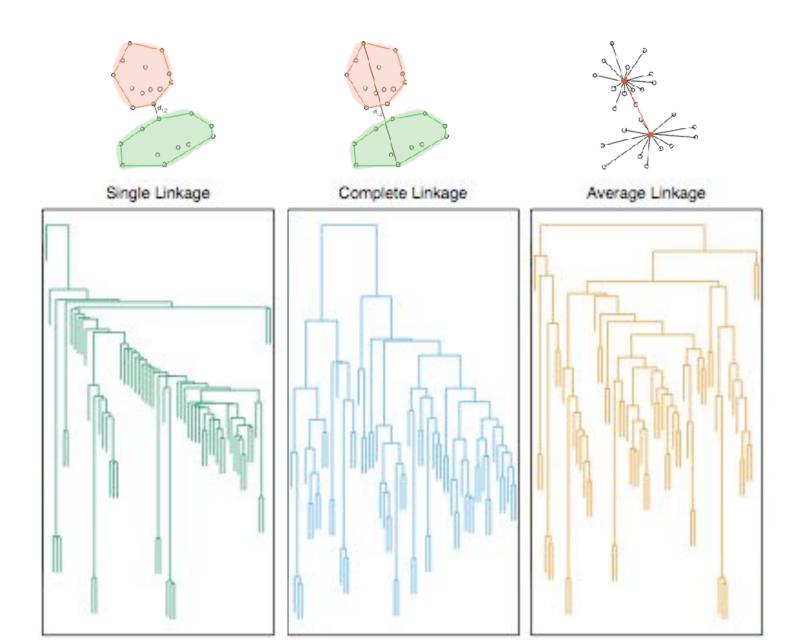
Medoids



More robust to outliers



Different options lead to different tree shapes

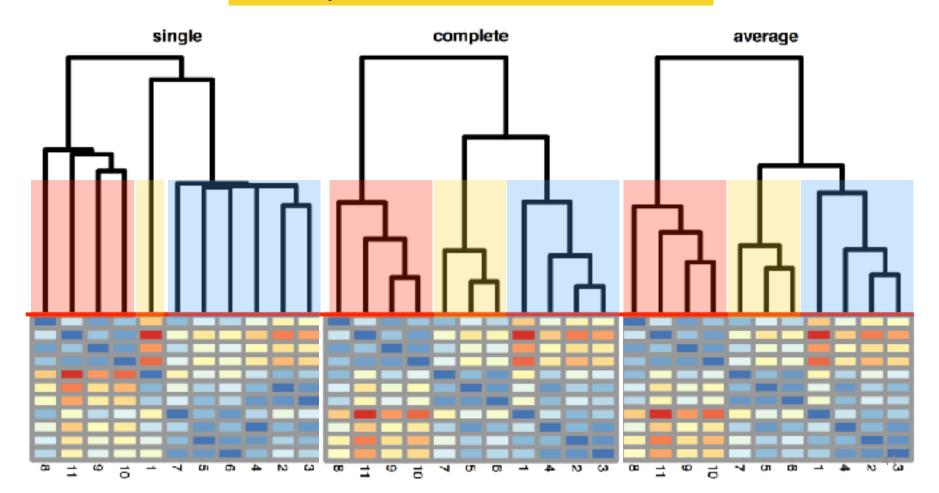


of clusters <-> max dist between groups

You cut the tree such that

- you have a given number of cluster
- at a given height (distance between the groups)

How do you choose the number of clusters?



Goals for this lecture

- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**

- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.



Clustering methods WILL find clusters ... even if there are none

How do we validate the clusters?

How do we find the optimal number of cluster?

We want to

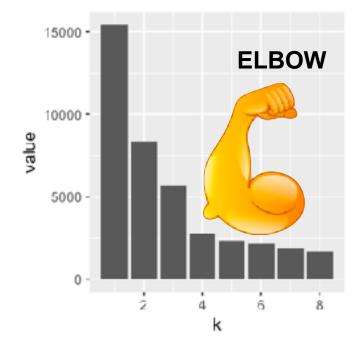
Minimize the distance between the points of a cluster (cohesion), **Maximizing** the distance between clusters (separation).

There are different ways to assess this.

- > WSS (Within-group sum of squares)
- Elbow method
- Calinski-Harabasz index
- Gap statistic
- Silhouette method

WSS

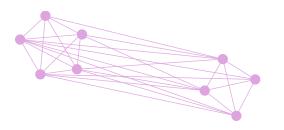
$$\mathrm{WSS}_k = \sum_{\ell=1}^k \sum_{x_i \in C_\ell} d^2(x_i, ar{x}_\ell)$$



Data

•

2 clusters



3 clusters









Calinski-Harabasz index

We want to maximize

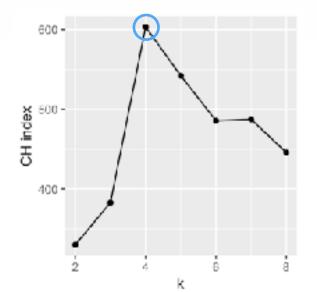
$$\operatorname{CH}(k) = \frac{\text{Between-groups sum of squares}}{\text{Within-groups sum of squares}}$$

We want to maximize

We want to minimize

$$ext{CH}(k) = rac{ ext{BSS}_k}{ ext{WSS}_k} imes rac{N-k}{N-1}$$

$$ext{where} \quad ext{BSS}_k = \sum_{\ell=1}^k n_\ell (ar{x}_\ell - ar{x})^2,$$



where \bar{x} is the overall center of mass (average point).

The Gap statistic

Algorithm for computing the gap statistic (Tibshirani, Walther, and Hastie 2001):

- Cluster the data with k clusters and compute WSS_k for the various choices of k.
- 2. Generate B plausible reference data sets, using Monte Carlo sampling from a homogeneous distribution and redo Step 1 above for these new simulated data. This results in B new within-sum-of-squares for simulated data W_{ab}^* , for $b=1,\ldots,B$.
- Compute the gap(k)-statistic:

$$\operatorname{gap}(k) = l_k - \log \operatorname{WSS}_k$$
 with $l_k = \frac{1}{B} \sum_{k=1}^{B} \log W_{kk}^*$

Note that the first term is expected to be bigger than the second one if the clustering is good (i.e., the WSS is smaller); thus the gap statistic will be mostly positive and we are looking for its highest value.

We can use the standard deviation.

$$\operatorname{sd}_k^2 = \frac{1}{B-1} \sum_{k=1}^B \left(\log(W_{kk}^*) - \bar{l}_k \right)^2$$

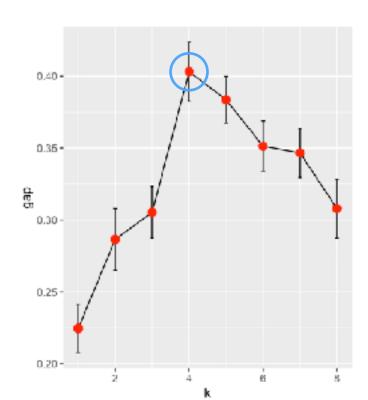
to help choose the best k. Several choices are available, for instance, to choose the smallest k such that

$$\operatorname{gap}(k) \geq \operatorname{gap}(k+1) - s'_{k+1} \qquad \text{where } s'_{k+1} = \operatorname{sd}_{k+1} \sqrt{1 + 1/B}.$$

The packages cluster and clusterCrit provide implementations.



Monte Carlo



The Silhouette method

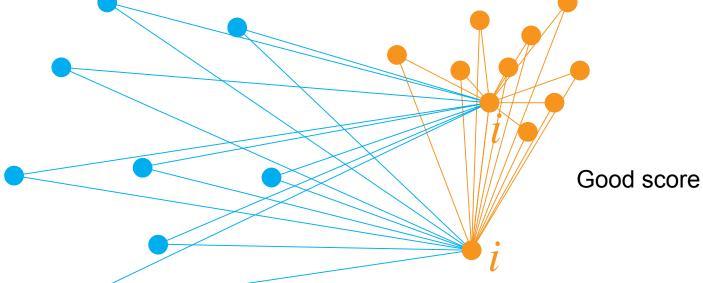
For each datapoint we compare

- the average distance to all the points of its own cluster
- the average distance to all the points of the nearest cluster

$$a_i = \text{average of}$$
 $b_i = \text{average of}$

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

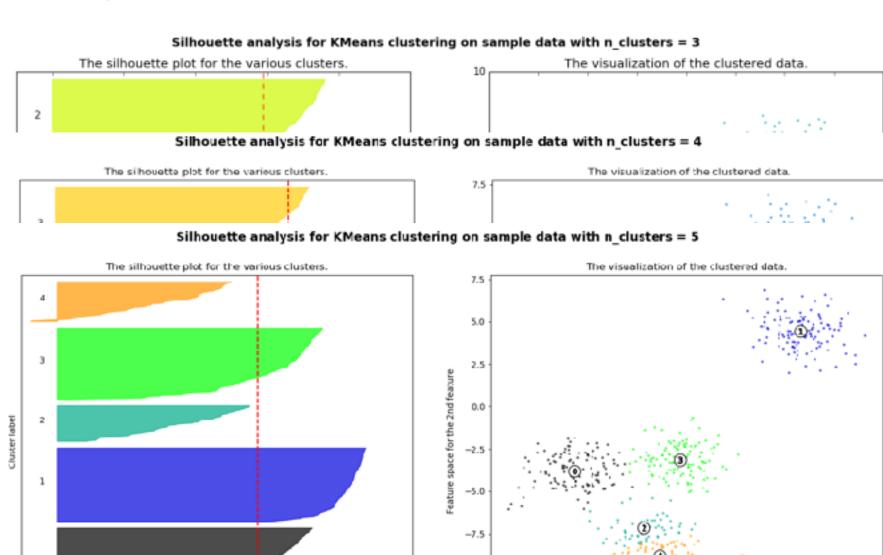
$$-1 \le s_i \le 1$$



Bad score:

Not very well integrated in the cluster 48

The Silhouette method



-10.0

Feature space for the 1st feature

1.0

0.8

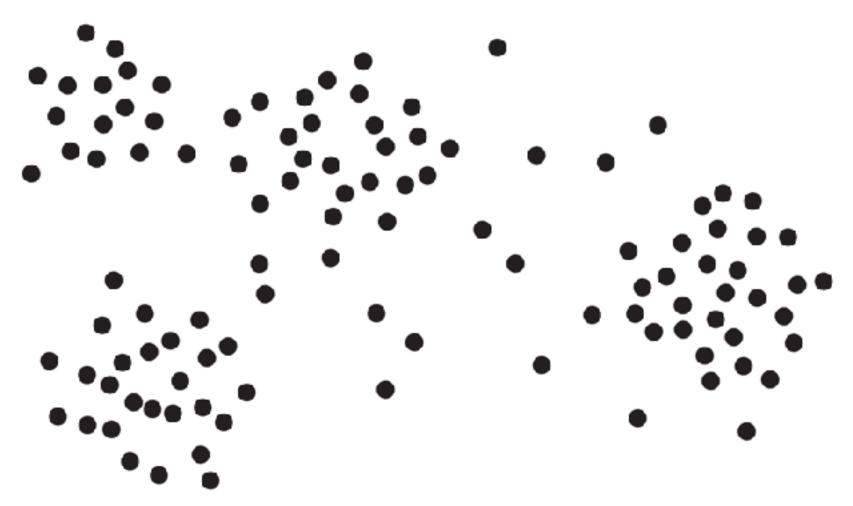
0.4

The silhouette coefficient values

-0.1

The bootstrap to validate clusters

How do we apply the bootstrap?

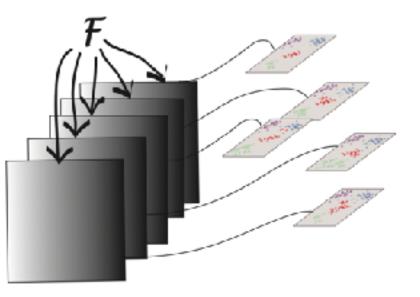


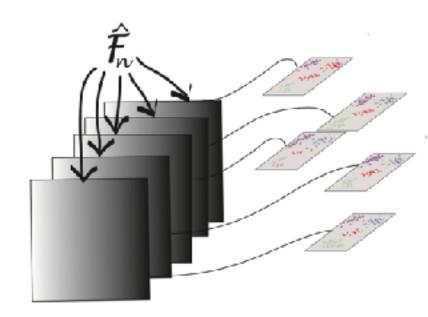
We create new datasets by subsampling the original dataset and evaluate how clusters are conserved.

Bootstrap for Clustering methods



Monte Carlo





Goals for this lecture

- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**

- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.

Some extra notes

No need to invent your own clustering

There are already too many!

https://cran.r-project.org/web/views/Cluster.html

THE THE THE THE COURSE AMERICA STREET

Management Transfers Corects and Better Cores.

Contract Station Communication of

Service: 2019-01-29

This WAY Tass Van contains a list of publicage that can be used in finding groups indial and modeling underwest cross sectional policy with the Ward Inscirum that must disk begins listed only more a quick starting wints rather than another than another

Riccordinal Contribute

- Participation of the Participation of the Contract Cont
- Yurkige and Test acceptate welfads for detection of desters in Resultsical clustering destroyants.

Next of the produporalistics in this CLANTIAN New Instruct of any distributed under the CP., Please have a last the DESCRIPTION for of such trackage to check under which hower it is distributed.

- of durining dynamics with linkage electro witch is a value of the single linkage excited on

- Indicated in information of the contract of the contract

- a Vanciere, use: from pudage large informate partitioning around resistance control withouthings formace. Function (a.e.g.) in wapperto pace; or larger into set. Silicontro plots and quanting eligner control and five vinalization.

 1 Sudage and interpretation of the pudage are analyses or the Mariato colo printed by Frey and Danik.

- Package and the allow to could for be optima classorie; provider for a given detaset.
- Subage and Salam implement though is protopic extension of London for windings date.

 Nakage and improves valva classical algorithm fad produce contal potition, i.e., a set of freepiter Badic man function regressing to membrably of erjective clusters.
- a franksije 🚾 primisis is commit cinnor significanciar primary sistema micropri, sistema propriore famility, manul par and Q* cinnoring. Proglement graphs and integer, form of personn par proglement or construction in the of sinch content of the cinnoring franks primary in the the cinnoring franks primary i
- Vactors of new idea; weighted knowl vegin; of he is town admirther by manuac and sterioral distortion by many
 Package of periods is nown classific specifically for interiorabil (simples).

Mode-Basel Clustering:

#II. minution i rescensi or person sources protects, whether apart or the observation abstract given sets on the support of the control of the observation abstract given to the support of the control of the support of the control of th

- Different control TM distributes and account of face in internal state of the control of the c

- Suggest Section for the incidence with Common mixture moved to high disconnected due where it is assumed that the data incomine is been objected upon.

 The common mixture is all the contract of the common mixture models being a classrooper of com Recept that the recept of the second of the

- a single-profile thing will be Midwales for passaring and an apparating an apparatin

- Planage Mark Transfer personations are included in Construction and discharacters and another in product and discharacters are in the Mark Construction and the Secondary of the the Secondary of

- The procession of the control of management of the control of management of procession of the control of management of the control of management of the control of the cont
- a Physical Security Street, Security religion using SACI. a Parage has allows Repeate clottering using aspite ani shift bienethiad maled and insuitable for clustering high dimensional lists.

 - Parage has provided Repeater clottering using aspite ani shift bienethiad maled and insuitable for clustering high dimensional lists.

 - Pacage provide Byonia estrutes of tale mixture of salvante famou an armal destitation.
- For the contract of the contra
- Process Conference of the Conference of the
- Parage Mail is a safego forgrafio agencias, obias is albeitable means Reposts observing where is a Parage provides as intellar to the IACH MCMC Meany eliabine lades a month for minimum mobility.

- or constant network

 Through Shift allows to fit an adjutive; nitrout of Student i distribution to approximate around related to be the Shift Sh

- Nedage and the real electricity distinctional data found in a tree disconnectable time gain. This is because any process attention of a means and applicament to binarchical clustering.
 Nedage are previous extensive application contracts because on the electronic tree.

- Yackage 🗫 instruments a chainer algorithm that in same on expain functions and fluctions and fluctions affected a group discoverations according to the multivariate degradates structure of the generating process without any assumptions carries margins.
- Variage and be provide complimentary biserchied distance which was opening uniqued for misrource data to account reaction proves in the data that arise from 'walk' grass a Tackage and provides a line complementation or the 2000 AN intensity transleps of applications with securi approxime using a lab row.

If you have VERY large datasets

Computing the distance between each pair of datapoint is NOT reasonable

Function clara() is a wrapper to pam() for larger data sets.

Partitioning method (k-medoids)

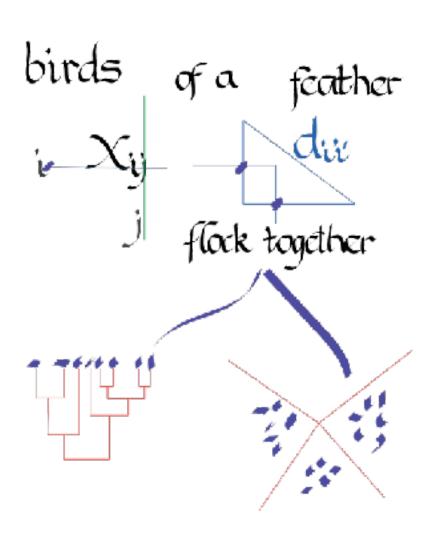
It uses **subsamples** of the original dataset, looks for **conserved clusters**, Finds their **medoids**, Compute the **distance of the remaining data points** to the medoids, **Attribute a cluster** to each datapoint.

Goals for this lecture

- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain non-parametric methods such as k-means or k-medoids (also mentioned density based clustering: however cursed)
- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.

Summary

Summary: the clustering workflow



- Start with the data.
 What are the measurements?
 What type of data?
- 2. Define or select a **metric** (**distance**) to evaluate the (dis)similarity between 2 samples.
- 3. Choose a **clustering method**: bottom-up (*hierarchical*) or top-down (*k-methods*)
- 4. **Validate** the clustering and evaluate the **optimal number** of clusters
- 5. **Augment** your data with the cluster information

Distances

We saw at the start of the Lecture how finding the **right** distance is an essential first step in a clustering analysis. This is a case when the **garbage in, garbage out** motto is in full force. Always choose a distance which is **scientifically meaningful** and compare output from as many distances as possible. Sometimes the same data require **different distances** when **different scientific objectives** are pursued.

Partitioning and Aggregating

We saw two different types of clustering approaches:

iterative partitioning approaches such as kmeans and kmedoids (PAM) that alternated between estimating the clusters and assigning points to them and **hierarchical clustering** approaches that agglomerate points and then small clusters into larger ones in a nested sequence of sets that can be represented by hierarchical clustering trees.

Cluster validation

Clustering algorithms *always* deliver clusters so we need to assess their quality and the number of clusters to choose carefully.

These validation steps are done using visualization tools and repeating the clustering on many resamples of the data. We saw how statistics such as the bss/wss or log(wss) can be calibrated using simulation on data where we understand the group structure and can provides useful benchmarks for choosing the number of cluster.

The choice or definition of distance depends on the data

Euclidean distance (L2)

$$d(A,B) = \sqrt{(a_1-b_1)^2 + (a_2-b_2)^2 + \ldots + (a_p-b_p)^2}.$$

Manhattan distance (L1)

$$d(A,B) = |a_1 - b_1| + |a_2 - b_2| + \ldots + |a_p - b_p|.$$

Maximum distance (L∞)

$$d_{\infty}(A, B) = \max_{i} |a_i - b_i|.$$

Minkowski distance (Lm)

$$d(A,B) = ((a_1-b_1)^m + (a_2-b_2)^m + \ldots + (a_p-b_p)^m)^{rac{1}{m}}.$$

Edit (Hamming) distance

Binary distance

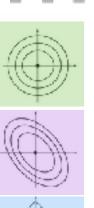
$$d_J(S,T) = 1 - J(S,T) = rac{f_{01} + f_{10}}{f_{01} + f_{10} + f_{11}}$$
 .

Correlation-based distance

$$d(A,B) = \sqrt{2(1-\operatorname{cor}(A,B))}.$$

Weighted Euclidean distance

Mahalanobis distance



Questions

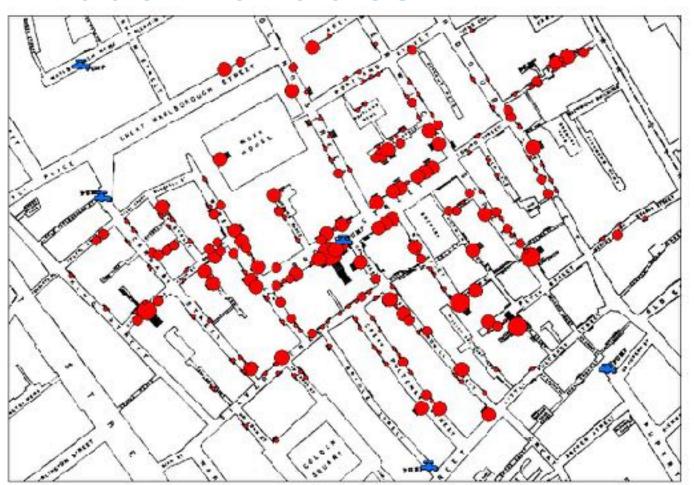


Goals for this lecture

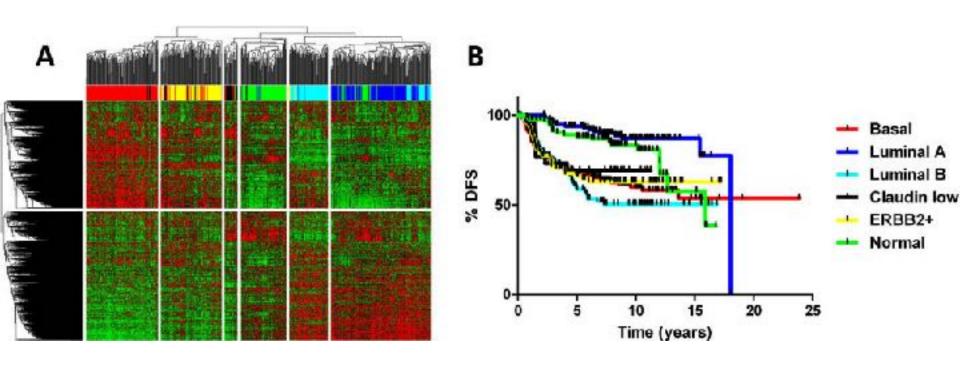
- 1. Review measures of (dis)similarity that help us define clusters.
- 2. Explain **non-parametric methods** such as **k-means** or **k-medoids**
- 3. Explain the **recursive approach** to clustering that combines observations and groups into a hierarchy of sets and called *hierarchical clustering*.
- 4. Understand how to **validate the clusters** and select the **optimal number of clusters**.
- 5. Examples in R lab is important.
- 6. Why do we cluster?

Why do we cluster?

- Hidden variables:



Why do we cluster?



Why do we cluster?

- Hidden variables:



