BIOINFORMATICS

How can we cluster biological data?

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

May 2019

Outline

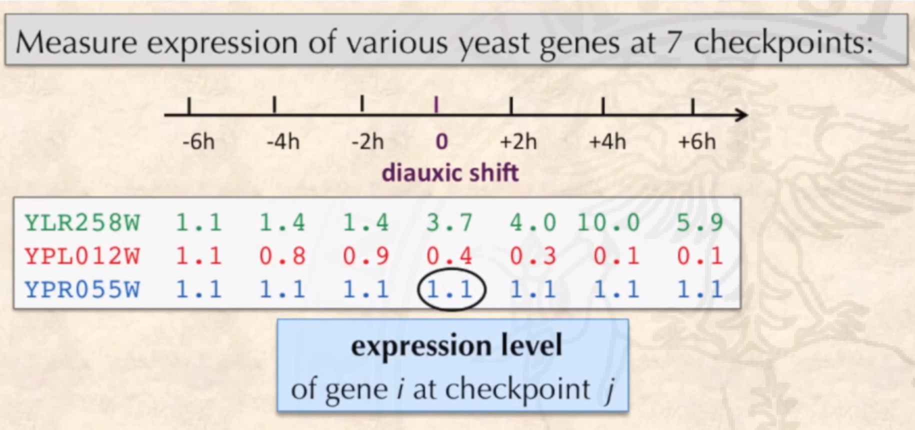
1. Gene expression and clustering;
2. Clustering as optimization problem;
3. K-means Clustering;
4. Hierarchical Clustering;

Chapter 8 in ***Bioinformatics Algorithms: An active Learning Approach (Vol.2).***

Part 1

Gene expression and clustering

Gene expression and clustering



Gene expression and clustering

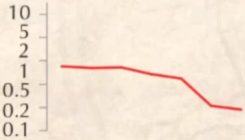
Measure expression of various yeast genes at 7 checkpoints:

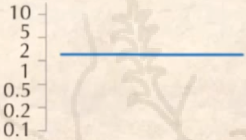
-6h -4h -2h 0 +2h +4h +6h

diauxic shift

YLR258W	1.1	1.4	1.4	3.7	4.0	10.0	5.9
YPL012W	1.1	0.8	0.9	0.4	0.3	0.1	0.1
YPR055W	1.1	1.1	1.1	1.1	1.1	1.1	1.1

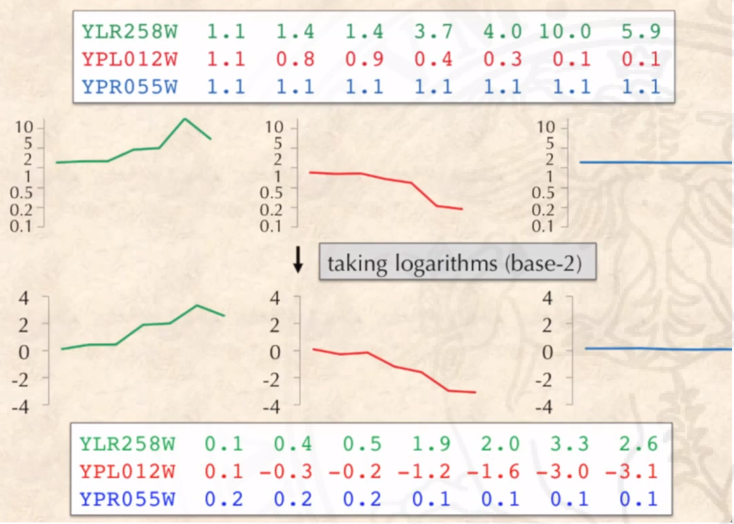
expression level
of gene *i* at checkpoint *j*





Gene expression and clustering

- Switching to Logarithms of Expression Level

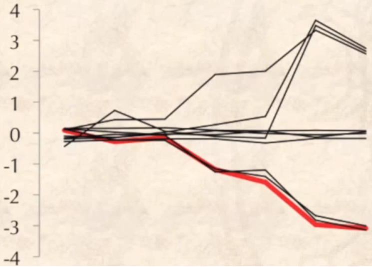


Gene expression and clustering

- Gene expression matrix

YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07

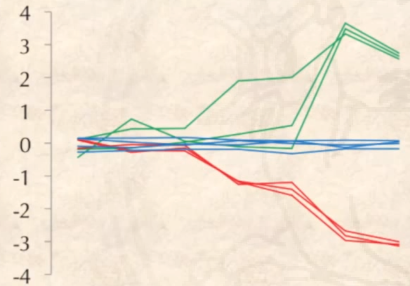
gene expression vector



Gene expression and clustering

- Gene expression matrix

YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07



Gene expression and clustering

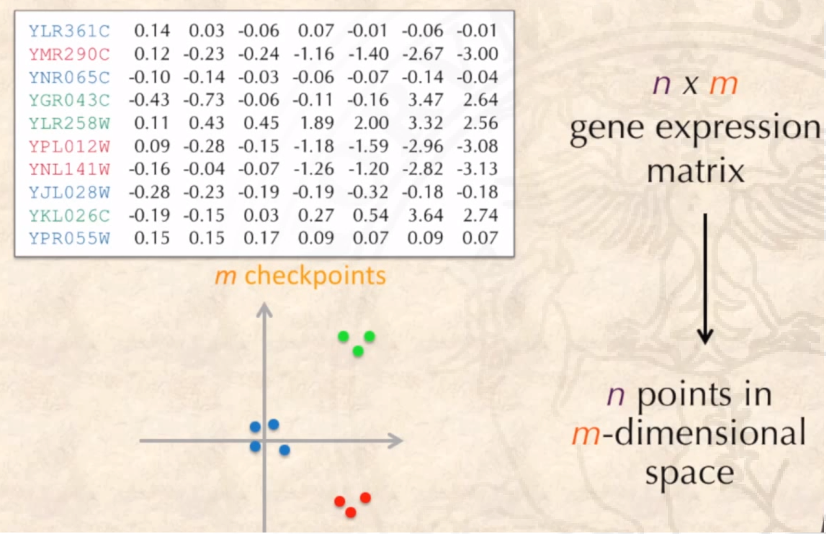
- In 1997 Joseph deRisi measured expression of 6,400 yeast genes at 7 checkpoints before and after diauxic shift;
- Expression matrix with 6,400 x 7;
- **Goal:** partition all yeast genes into clusters so that:
  - genes in the same cluster have similar behavior;
  - gene in different clusters have different behavior.

Gene expression and clustering

- In 1999 Uri Alon measured expression of 2,000 genes from 40 colon tumor patients and 40 healthy people;
- Two Expression matrices with dimension 2,000 x 40;
- **Goal:** find genes with significantly different expression vectors between tumor patients and healthy (potential cancer bio-markers)

Gene expression and clustering

- Gene as Points in a Multidimensional Space



YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07

Gene expression and clustering

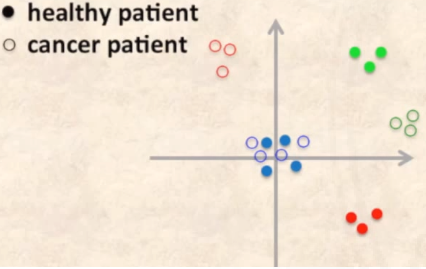
- Gene as Points in a Multidimensional Space

YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07

n x *m* gene expression matrix

↓

n points in *m*-dimensional space



Gene expression as a Cancer Biomarker

MammaPrint: a test that evaluates the likelihood of breast cancer recurrence based on the expression of just 70 genes.

- **but how did scientists discover these 70 human genes?**

Part 1

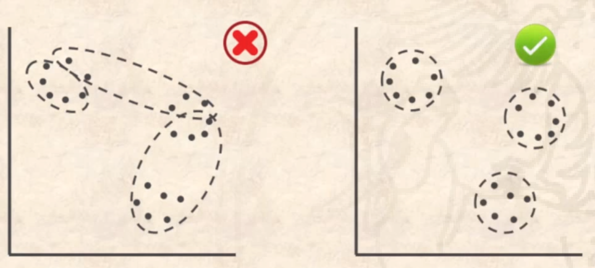
Clustering as optimization problem

Clustering as optimization problem

Toward a Computational Problem

Good Clustering Principle:

Elements within the same cluster should be closer to each other than elements in different clusters.



- we define a threshold Δ then:
  - distance between elements in the same cluster must be $\leq \Delta$;
  - distance between elements in different clusters must be $> \Delta$;

Clustering as optimization problem

Introducing the Clustering problem:

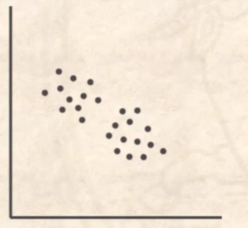
Clustering problem

Definition: find a partition of expression vector into clusters satisfying the **Good Clustering Principle**

Input: A collection of n vectors and an integer k.

Output: Partition of n vectors into k disjoint clusters satisfying the **Good Clustering Principle**

Can you find a partition into two clusters which is valid solution for the clustering problem?



Clustering as optimization problem

Introducing the Clustering problem:

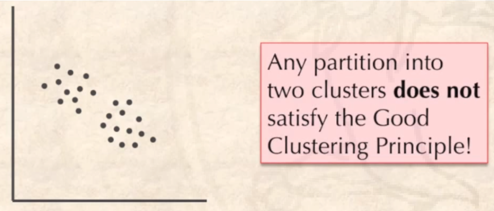
Clustering problem

Definition: find a partition of expression vector into clusters satisfying the **Good Clustering Principle**

Input: A collection of n vectors and an integer k.

Output: Partition of n vectors into k disjoint clusters satisfying the **Good Clustering Principle**

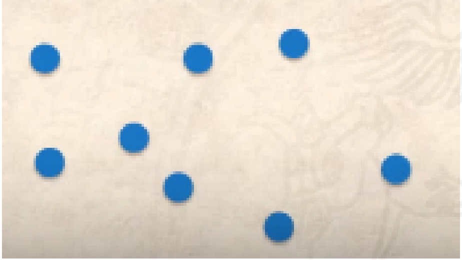
Can you find a partition into two clusters which is valid solution for the clustering problem?



Clustering as optimization problem

Clustering as Finding Centers

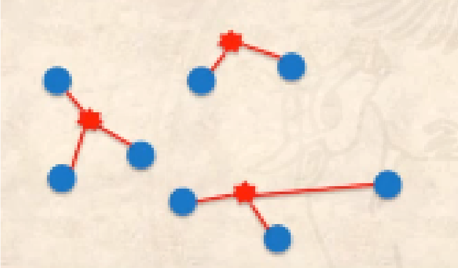
- **Goal:** partition a set ***Data*** into k clusters.



Clustering as optimization problem

Clustering as Finding Centers

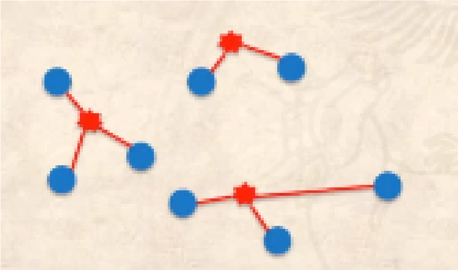
- **Goal:** partition a set ***Data*** into k clusters.
- **Equivalent goal:** find a set of k points ***Centers*** that will be the "centers" of the k clusters in ***Data***, and will minimize some notion of distance from ***Data*** to ***Centers***.



Clustering as optimization problem

Clustering as Finding Centers

- **Goal:** partition a set ***Data*** into k clusters.
- **Equivalent goal:** find a set of k points ***Centers*** that will be the "centers" of the k clusters in ***Data***, and will minimize some notion of distance from ***Data*** to ***Centers***.



What is the "distance" between Data and Centers

Clustering as optimization problem

Distance from a Single DataPoint to Centers

The distance from ***DataPoint*** in ***Data*** to ***Centers*** is defined as the distance from ***DataPoint*** to the closest center

$$d(\textit{DataPoint}, \textit{Centers}) = \min_{\forall \mathbf{x} \in \textit{Centers}} d(\textit{DataPoint}, \mathbf{x})$$

Clustering as optimization problem

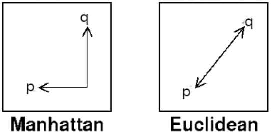
Distance from a Single DataPoint to Centers

- Different distance metrics can be used;
- The most used metrics are:
  - Euclidean distance:

$$d(p,q) = \sqrt{\sum_{i \in m} (p_i - q_i)^2}$$

  - Manhattan distance:

$$d(p,q) = \sum_{i \in m} |(p_i - q_i)|$$



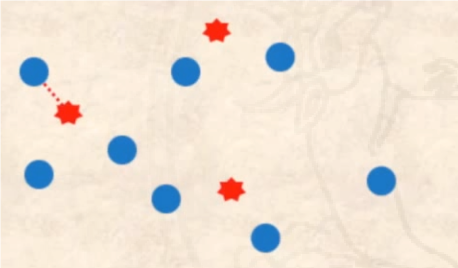
- hereafter we will use Euclidean distance, Manhattan distance works better in case of high dimensional vectors.

Clustering as optimization problem

Distance from a Single DataPoint to Centers

The distance from ***DataPoint*** in ***Data*** to ***Centers*** is defined as the distance from ***DataPoint*** to the closest center

$$d(\textbf{\textit{DataPoint}}, \textbf{\textit{Centers}}) = \min_{x \in \textbf{\textit{Centers}}} d(\textbf{\textit{DataPoint}}, x)$$

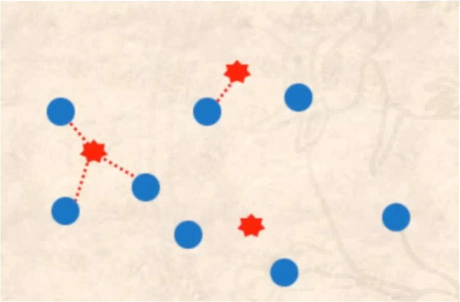


Clustering as optimization problem

Distance from a Single DataPoint to Centers

The distance from ***DataPoint*** in ***Data*** to ***Centers*** is defined as the distance from ***DataPoint*** to the closest center

$$d(\textbf{\textit{DataPoint}}, \textbf{\textit{Centers}}) = \min_{\forall \mathbf{x} \in \textbf{\textit{Centers}}} d(\textbf{\textit{DataPoint}}, \mathbf{x})$$

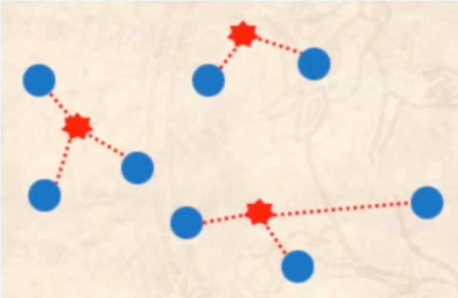


Clustering as optimization problem

Distance from a Single DataPoint to Centers

The distance from ***DataPoint*** in ***Data*** to ***Centers*** is defined as the distance from ***DataPoint*** to the closest center

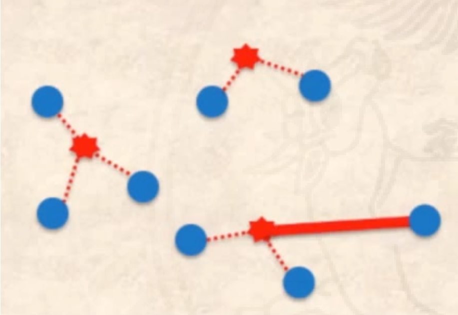
$$d(\textbf{\textit{DataPoint}}, \textbf{\textit{Centers}}) = \min_{\forall \mathbf{x} \in \textbf{\textit{Centers}}} d(\textbf{\textit{DataPoint}}, \mathbf{x})$$



Clustering as optimization problem

Distance from a Single DataPoint to Centers

$$MaxDistance(\textbf{Data}, \textbf{Centers}) = \max_{\forall \textbf{DataPoint} \in \textbf{Data}} d(\textbf{DataPoint}, \textbf{Centers})$$



Clustering as optimization problem

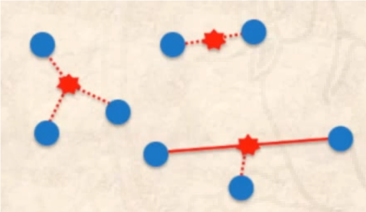
Introducing k-Center Clustering problem:

k-Center Clustering problem

Definition: Given a set of points **Data**, find k centers minimizing *MaxDistance(**Data**,**Centers**)*

Input: A collection of n vectors and an integer k.

Output: A set of k points ***Centers*** that minimizes *MaxDistance(**Data**,**Centers**)* over all possible choices of ***Centers***



Clustering as optimization problem

Introducing k-Center Clustering problem:

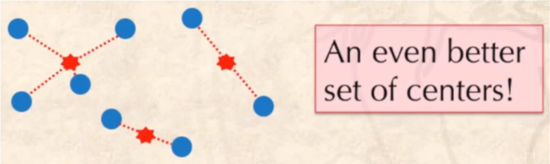
---

k-Center Clustering problem

Definition: Given a set of points **Data**, find k centers minimizing *MaxDistance(**Data**,**Centers**)*

Input: A collection of n vectors and an integer k.

Output: A set of k points ***Centers*** that minimizes *MaxDistance(**Data**,**Centers**)* over all possible choices of ***Centers***



Clustering as optimization problem

Introducing k-Center Clustering problem:

---

k-Center Clustering problem

Definition: Given a set of points **Data**, find k centers minimizing *MaxDistance(**Data**,**Centers**)*

Input: A collection of n vectors and an integer k.

Output: A set of k points ***Centers*** that minimizes *MaxDistance(**Data**,**Centers**)* over all possible choices of ***Centers***

- This problem is intractable;
- Since it is a hard problem → **approximation algorithms** were developed.

Clustering as optimization problem

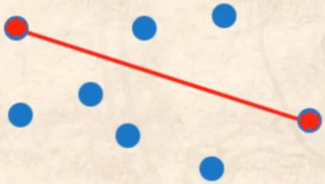
k-Center Clustering heuristic

FarthestFirstTraversal(*Data*, *k*)
  Centers ←select first center randomly.
  while *Centers* have fewer than *k* points
    DataPoint ←a point in *Data* maximizing *d*(*DataPoint*, *Centers*)
        among all data points
    add *DataPoint* to *Centers*

Clustering as optimization problem

k-Center Clustering heuristic

FarthestFirstTraversal(*Data*, *k*)
  Centers ← select first center randomly.
  while *Centers* have fewer than *k* points
    DataPoint ← a point in *Data* maximizing *d*(*DataPoint*, *Centers*)
          among all data points
    add *DataPoint* to *Centers*

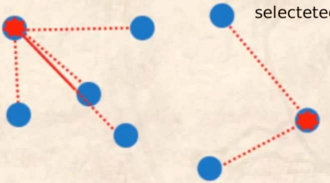


Clustering as optimization problem

k-Center Clustering heuristic

```
FarthestFirstTraversal(Data, k)
  Centers ←select first center randomly.
  while Centers have fewer than k points
    DataPoint ←a point in Data maximizing d(DataPoint, Centers)
                among all data points
    add DataPoint to Centers
```

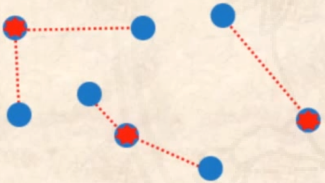
All blue nodes are split between the two center.
Then the most far for its center is selecteted as new center.



Clustering as optimization problem

k-Center Clustering heuristic

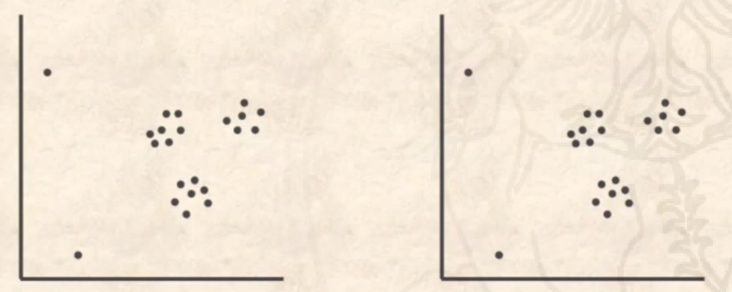
FarthestFirstTraversal(*Data*, *k*)
  Centers ←select first center randomly.
  while *Centers* have fewer than *k* points
    DataPoint ←a point in *Data* maximizing *d*(*DataPoint*, *Centers*)
            among all data points
    add *DataPoint* to *Centers*



Clustering as optimization problem

What is wrong with FarthestFirstTraversal?

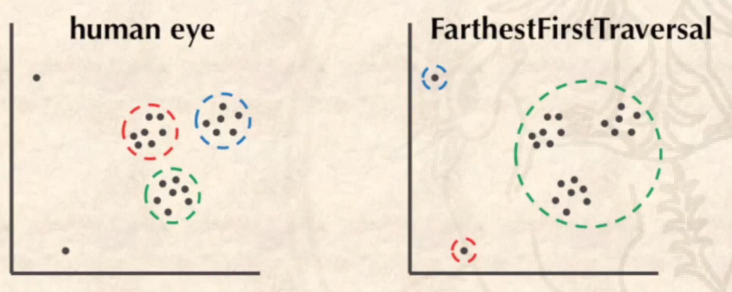
- FarthestFirstTraversal selectes **Centers** that minimize MaxDistance(**Data,Centers**).
- But biologists are interested in **typical** rather than **maximum** deviations: maximum deviations may represent **outliers** (experimental errors)



Clustering as optimization problem

What is wrong with FarthestFirstTraversal?

- FarthestFirstTraversal selectes **Centers** that minimize MaxDistance(**Data**,**Centers**).
- But biologists are interested in **typical** rather than **maximum** deviations: maximum deviations may represent **outliers** (experimental errors)



Clustering as optimization problem

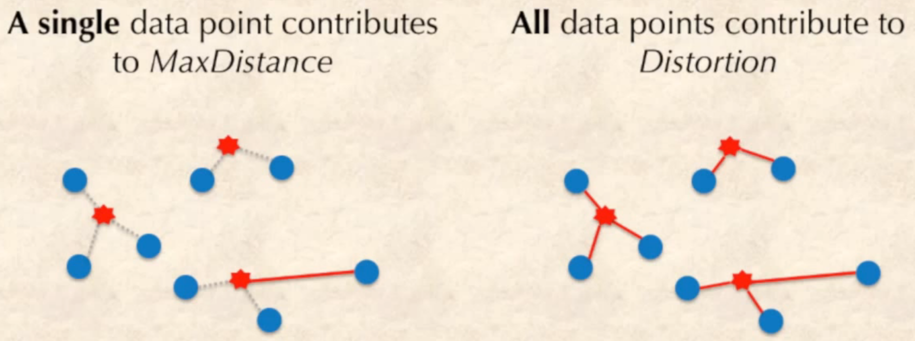
Modifying objection function

The **maximal distance** between *Data* and *Centers*:

$$MaxDistance(Data, Centers) = \max_{DataPoint \text{ from } Data} d(DataPoint, Centers)$$

The **squared error distortion** between *Data* and *Centers*:

$$Distortion(Data, Centers) = \sum_{DataPoint \text{ from } Data} d(DataPoint, Centers)^2/n$$



Clustering as optimization problem

***k*-Center Clustering Problem:**
Input: A set of points *Data* and an integer *k*.
Output: A set of *k* points *Centers* that minimizes

MaxDistance(DataPoints,Centers)

over all choices of *Centers*.

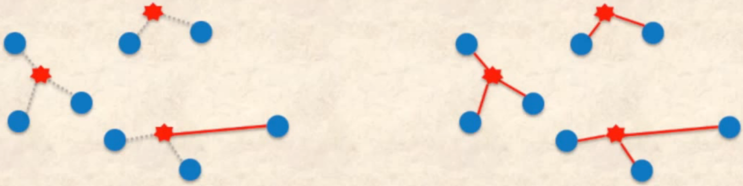
***k*-Means Clustering Problem:**
Input: A set of points *Data* and an integer *k*.
Output: A set of *k* points *Centers* that minimizes

Distortion(*Data*,*Centers*)

over all choices of *Centers*.

A single data point contributes to *MaxDistance*

All data points contribute to *Distortion*



Clustering as optimization problem

***k*-Center Clustering Problem:**
Input: A set of points *Data* and an integer *k*.
Output: A set of *k* points *Centers* that minimizes

MaxDistance(DataPoints,Centers)

over all choices of *Centers*.

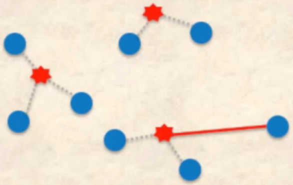
***k*-Means Clustering Problem:**
Input: A set of points *Data* and an integer *k*.
Output: A set of *k* points *Centers* that minimizes

Distortion(*Data*,*Centers*)

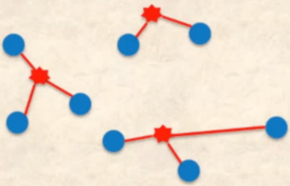
over all choices of *Centers*.

NP-Hard for k>1

A single data point contributes to *MaxDistance*



All data points contribute to *Distortion*



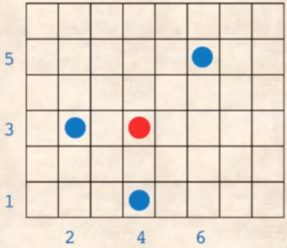
Clustering as optimization problem

k-Means Clustering Problem

Center of Gravity Theorem: The center of gravity of points *Data* is the only point solving the 1-Means Clustering Problem.

The **center of gravity** of points *Data* is

$\sum_{\text{all points } DataPoint \text{ in } Data} DataPoint$ / #points in *Data*

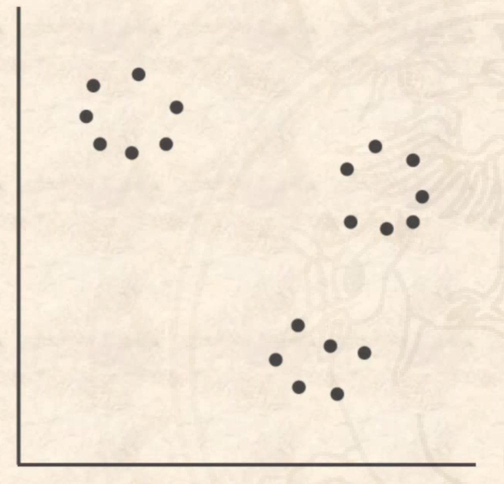


i-th coordinate of the center of gravity = the average of the *i*-th coordinates of datapoints:

((2+4+6)/3, (3+1+5)/3) = (4, 3)

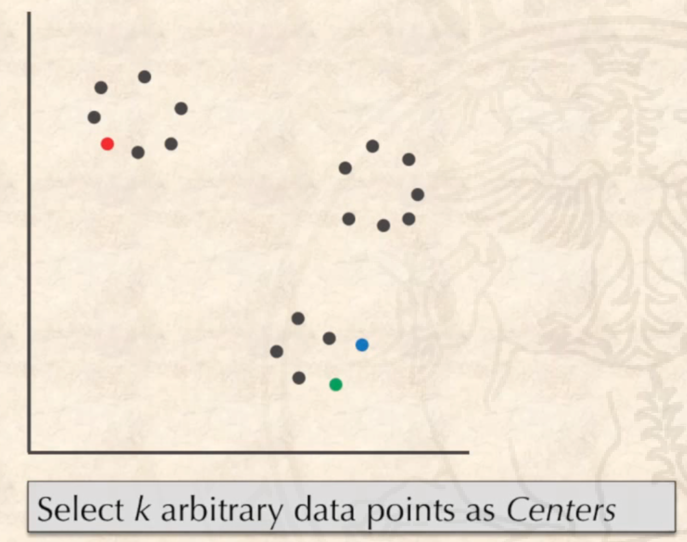
Clustering as optimization problem

***Lloyd approximation algorithm* for k-Means Clustering Problem**



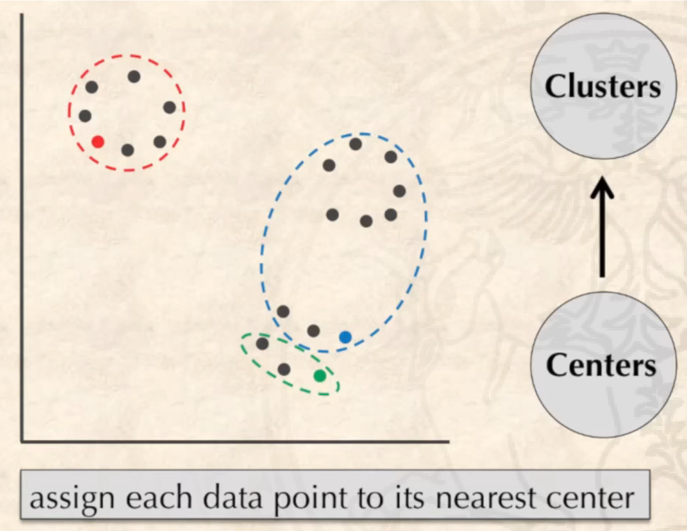
Clustering as optimization problem

***Lloyd approximation algorithm* for k-Means Clustering Problem**



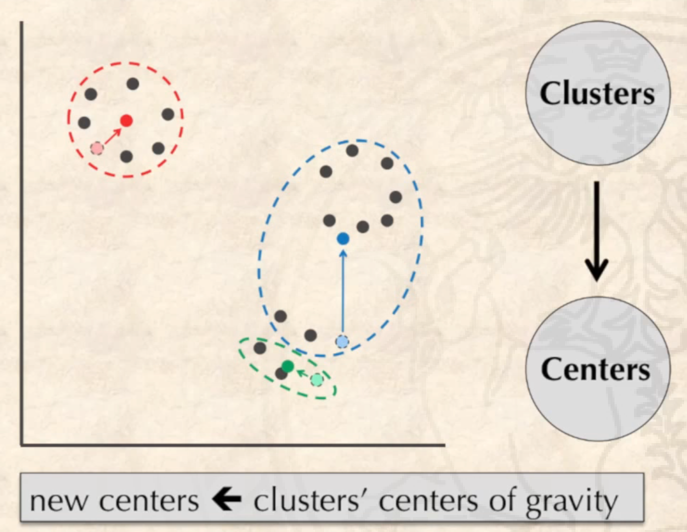
Clustering as optimization problem

***Lloyd approximation algorithm* for k-Means Clustering Problem**



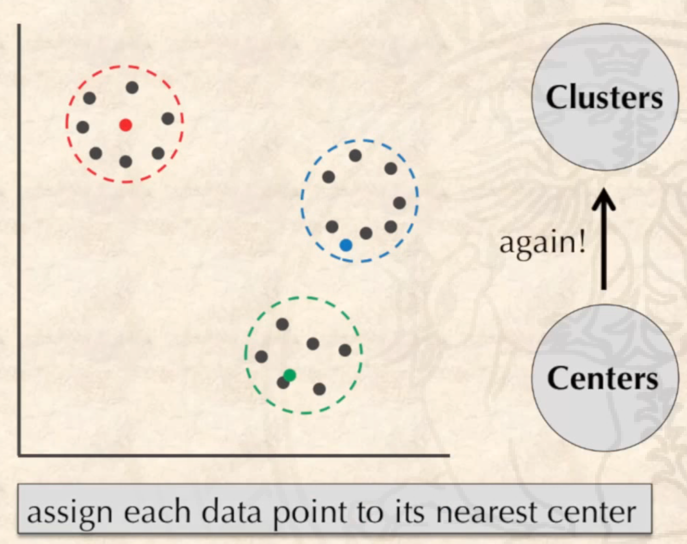
Clustering as optimization problem

***Lloyd approximation algorithm* for k-Means Clustering Problem**



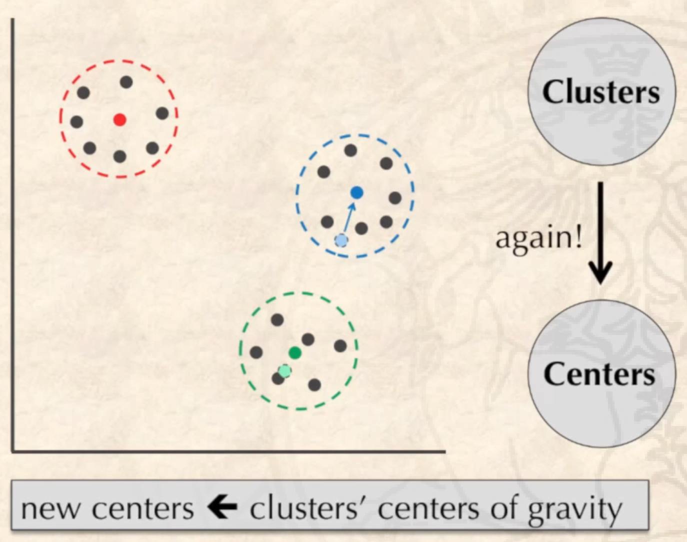
Clustering as optimization problem

Lloyd approximation algorithm **for k-Means Clustering Problem**



Clustering as optimization problem

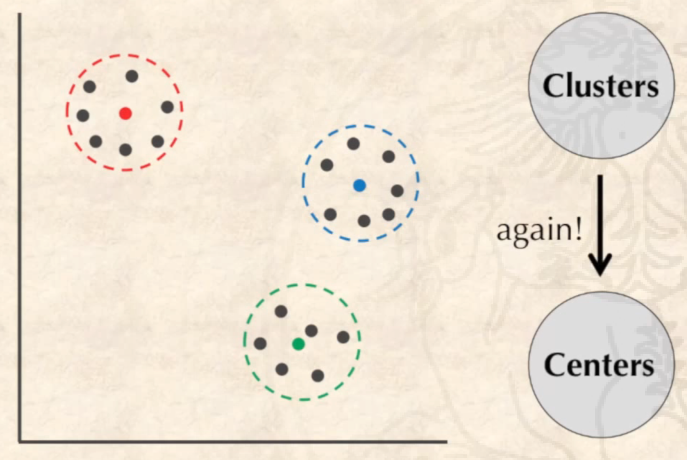
***Lloyd approximation algorithm* for k-Means Clustering Problem**



Clustering as optimization problem

***Lloyd approximation algorithm* for k-Means Clustering Problem**

It ends when the Centers stop to move.



Clustering as optimization problem

Lloyd algorithm

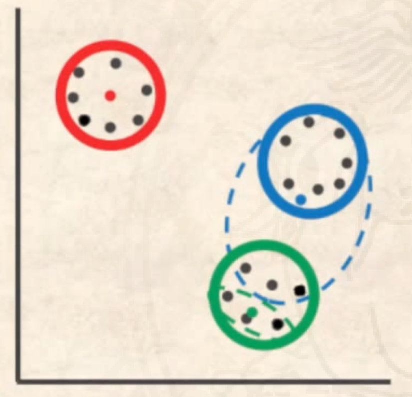
Select k arbitrary data points as *Centers* and then iteratively perform the following steps:

- **Centers to Clusters**: Assign each data point to the cluster corresponding to its nearest center (ties are broken arbitrarily).
- **Clusters to Centers**: After the assignment of data points to k clusters, compute new centers as clusters' center of gravity.

Clustering as optimization problem

Lloyd algorithm **converges!!!**

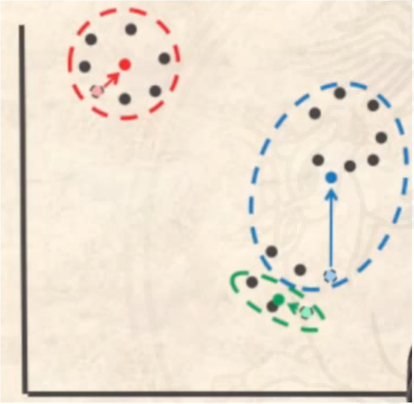
- if a data point is assigned to a new center during the ***Centers to Clusters*** step:
  - the squared error distortion is reduced because this center must be closed to the point than the previous center.



Clustering as optimization problem

Lloyd algorithm **converges!!!**

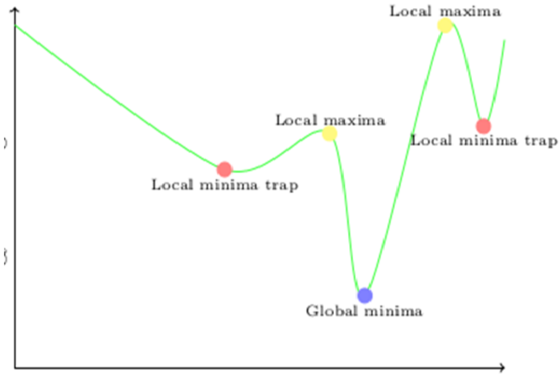
- if a data point is assigned to a new center during the ***Clusters to Centers*** step:
  - the squared error distortion is reduced because the center of gravity is the only point minimizing the distortion.



Clustering as optimization problem

Lloyd algorithm **converges!!!**

- It converges to **local minimum**. Thus several runs are required to discover the best solution;
- It can take time to converge.



How can we choose a “good" K for K-means clustering?

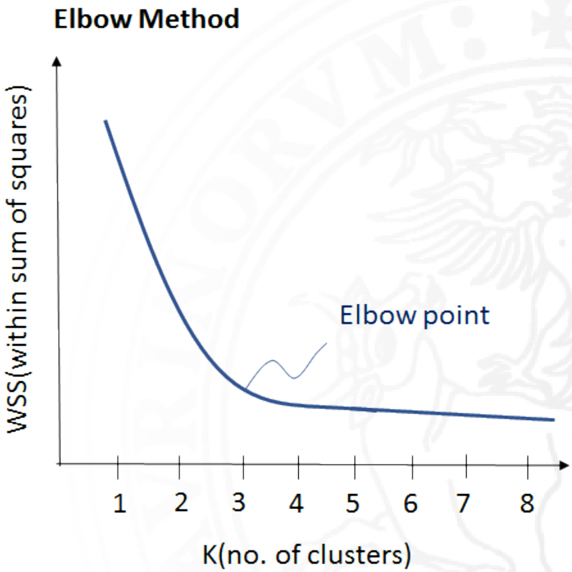
- There is no method for determining the exact value of K;
- One of the metrics that is commonly used to compare results across different values of K is **elbow method**
- It graphs the average internal per cluster sum of squares distance vs the number of clusters to find a visual “elbow” which is the optimal number of clusters.

$$W_k = \sum_{r=1}^{k} \frac{1}{n_r} D_r \tag{1}$$

Where k is the number of clusters, n_r is the number of points in cluster r and D_r is the sum of distances between all points in a cluster:

$$D_r = \sum_{i=1}^{n_r-1} \sum_{j=i}^{n_r} (d_i - d_j)^2 \tag{2}$$

Elbow plot


Elbow Method
WSS(within sum of squares)
Elbow point
1
2
3
4
5
6
7
8
K(no. of clusters)