

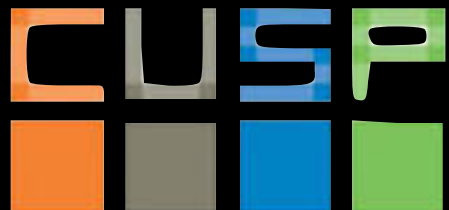
Urban Informatics

Fall 2015

dr. federica bianco fbianco@nyu.edu



@fedhere

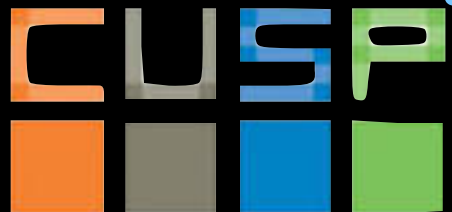


Recap:

- Good practices with data: falsifiability, reproducibility
- Basic data retrieving and munging: APIs, Data formats
- SQL
- Basic statistics: distributions and their moments
- Hypothesis testing: p -value, statistical significance
- Statistical and Systematic errors
- Goodness of fit tests
- Likelihood
- OLS
- Topics in (time) series analysis
- Visualizations
- Geospatial analysis
- Topics in time series analysis

Today:

- Clusters



machine learning

algorithms that can learn from and make predictions on data.

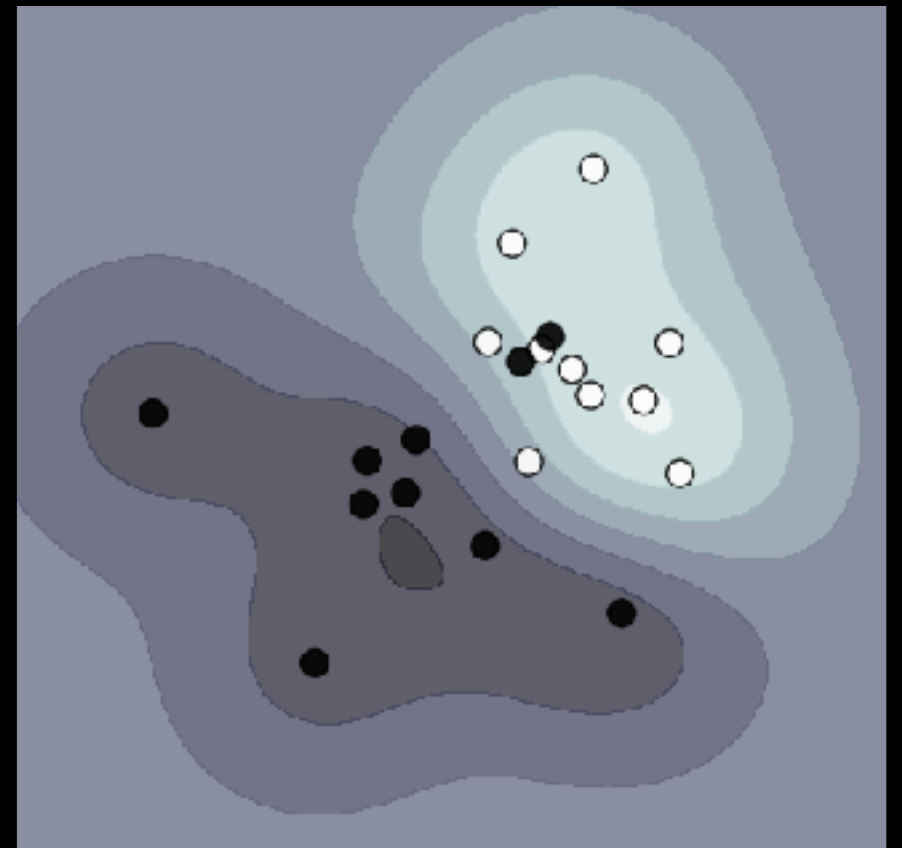
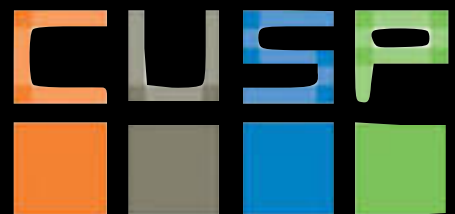


machine learning

algorithms that can learn from and make predictions on data.



supervised learning
extract features and create
models that allow
prediction where the
correct answer is known for
a subset of the data



XI: Clustering

machine learning

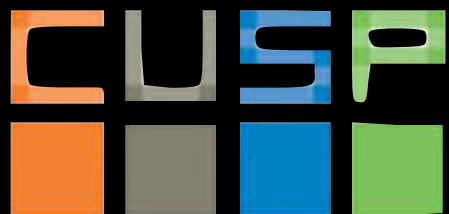
algorithms that can learn from and make predictions on data.



supervised learning
extract features and create
models that allow
prediction where the
correct answer is known for
a subset of the data



unsupervised learning
identify features and create
models that allow to
understand structure in the
data



machine learning

algorithms that can learn from and make predictions on data.



supervised methods

classification

prediction

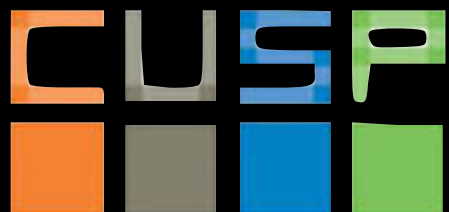


unsupervised methods

understanding structure

organizing + compressing data

(classification, feature learning)



machine learning

algorithms that can learn from and make predictions on data.



supervised methods

classification

prediction

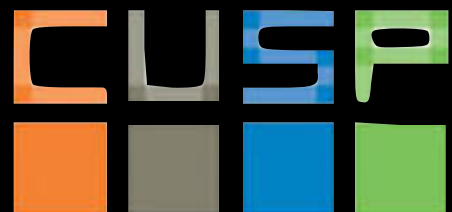


unsupervised methods

understanding structure

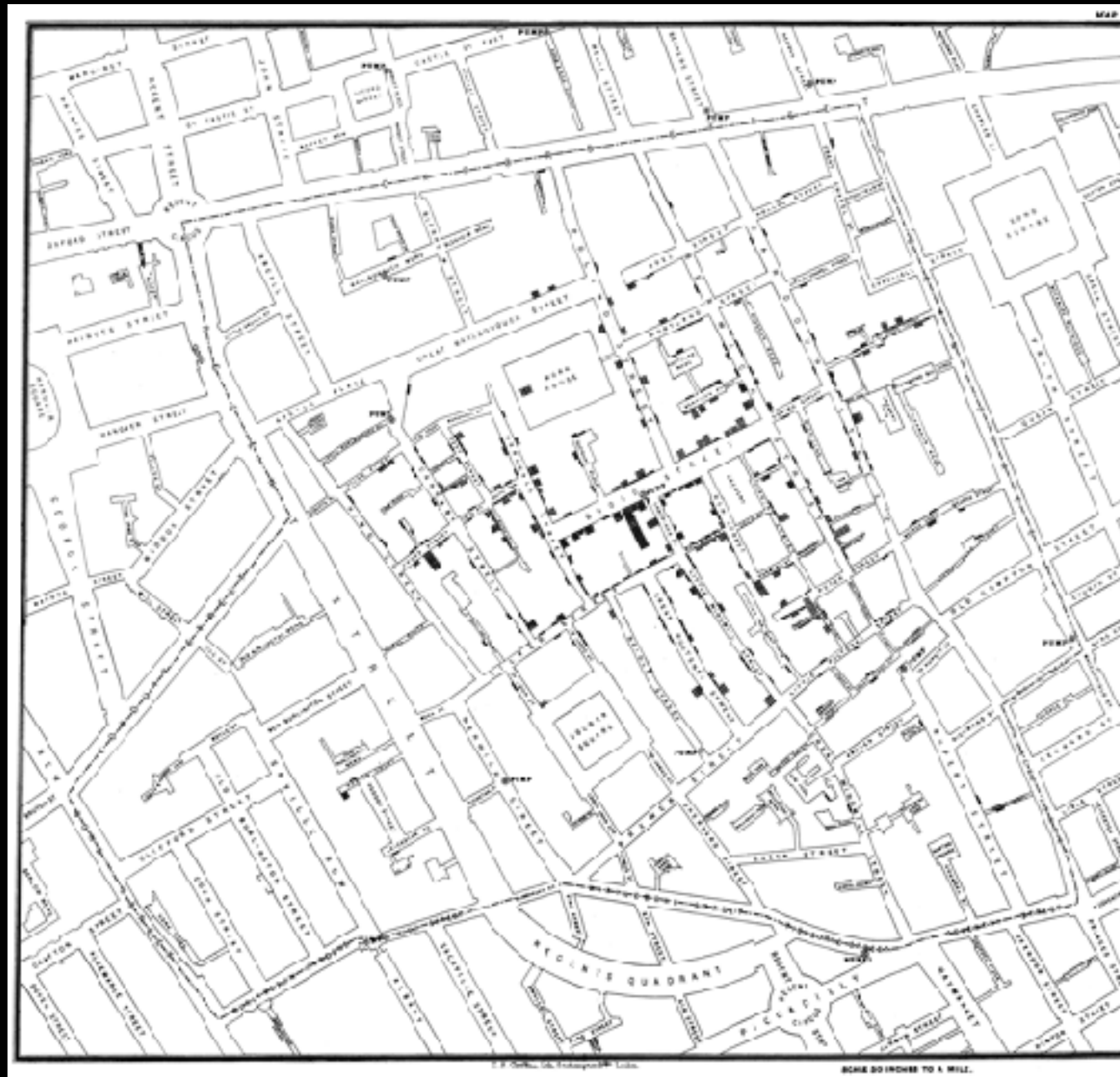
organizing + compressing data

(classification, feature learning)



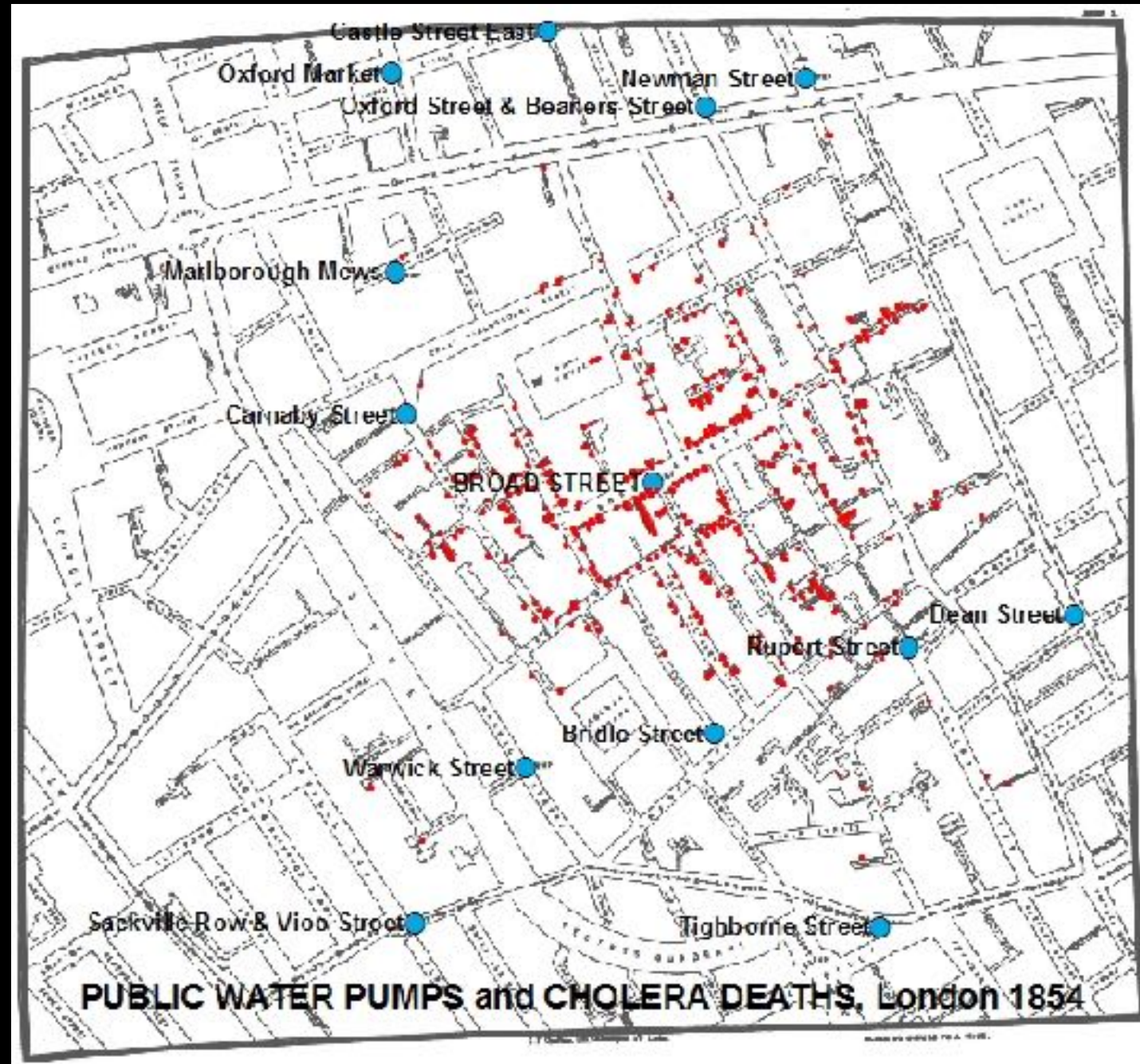
What is clustering?

Dr. John Snow September 1854 cholera outbreak map

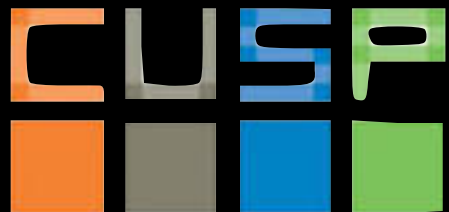


Steven Johnson's 2006 book *The Ghost Map: the Story of London's Most Terrifying Epidemic, and How it Changed Science*.

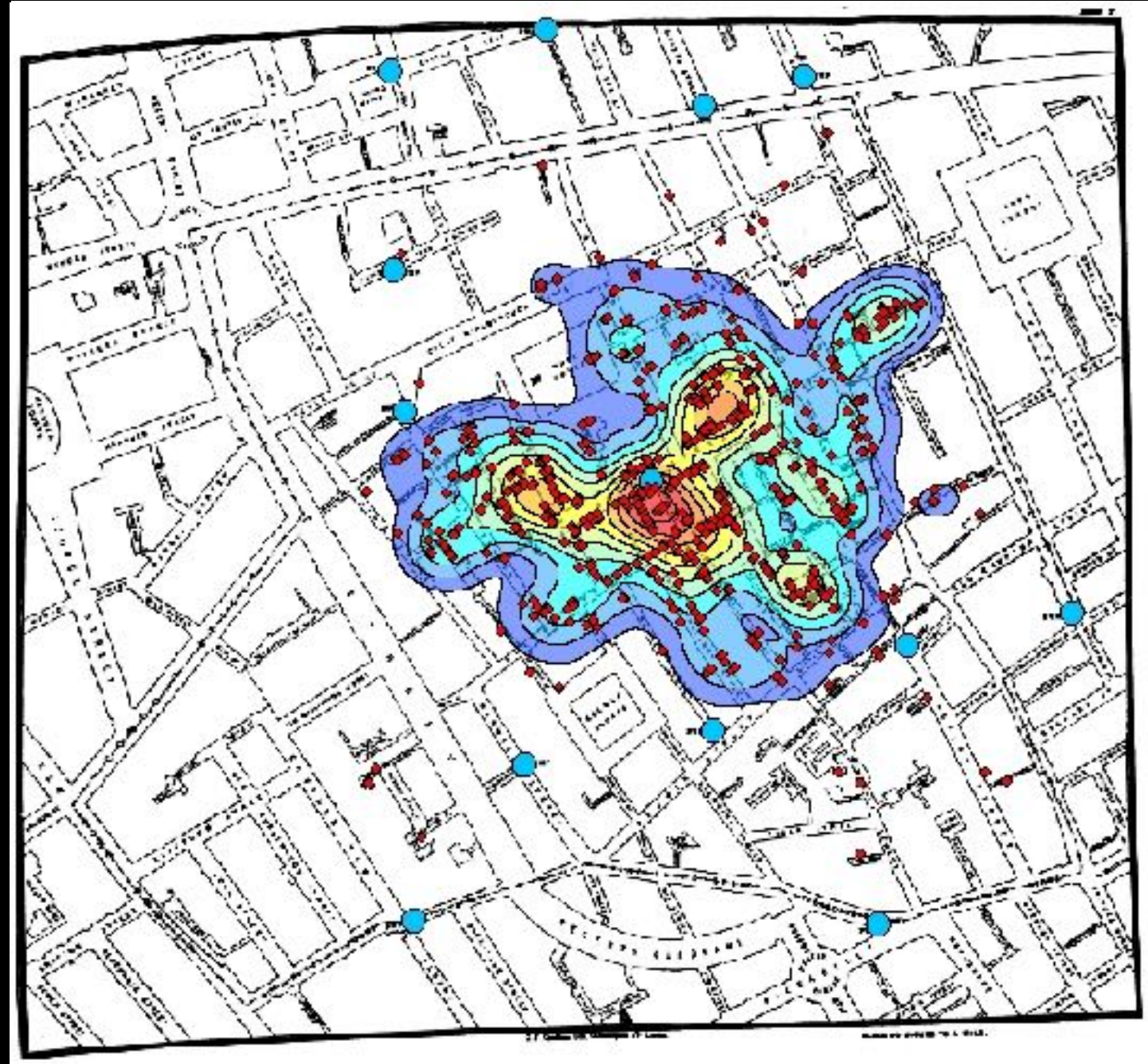
Dr. John Snow September 1854 cholera outbreak map



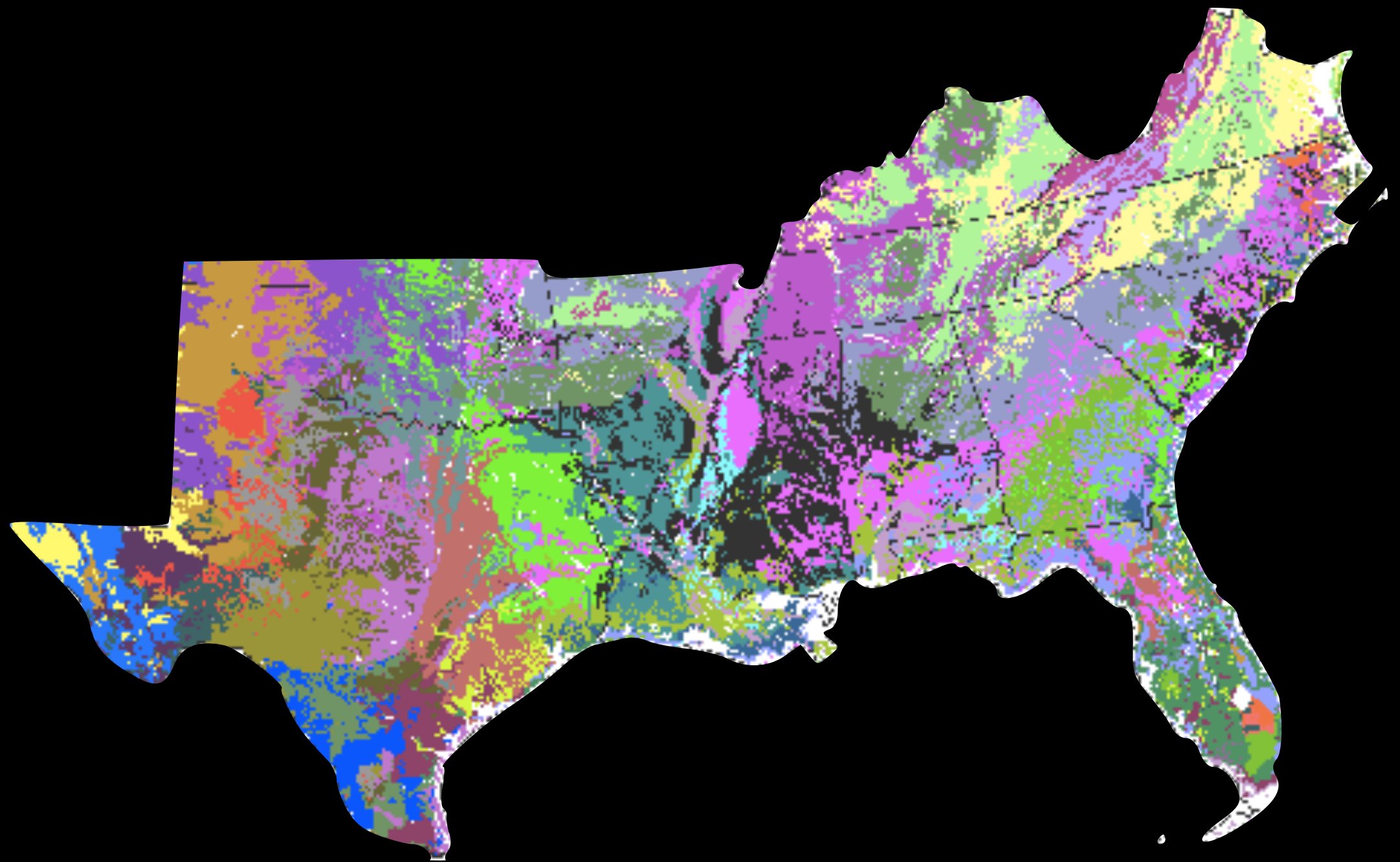
digitization by John Mackenzie, University of Delaware
<https://www1.udel.edu/johnmack/frec682/cholera/cholera2.html>



Dr. John Snow September 1854 cholera outbreak map

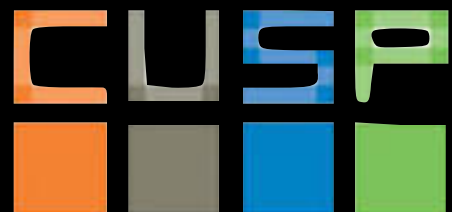


digitization by John Mackenzie, University of Delaware
<https://www1.udel.edu/johnmack/frec682/cholera/cholera2.html>

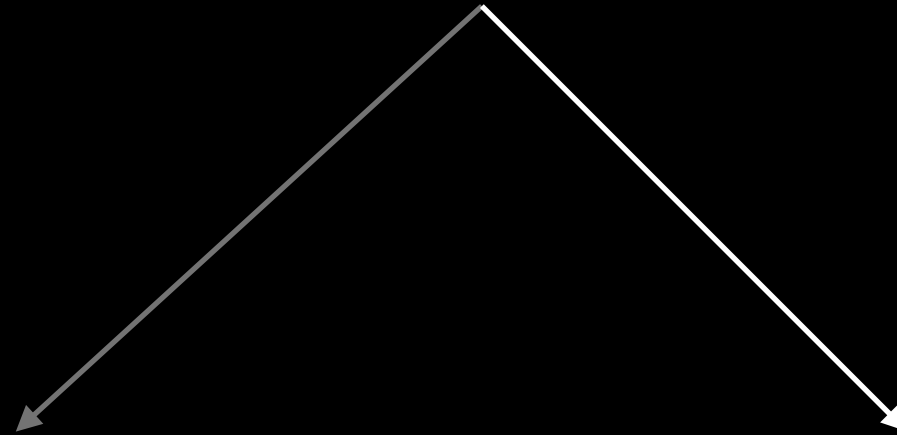


A Spatial Clustering Technique for the Identification of Customizable Ecoregions

William W. Hargrove and Robert J. Luxmoore
50-year mean monthly **temperature**, 50-year mean
monthly **precipitation**, **elevation**, total plant-available
water content of soil, total **organic matter in soil**, and
total Kjeldahl **soil nitrogen**



machine learning



supervised methods

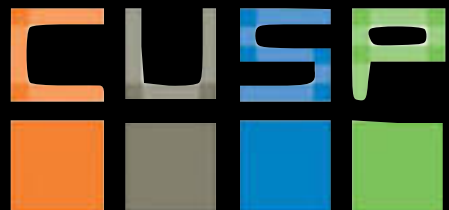
classification
prediction

unsupervised methods

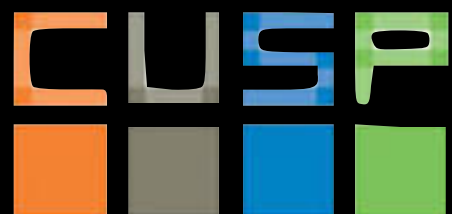
understanding structure
organizing + compressing data

GOAL:

**partitioning data in *maximally homogeneous,*
maximally distinguished subsets.**



machine learning



<http://www-bcf.usc.edu/~soltanol/Applications.html>

XI: Clustering

what is a cluster?

- **internal criterion:** members of the cluster should be similar to each other (**intra cluster compactness**)

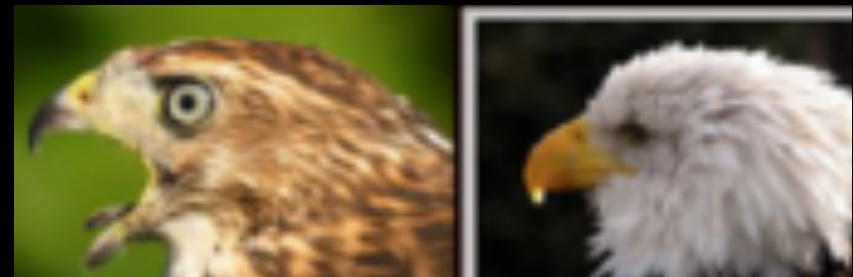
tigers



whales

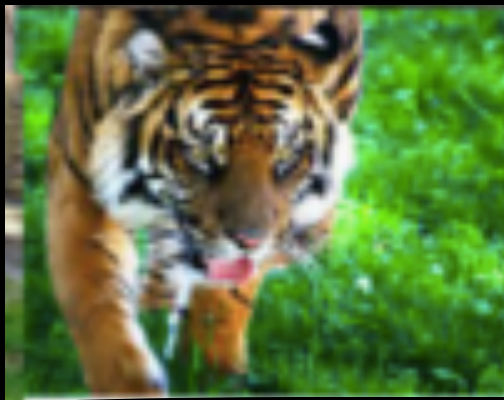
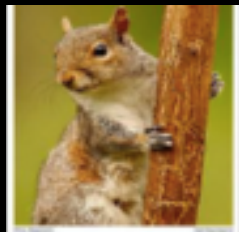


eagles



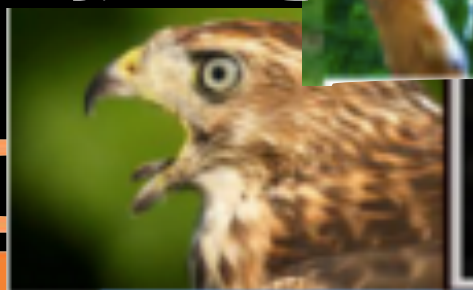
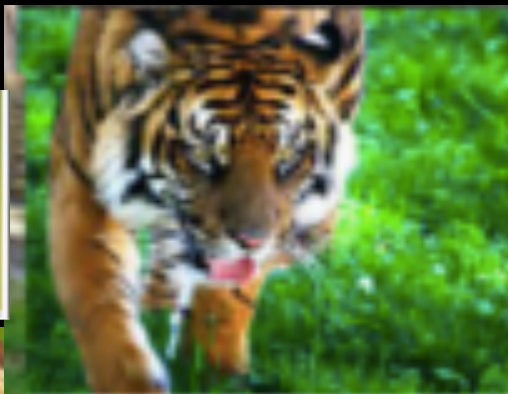
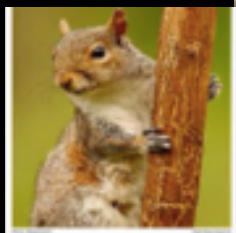
what is a cluster?

- **internal criterion:** members of the cluster should be similar to each other
- **external criterion:** objects outside the cluster should be dissimilar from the objects inside the cluster



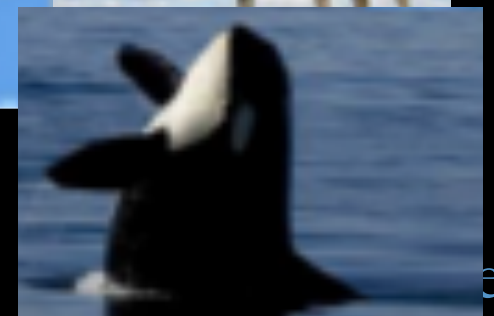
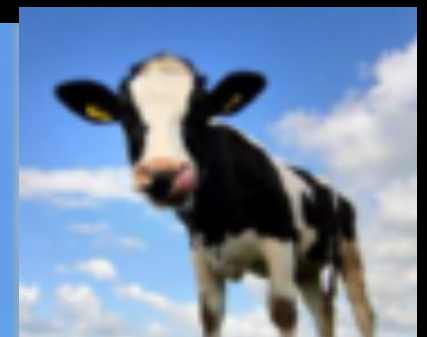
what is a cluster?

- **internal criterion:** members of the cluster should be similar to each other
- **external criterion:** objects outside the cluster should be dissimilar from the objects inside the cluster



green
brown & white

blue
black
&
white



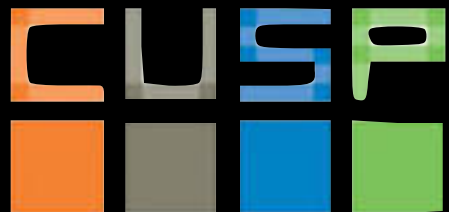
ering



[https://github.com/fedhere/Ulnotebooks/blob/master/cluster/
imageProcessingKmeans.ipynb](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/imageProcessingKmeans.ipynb)

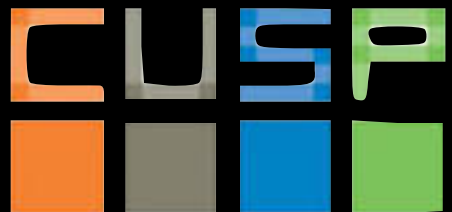
[https://github.com/fedhere/Ulnotebooks/blob/master/cluster/
clusteringCUSPfaces.ipynb](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/clusteringCUSPfaces.ipynb)

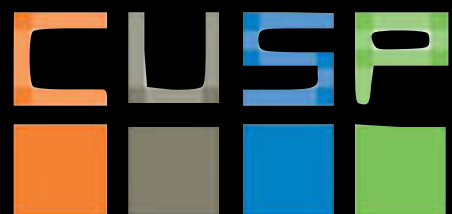
[https://github.com/fedhere/Ulnotebooks/blob/
master/cluster/hardVSsoftClustering.ipynb](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/hardVSsoftClustering.ipynb)



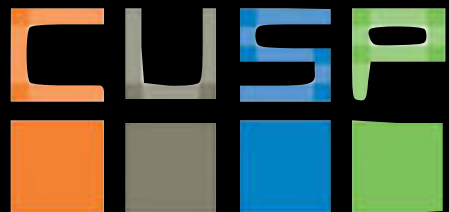
The ideal clustering algorithm:

- **Scalability (naive algorithms are N^2 hard)**
- **Ability to deal with different types of attributes**
- **Discovery of clusters with arbitrary shapes**
- **Minimal requirement for domain knowledge**
- **Deals with noise and outliers**
- **Insensitive to order**
- **Allows incorporation of constraints**
- **Interpretable**





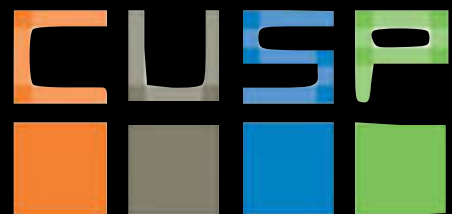
Defining the distance



Distance Metrics Continuous variables

Minkowski family of distances

$$D(i,j) = \sqrt[p]{|x_{i1}-x_{j1}|^p + |x_{i2}-x_{j2}|^p + \dots + |x_{iN}-x_{jN}|^p}$$

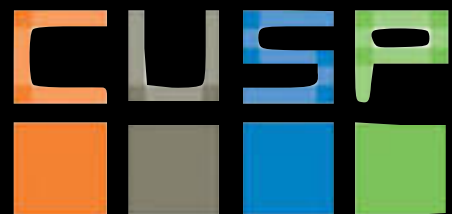


Distance Metrics Continuous variables

Minkowski family of distances

$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p}$$

N features (dimensions)



Distance Metrics Continuous variables

Minkowski family of distances

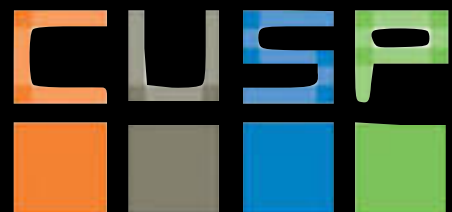
$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p} \quad \text{N features (dimensions)}$$

$$D(i,j) > 0$$

$$D(i,i) = 0$$

$$D(i,j) = D(j,i)$$

$$D(i,j) \leq D(i,k) + D(k,j)$$



Distance Metrics Continuous variables

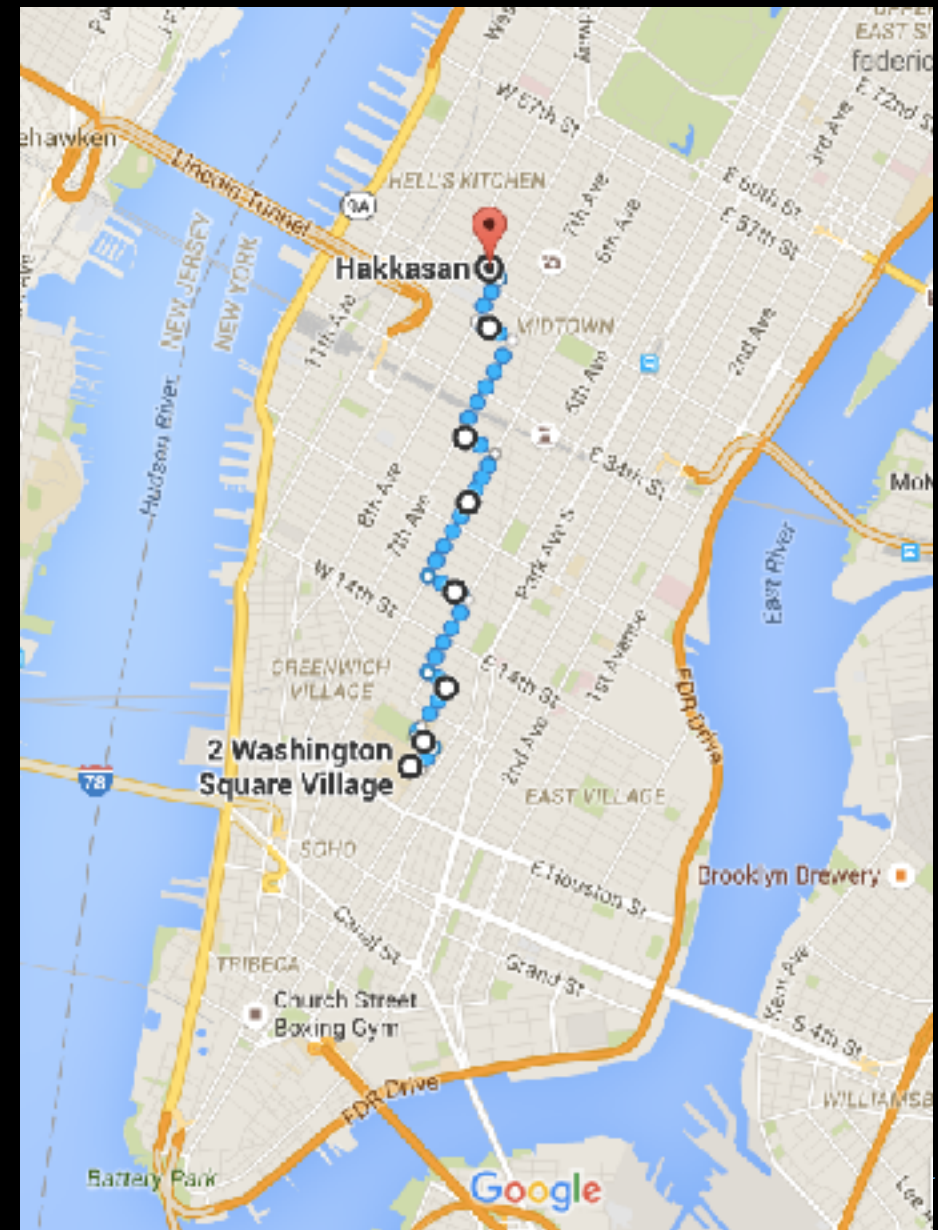
Minkowski family of distances

$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p}$$

N features (dimensions)

Manhattan: $p = 1$

$$D_{Man}(i,j) = \sum_{k=1}^N |x_{ik} - x_{jk}|$$



ustering

Distance Metrics Continuous variables

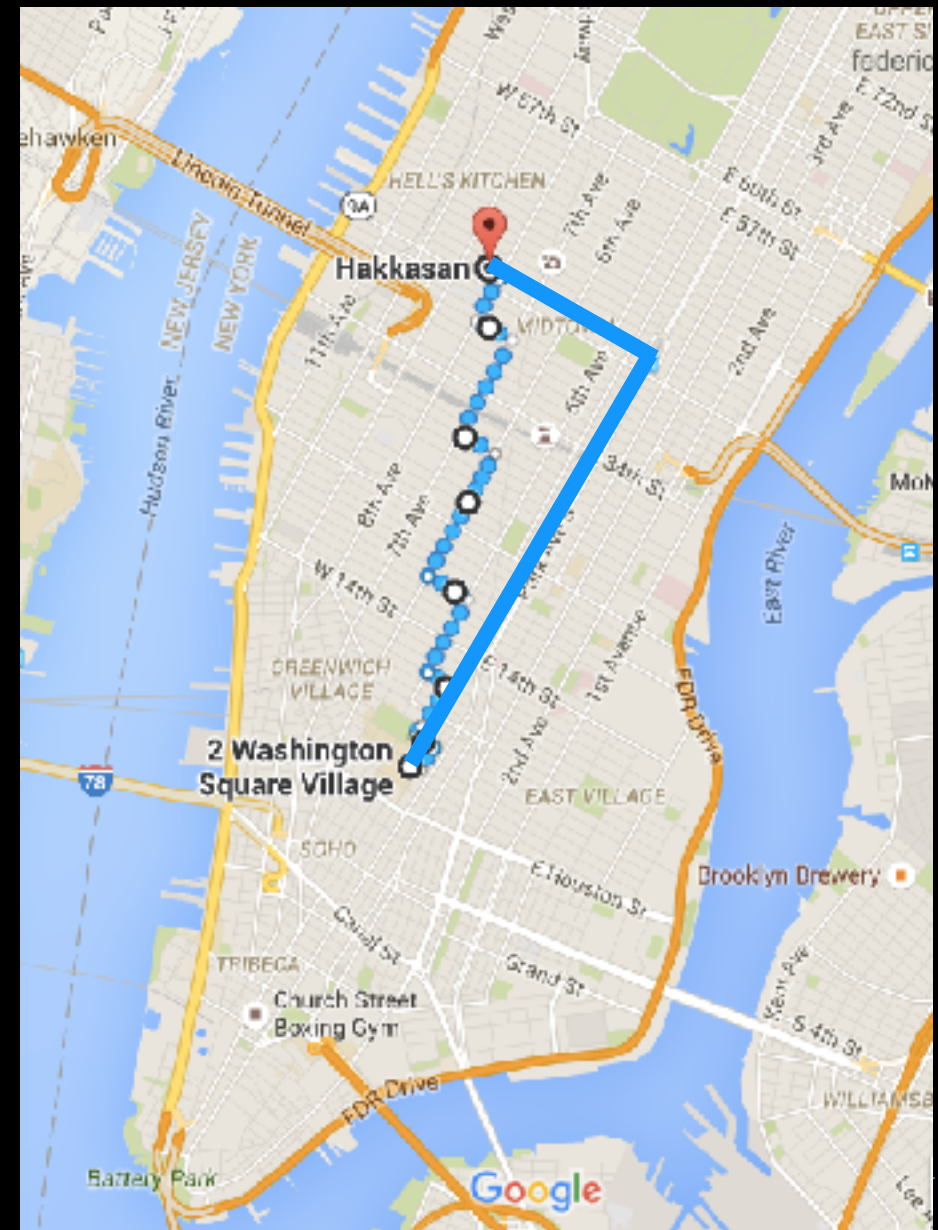
Minkowski family of distances

$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p}$$

N features (dimensions)

Manhattan: $p = 1$

$$D_{Man}(i,j) = \sum_{k=1}^N |x_{ik} - x_{jk}|$$



ustering

Distance Metrics Continuous variables

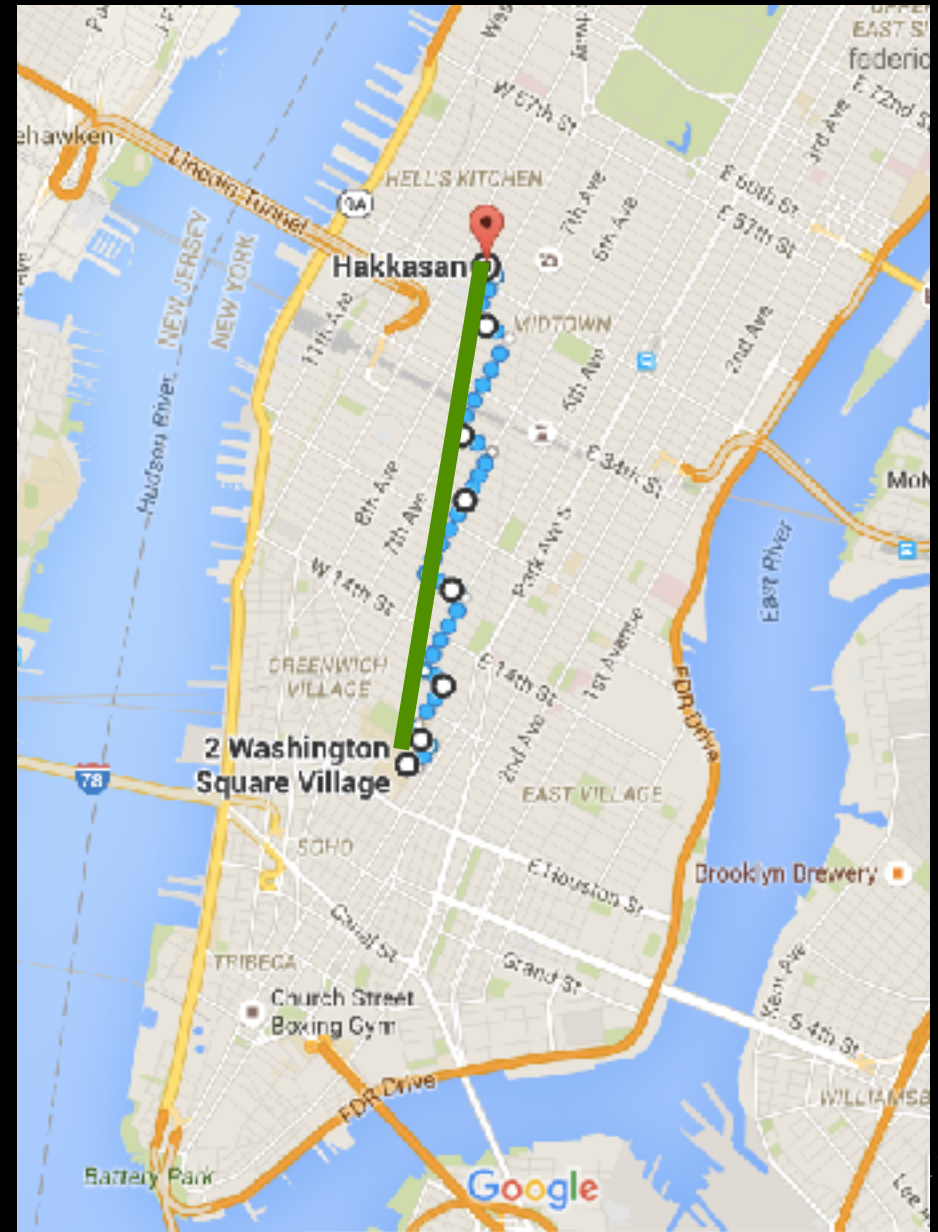
Minkowski family of distances

$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p}$$

N features (dimensions)

Euclidean: $p = 2$

$$D_{Euc}(i,j) = \sqrt{\sum_{k=1}^N |x_{ik} - x_{jk}|^2}$$



stering

Distance Metrics Continuous variables

Minkowski family of distances

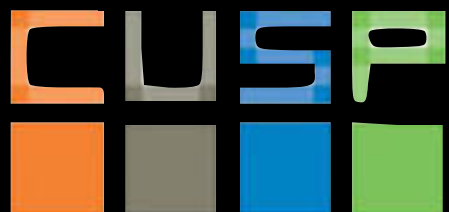
$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p}$$



Great Circle distances: $\phi_i, \lambda_i, \phi_j, \lambda_j$

geographical latitude and longitude

$$D(i,j) = R \arccos(\sin \phi_i \cdot \sin \phi_j + \cos \phi_i \cdot \cos \phi_j \cdot \cos(\Delta\lambda))$$



Distance Metrics Continuous variables

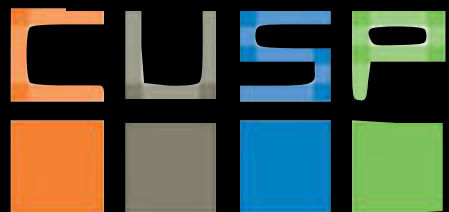
Minkowski family of distances

$$D(i,j) = \sqrt[p]{\sum_{k=1}^N |x_{ik} - x_{jk}|^p}$$

N features (dimensions)

Weighted distances:

$$D(i,j) = \sqrt[p]{w_1 |x_{i1} - x_{j1}|^p + w_2 |x_{i2} - x_{j2}|^p + \dots + w_N |x_{iN} - x_{jN}|^p}$$

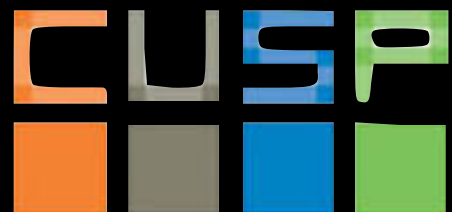


Distance Metrics Binary variables

Uses presence/absence data in two samples

**Simple similarity
coefficient *SMC***

$$S_{ij} = \frac{M_{i=0j=0} + M_{i=1j=1}}{M_{00} + M_{01} + M_{10} + M_{11}}$$



Distance Metrics Binary variables

	1	0	<i>sum</i>
1	<i>a</i>	<i>b</i>	<i>a+b</i>
0	<i>c</i>	<i>d</i>	<i>c+d</i>
<i>sum</i>	<i>a+c</i>	<i>b+d</i>	<i>p</i>

Uses presence/absence data in two samples

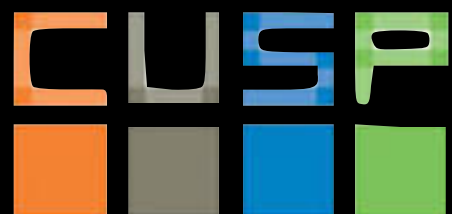
Simple similarity coefficient *SMC*

$$S_{ij} = \frac{b+c}{a+b+c+d}$$

a = number of items in common,

b = number of items unique to the first set

c = number of items unique to the second set

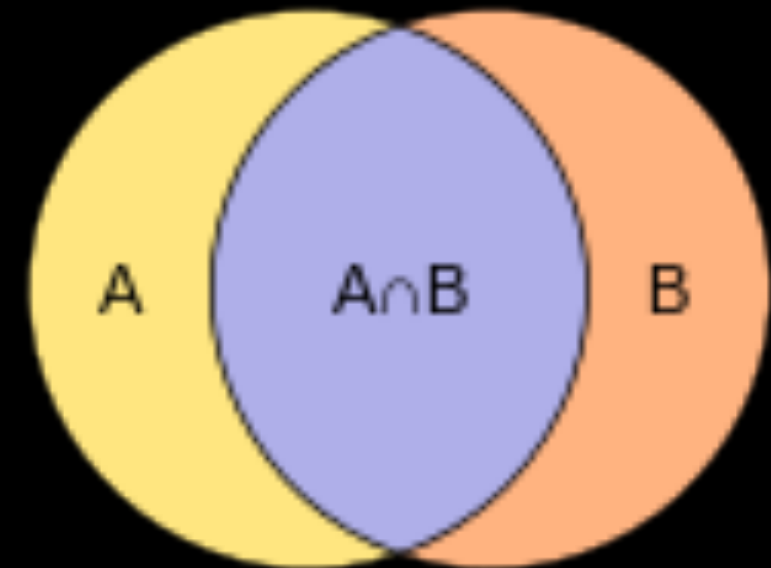


Distance Metrics Sets variables

Uses presence/absence data

Jaccard similarity coefficient S_j

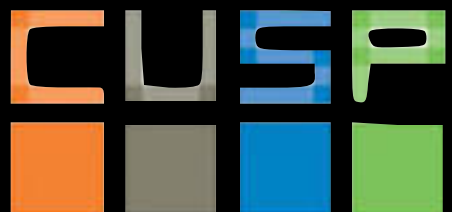
$$S_j = \frac{a}{a+b+c}$$



a = number of items in common,

b = number of items unique to the first set

c = number of items unique to the second set

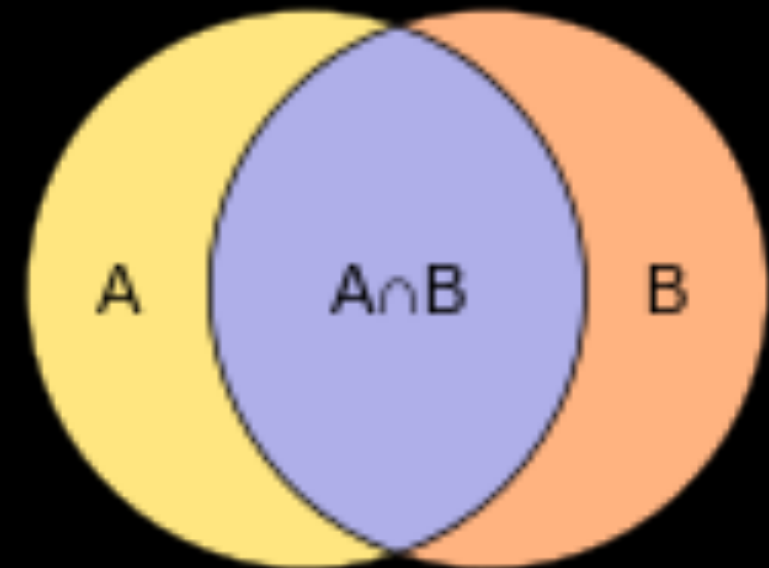


Distance Metrics Sets variables

Uses presence/absence data

**Jaccard similarity
coefficient S_j**

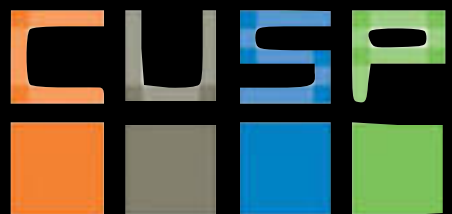
$$S_j = \frac{A \cap B}{A \cup B}$$



a = number of items in common,

b = number of items unique to the first set

c = number of items unique to the second set

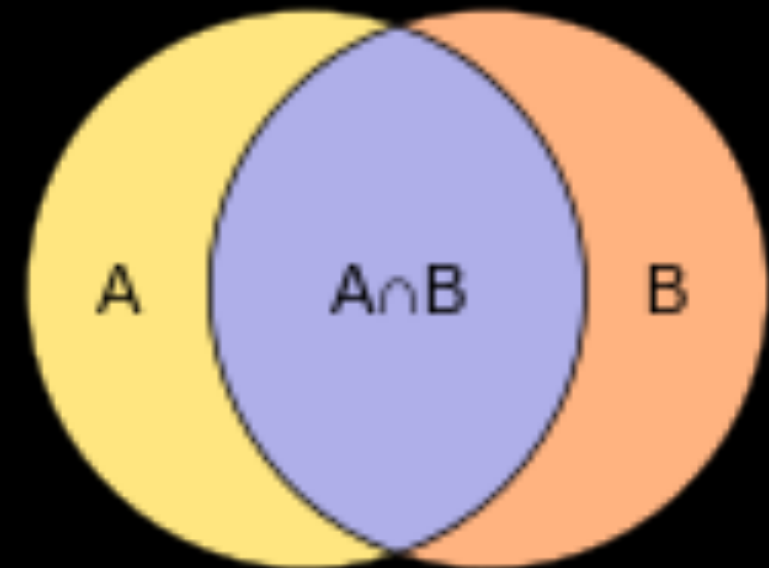


Distance Metrics Sets variables

Uses presence/absence data

Jaccard distance $D_j = 1 - S_j$

$$S_j = \frac{A \cap B}{A \cup B}$$



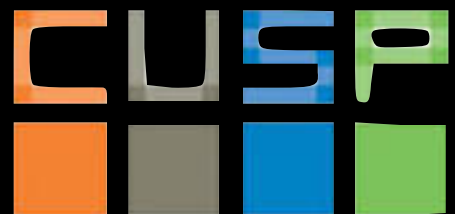
a = number of items in common,

b = number of items unique to the first set

c = number of items unique to the second set

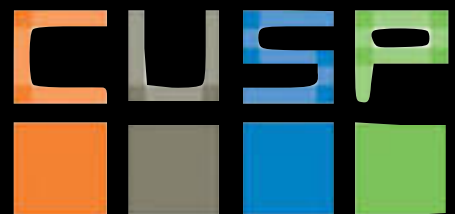


[https://github.com/fedhere/Ulnotebooks/blob/master/cluster/
categoricalDistance.ipynb](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/categoricalDistance.ipynb)

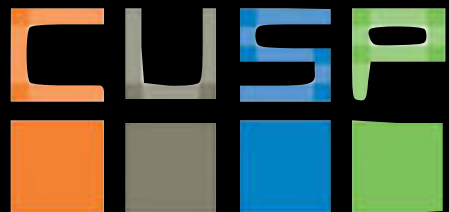




[https://github.com/fedhere/Ulnotebooks/blob/master/cluster/
thanksgivingClustering.ipynb](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/thanksgivingClustering.ipynb)



How clustering works



Clustering methods

- **Partitioning**

- Hard clustering**

- K-means (McQueen '67)

- K-medoids (Kaufman & Rausseeuw '87)

- Soft Clustering**

- Expectation Maximization (Dempster, Laird, Rubin '77)

- **Hierarchical**

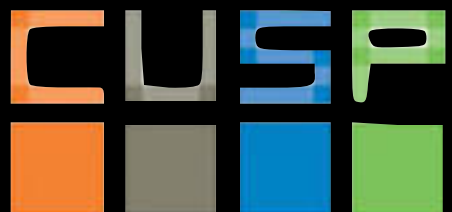
- agglomerative

- divisive

- **also:**
 - **Density based**

- **Grid based**

- **Model based**



Clustering methods

- **Partitioning**

- Hard clustering**

- K-means (McQueen '67)

- K-medoids (Kaufman & Rausseeuw '87)

- Soft Clustering**

- Expectation Maximization (Dempster, Laird, Rubin '77)

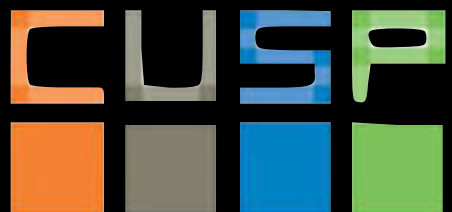
- **Hierarchical**
agglomerative

- divisive

- **also:**
 - **Density based**

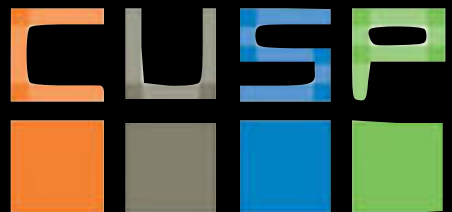
- **Grid based**

- **Model based**



K-means:

1. **Choose N “centers” guesses:** random points in the data space
2. **Calculate which center each datapoint is closest to:** these are the N clusters
3. **Calculate the new centers as means of the assigned clusters:** these are the new N centers
4. **Iterate 2&3 till convergence:** when clusters no longer change



K-means:

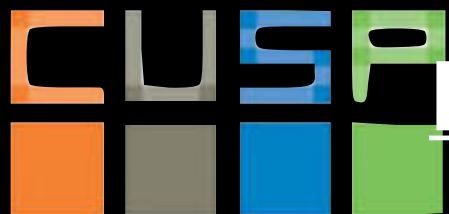
Minimizes the intra cluster gaussian

Order: #clusters #dimensions #iterations #datapoints
 $O(KdN)$

works on minimizing the aggregate distance within the cluster if the distance is Euclidean this is the same as minimizing the variance

Its non-deterministic: the result depends on the (random) starting point

It only works where the mean is defined: alternative is K-medoids which represents the cluster by its central member, rather than by the mean



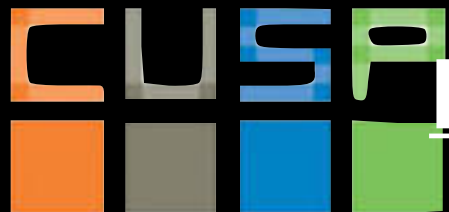
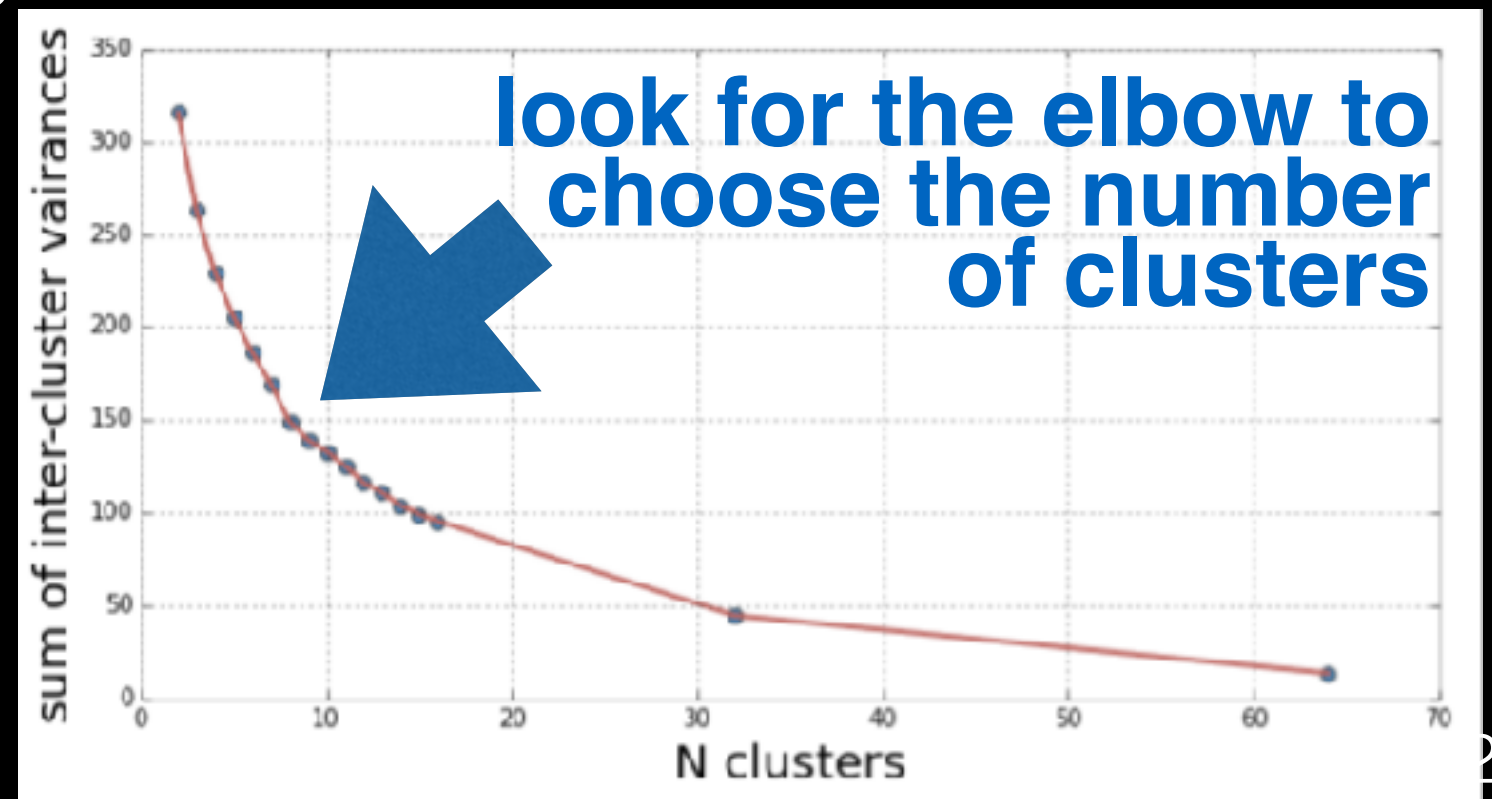
Must declare the number of clusters upfront

K-means:

Minimizes the intra cluster gaussian

Order: #clusters #dimensions #iterations #datapoints
 $O(KdN)$

works on minimizing the aggregate distance within the cluster if the distance is Euclidean this is the same as minimizing the variance

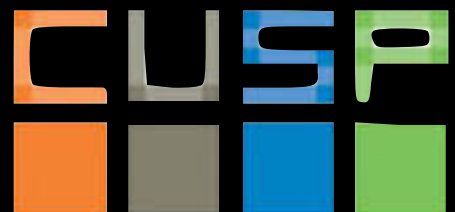


Must declare the number of clusters upfront



[https://github.com/fedhere/Ulnotebooks/blob/master/cluster/](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/kmeans.ipynb)

kmeans.ipynb



XI: Clustering

Clustering methods

- **Partitioning**

- Hard clustering**

- K-means (McQueen '67)

- K-medoids (Kaufman & Rausseeuw '87)

- Soft Clustering**

- Expectation Maximization (Dempster, Laird, Rubin '77)

- **Hierarchical**

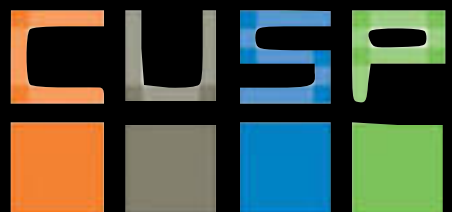
- agglomerative

- divisive

- **also:** **Density based**

- Grid based**

- Model based**

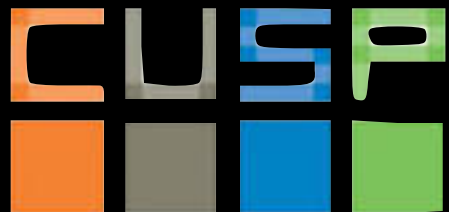


Hard Clustering:

each object in the sample belongs to only 1 cluster

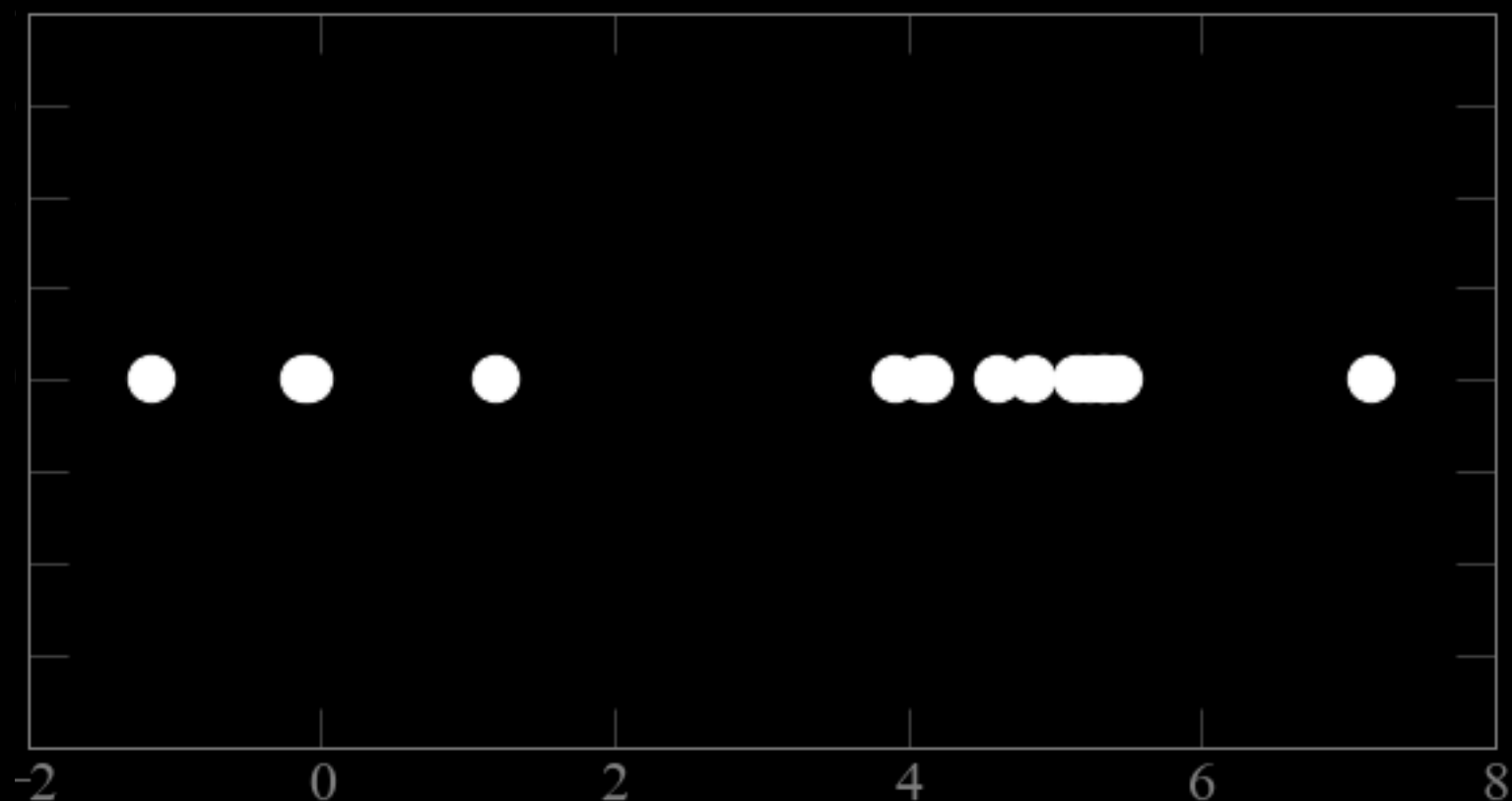
Soft Clustering:

to each object in the sample we assign a degree of belief that it belongs to a cluster

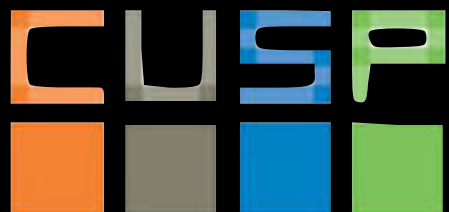


Mixture models

A probabilistic way to do soft clustering

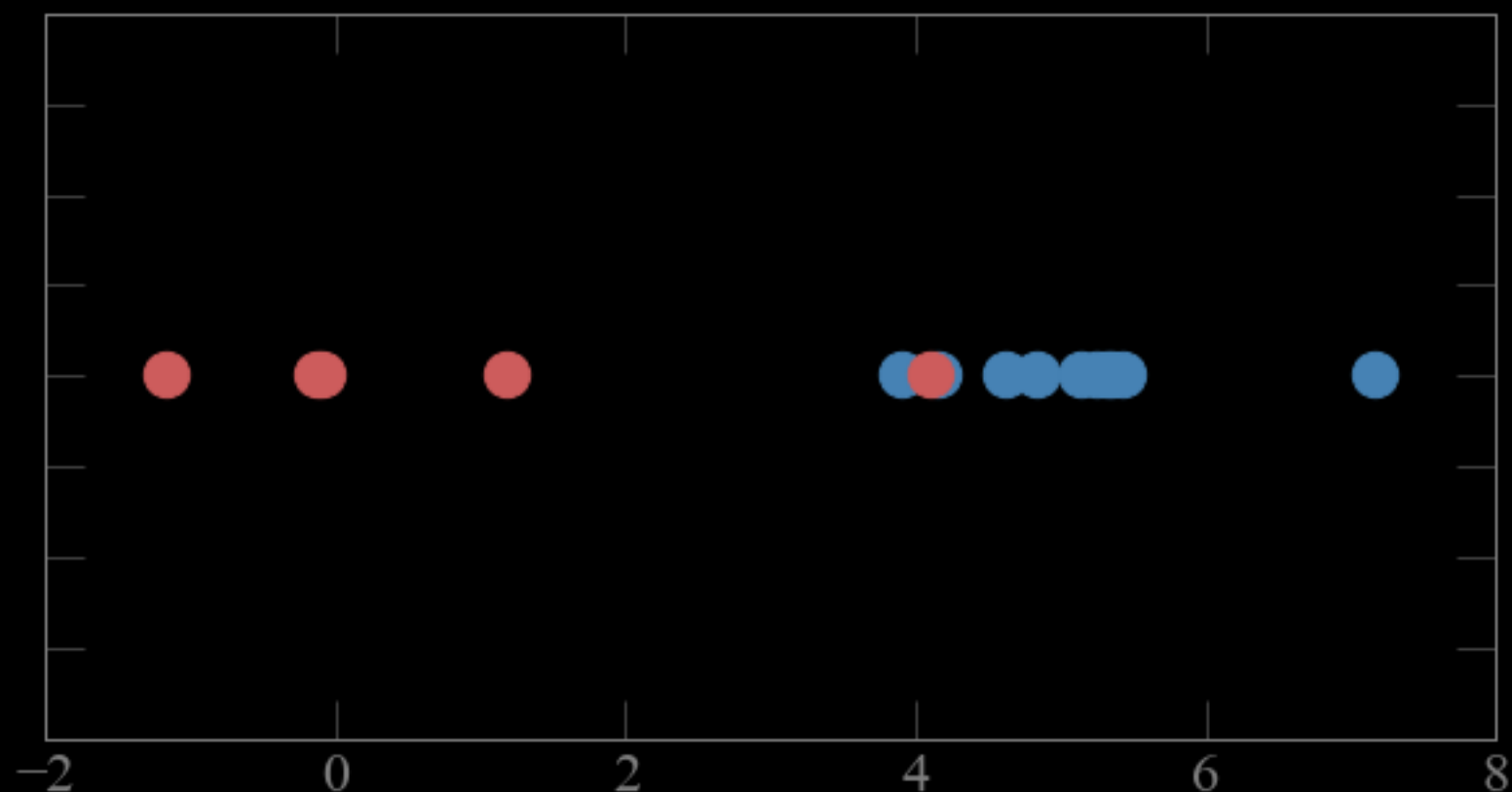


these points come from 2 gaussian distribution.
which point comes from which gaussian?

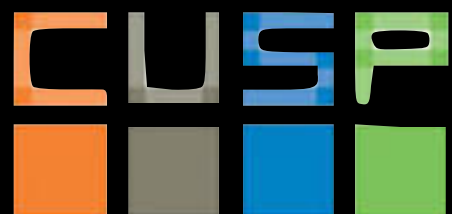


Mixture models

A probabilistic way to do soft clustering

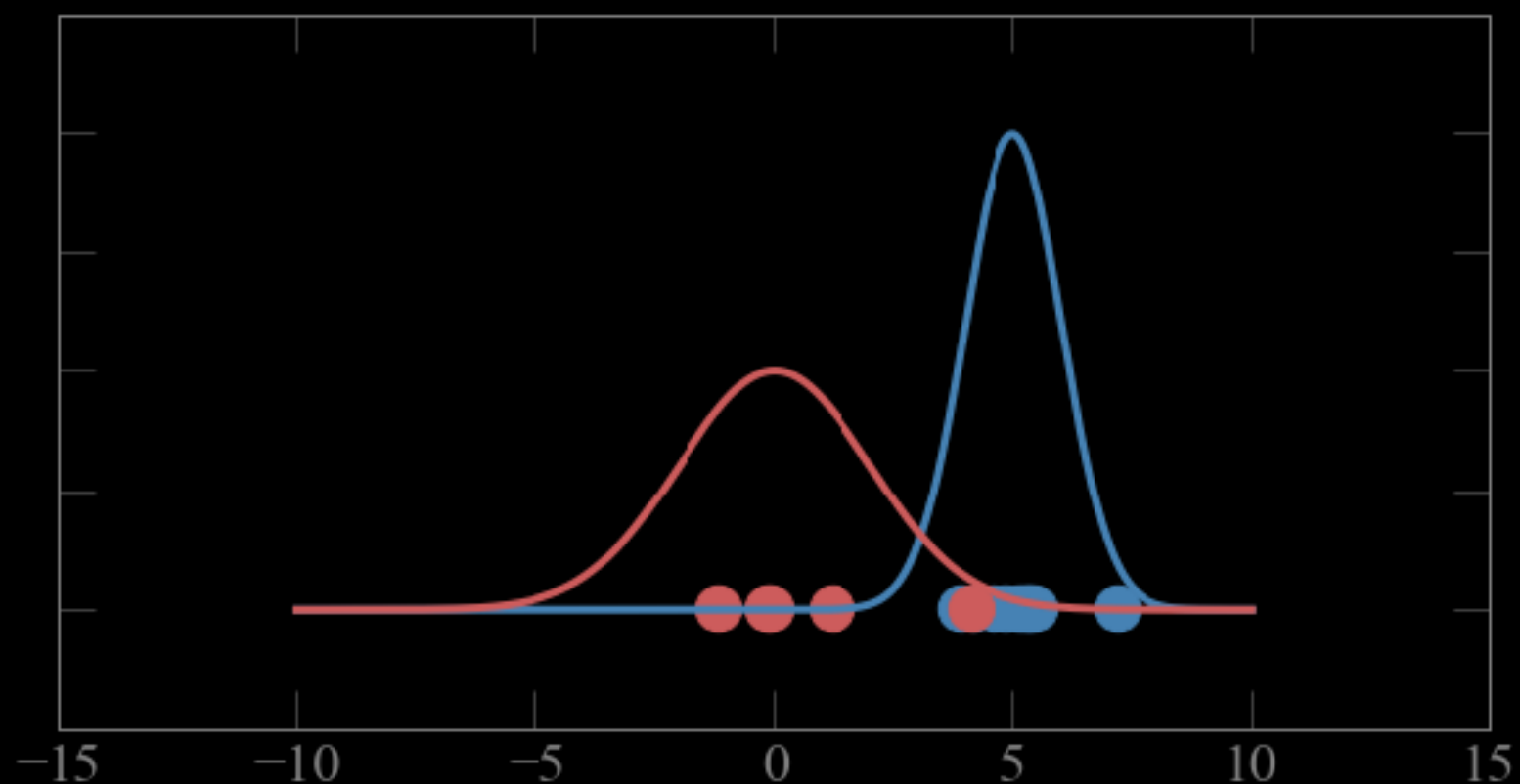


if i know which point comes from which gaussian
i can solve for the parameters of the gaussian
(e.g. maximizing likelihood)



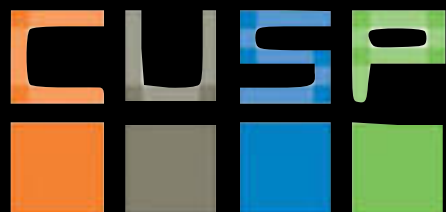
Mixture models

A probabilistic way to do soft clustering



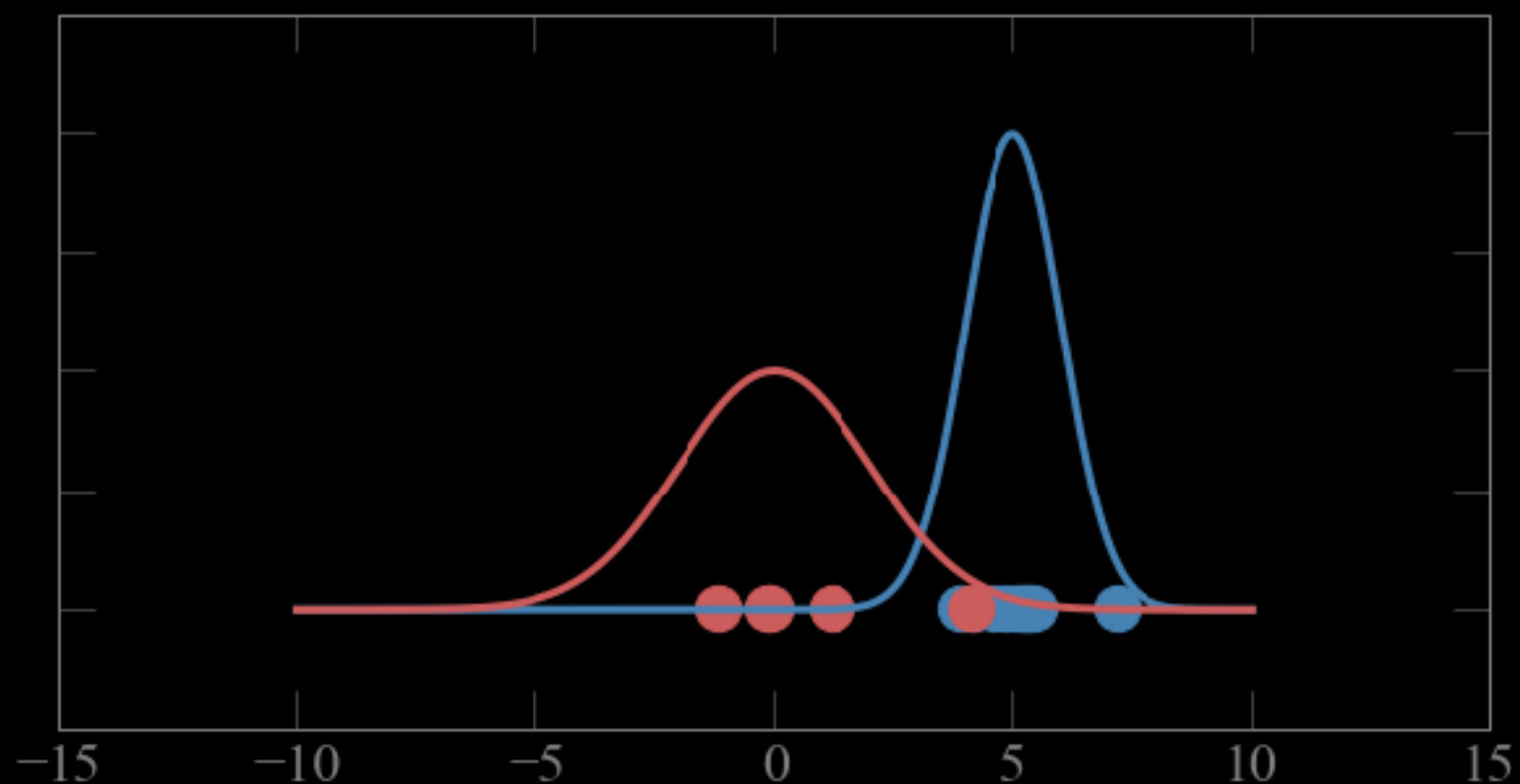
if i know which the parameters (μ, σ) of the gaussians
i can figure out which gaussian each point is most
likely to come from (calculate probability)

XI: Clustering

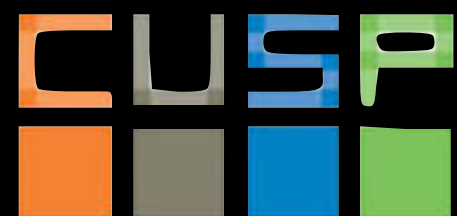


Mixture models

A probabilistic way to do soft clustering



if i know which the parameters (μ, σ) of the gaussians



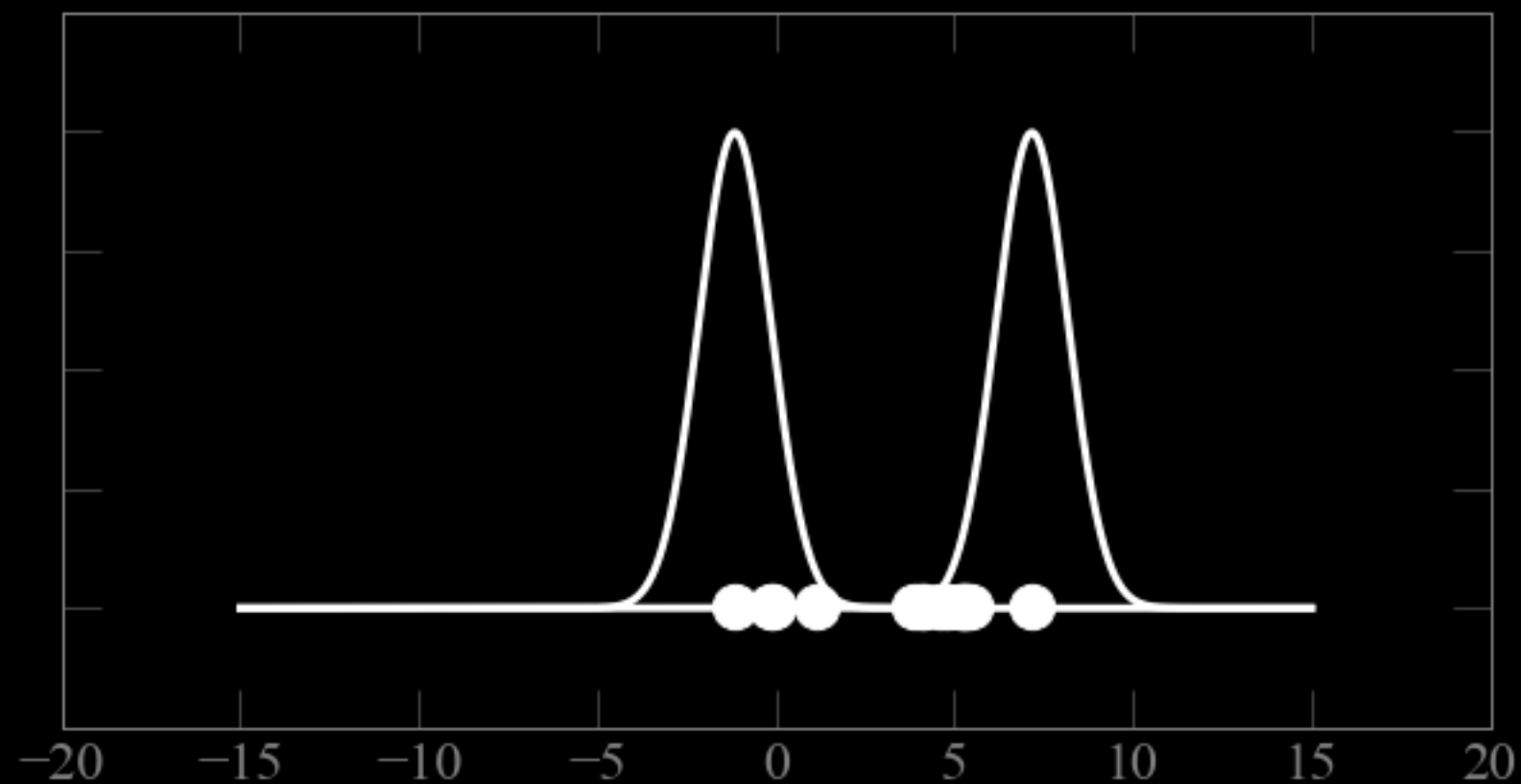
i can figure out which gaussian each point is most

likely to come from (calculate probability)

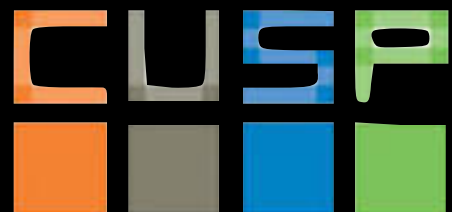
XI: Clustering

EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$

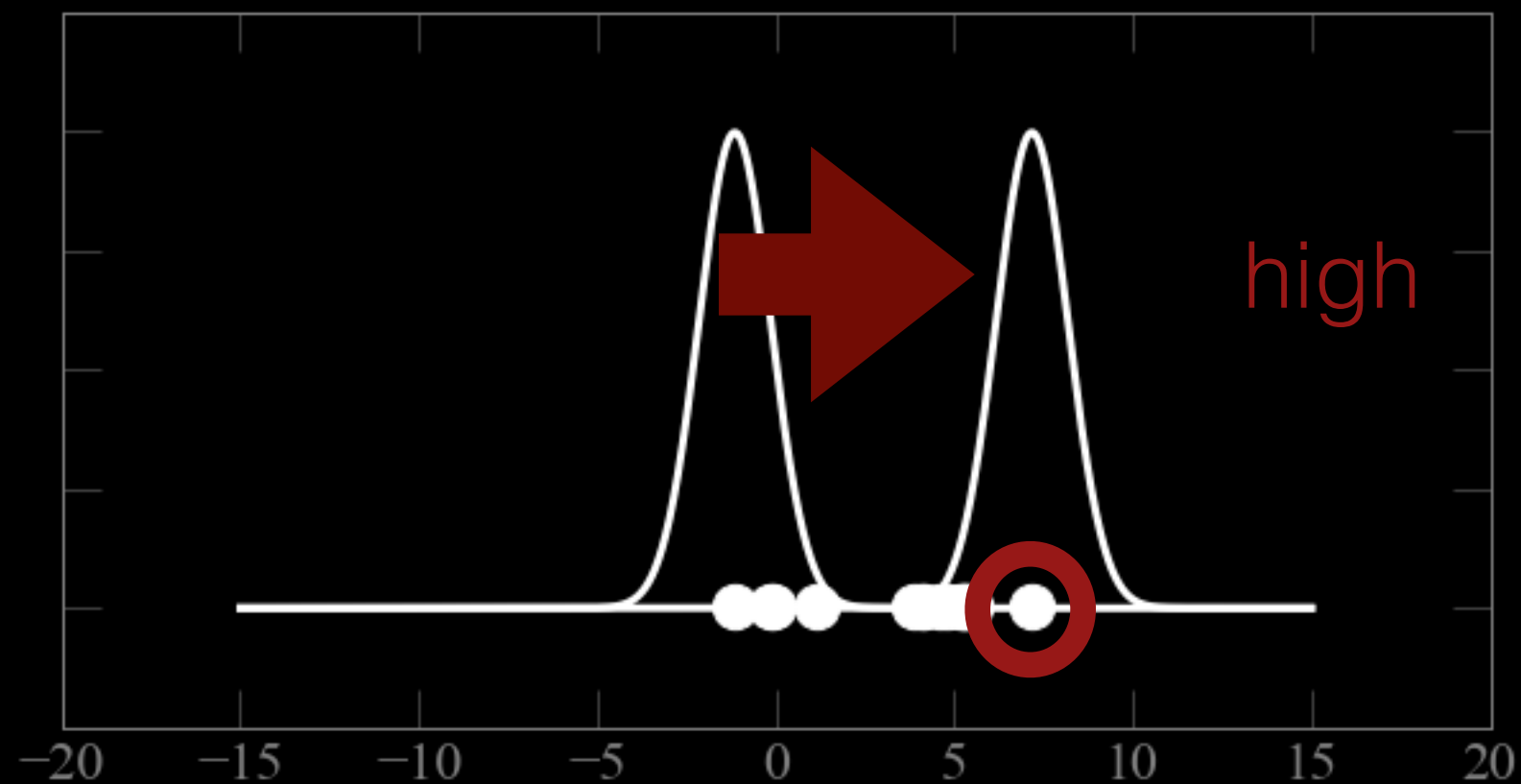


for every point calculate the probability it comes from either gaussian

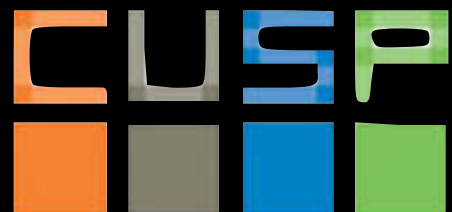


EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$

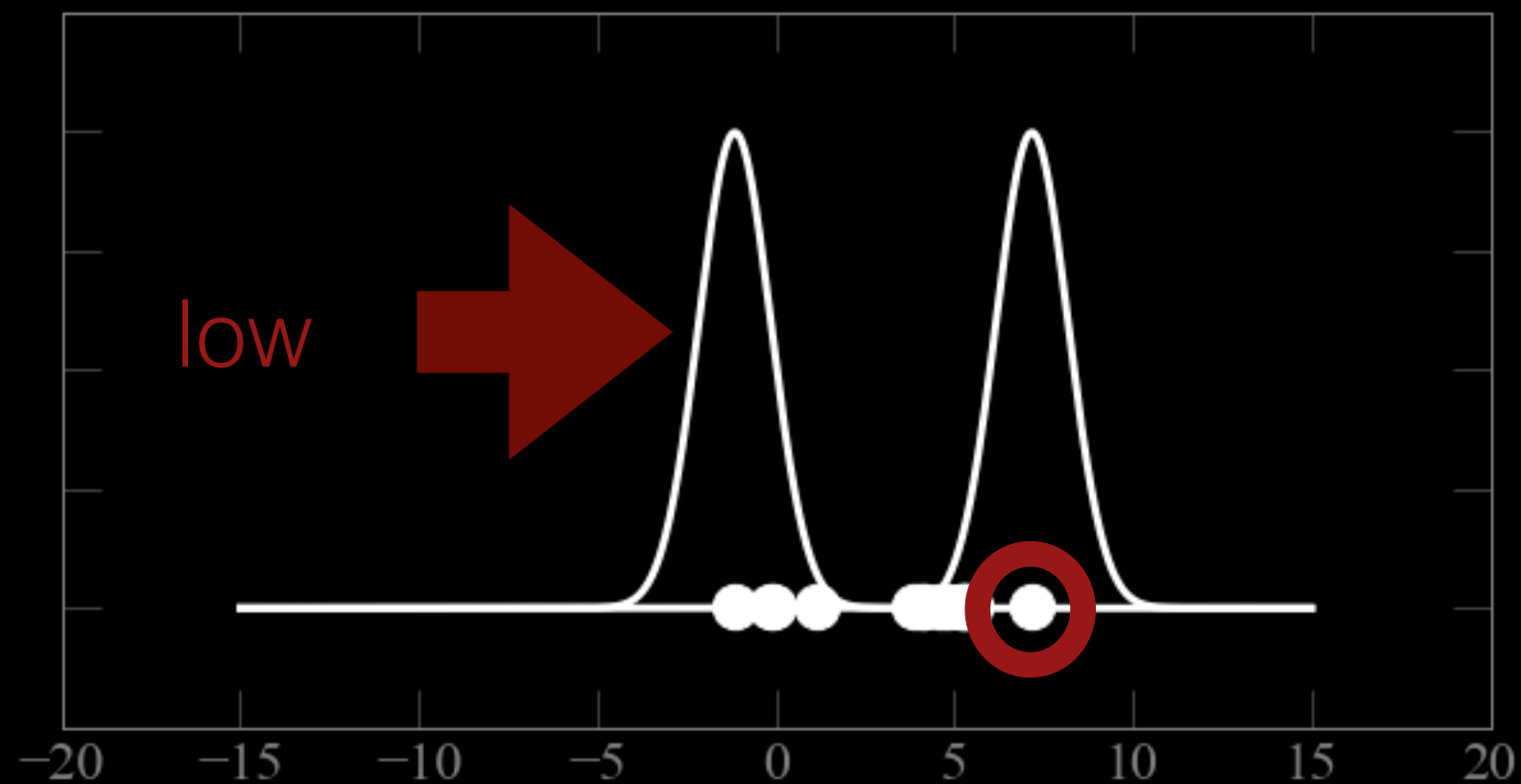


for every point calculate the probability it comes from either gaussian

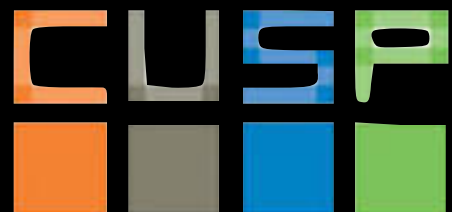


EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$

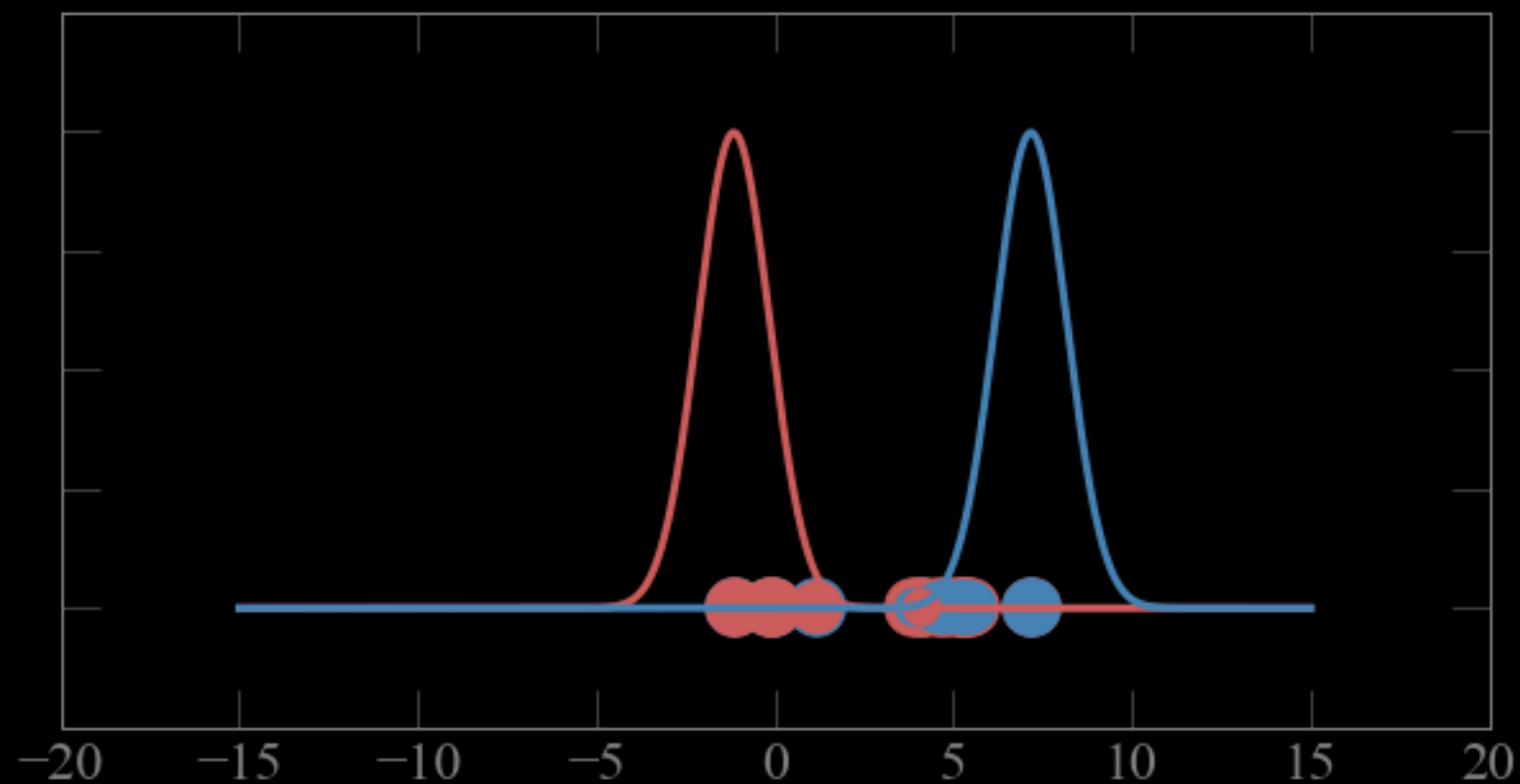


for every point calculate the probability it comes from either gaussian

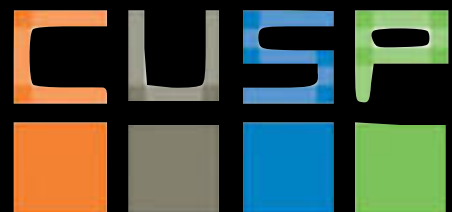


EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$



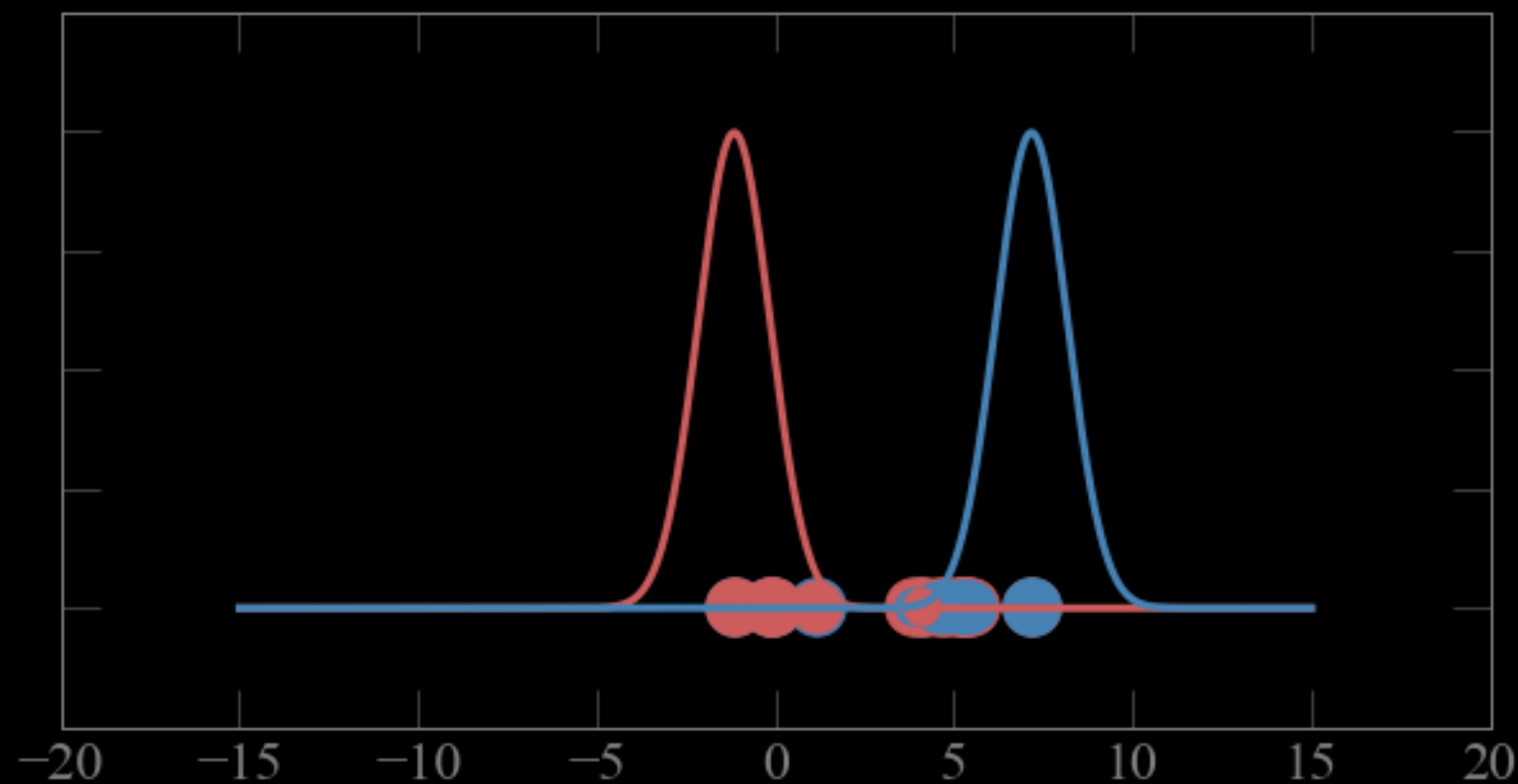
for every point calculate the probability it comes from either gaussian



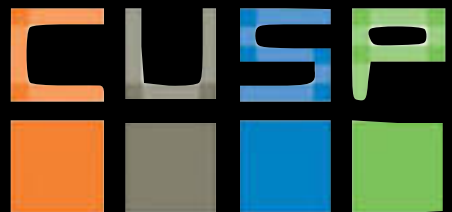
EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$

$$P(\mu_1, \sigma_1 | x_i) = \frac{P(x_i | \mu_1, \sigma_1)P(\mu_1, \sigma_1)}{P(x_i | \mu_1, \sigma_1)P(\mu_1, \sigma_1) + P(x_i | \mu_2, \sigma_2)P(\mu_2, \sigma_2)}$$



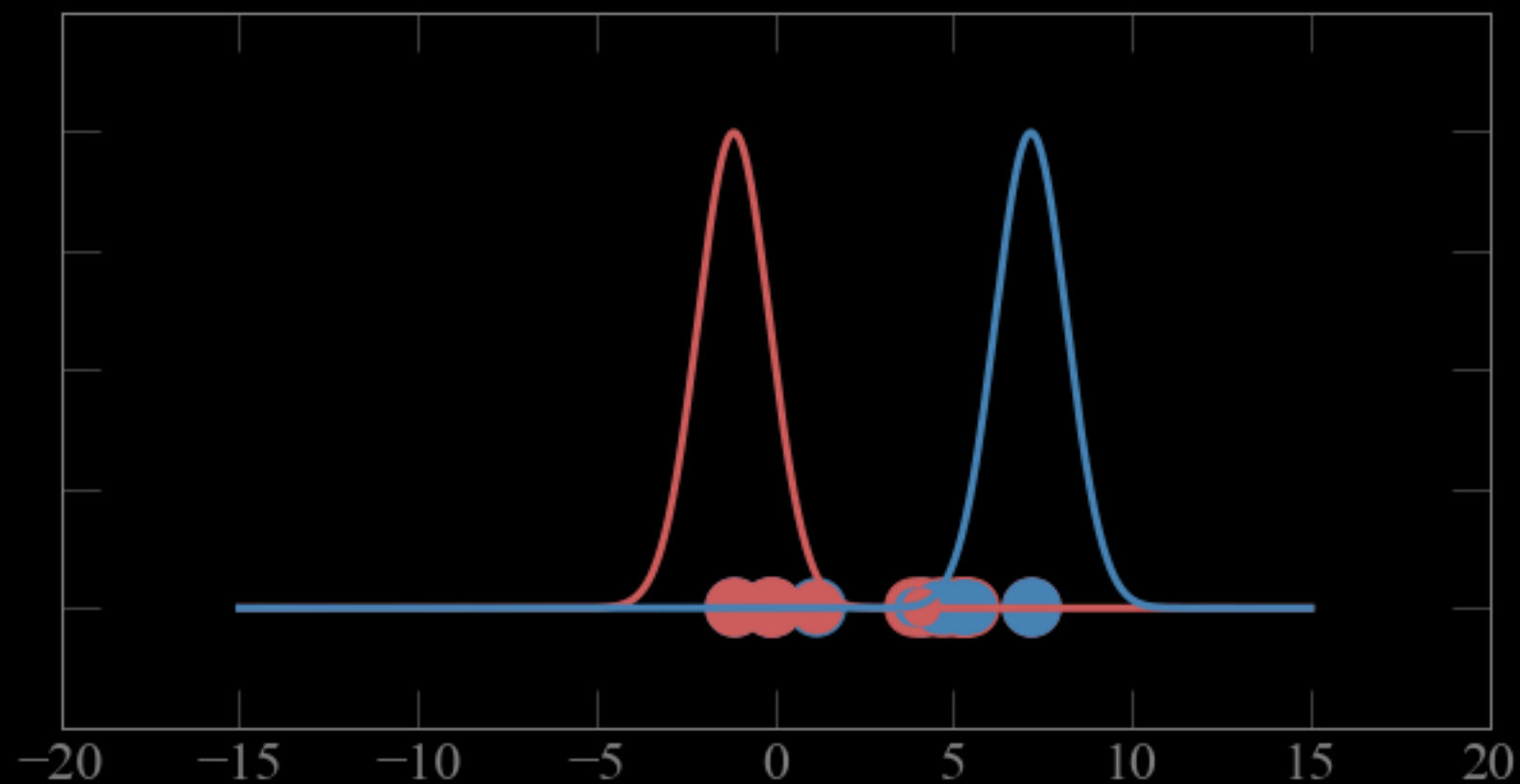
for every point calculate the probability it comes from either gaussian



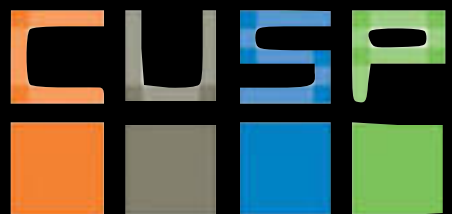
EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$

$$P(g_1 | x_i) = \frac{P(x_i | g_1)P(g_1)}{P(x_i | g_1)P(g_1) + P(x_i | g_2)P(g_2)}$$



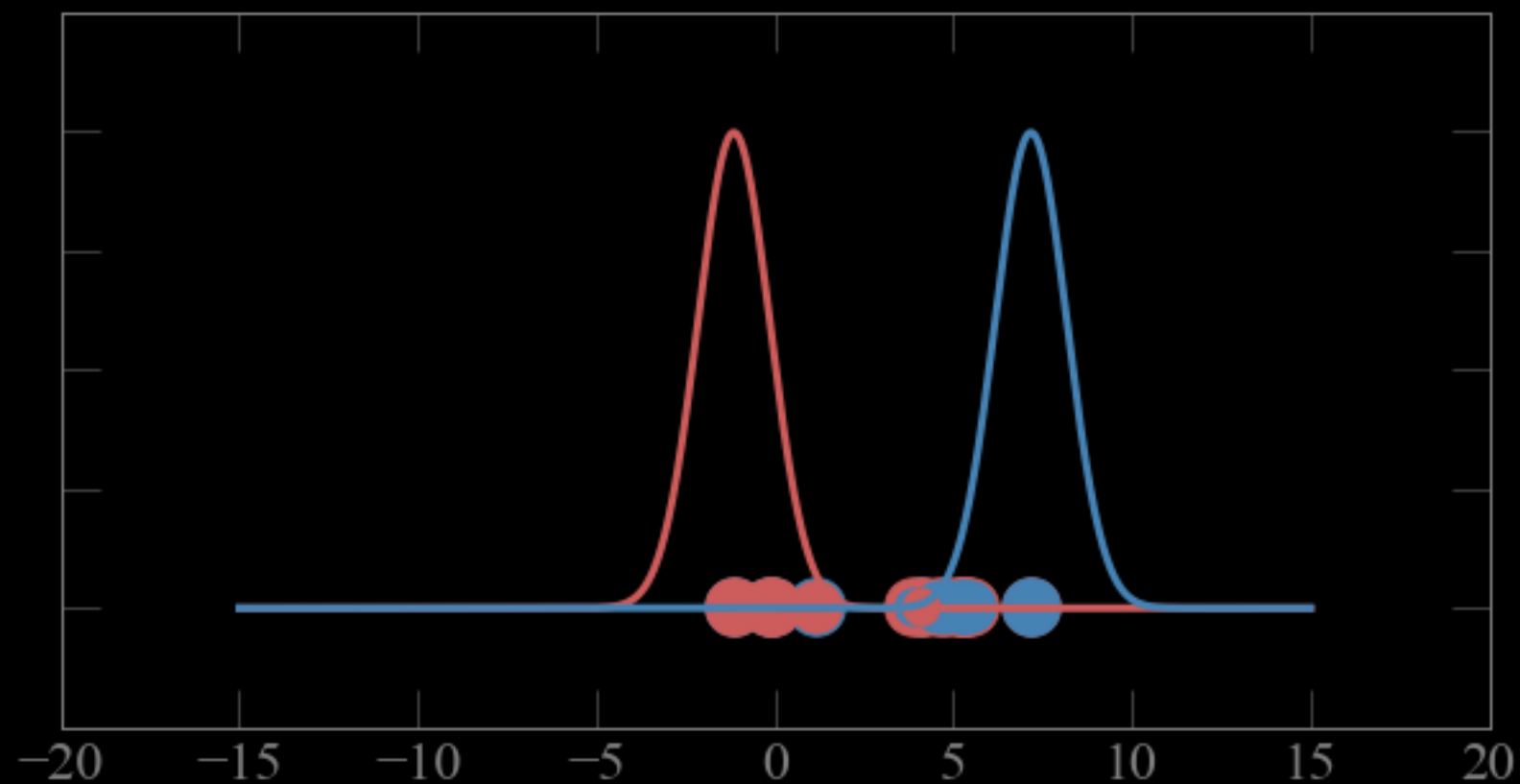
for every point calculate the probability it comes from either gaussian



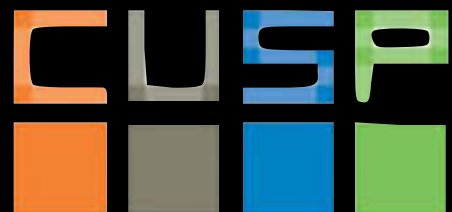
EM

Bayes Theorem!

$$P(x|\alpha)P(\alpha) = P(x|\beta)P(\beta)$$



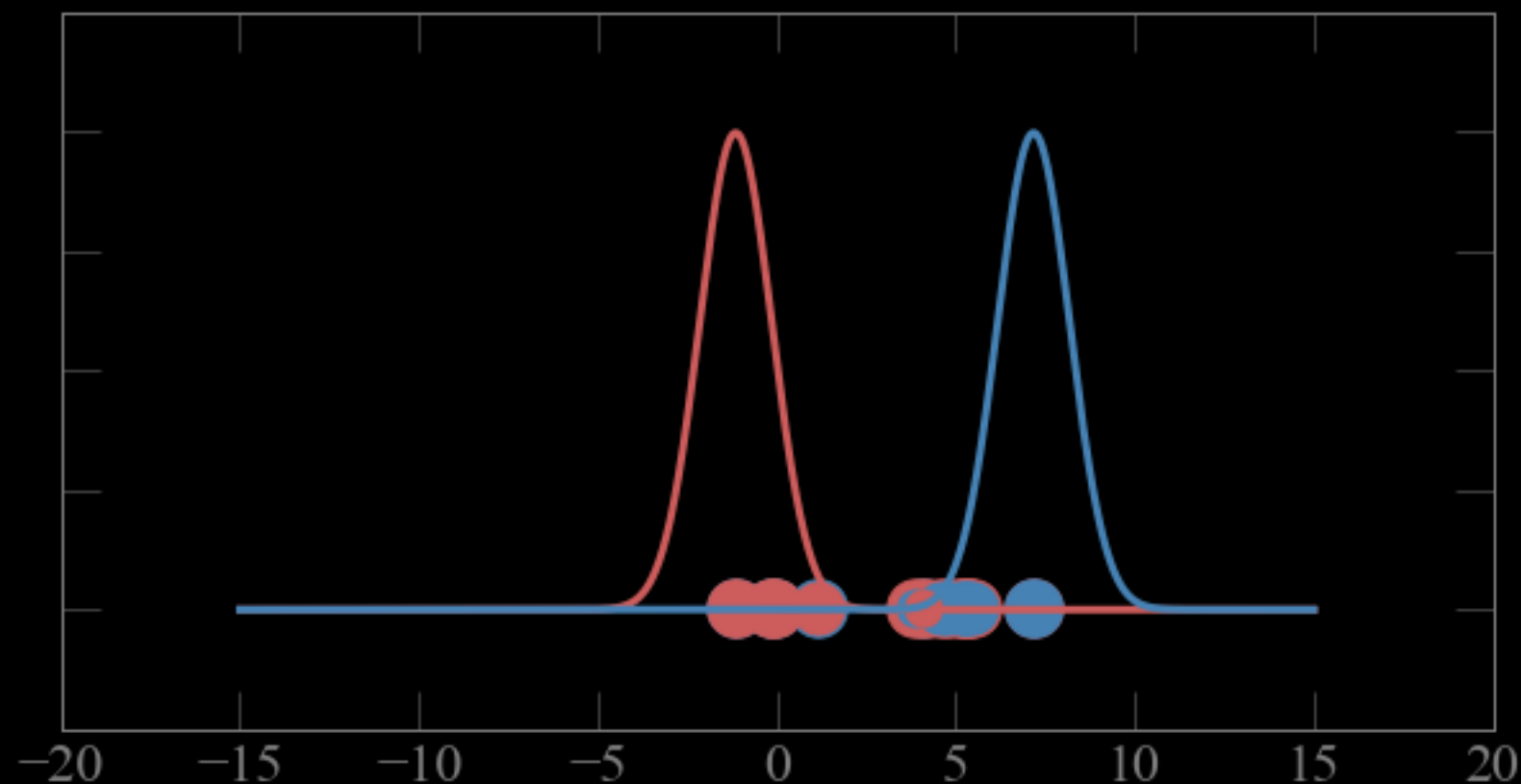
for every point calculate the probability it comes from either gaussian



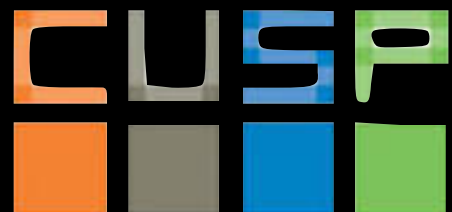
EM

$$P(x_i | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{x_i - \mu_j}{2\sigma_j^2}\right)$$

$$p_{ji} = P(g_1 | x_i) = \frac{P(x_i | g_1)P(g_1)}{P(x_i | g_1)P(g_1) + P(x_i | g_2)P(g_2)}$$

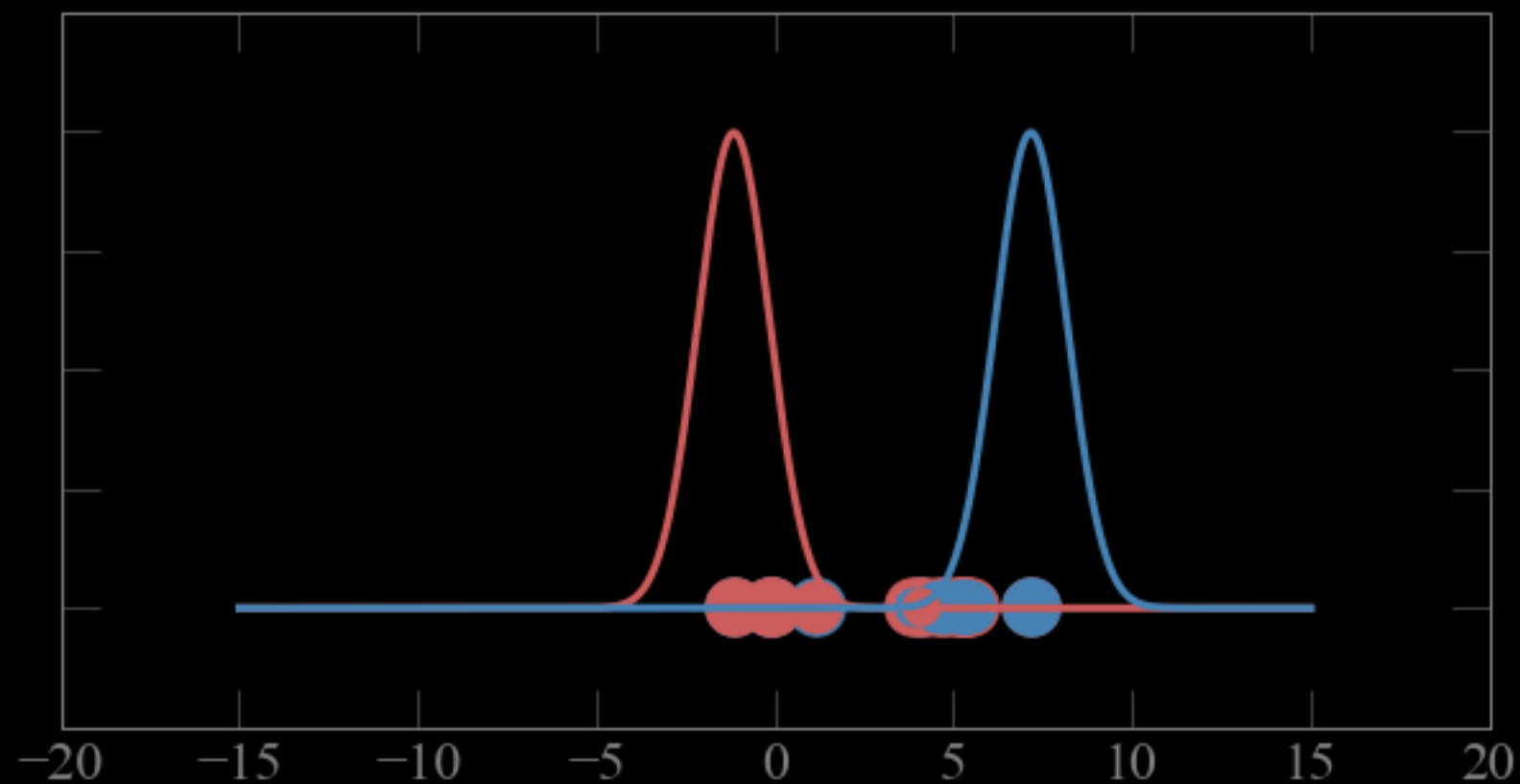


calculate the weighted mean of the cluster,
weighted by the p_{ji}

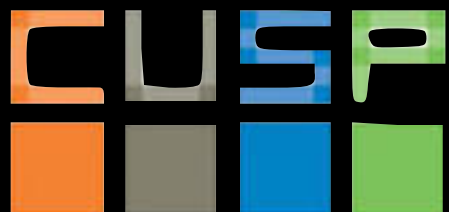


EM

$$\mu_i = \frac{\sum_j P(g_i | x_j) x_j}{\sum_j P(g_i | x_j)}$$

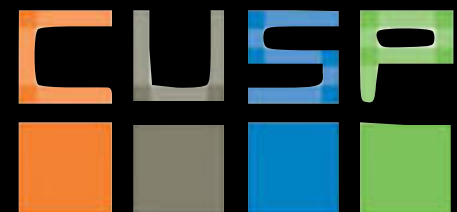
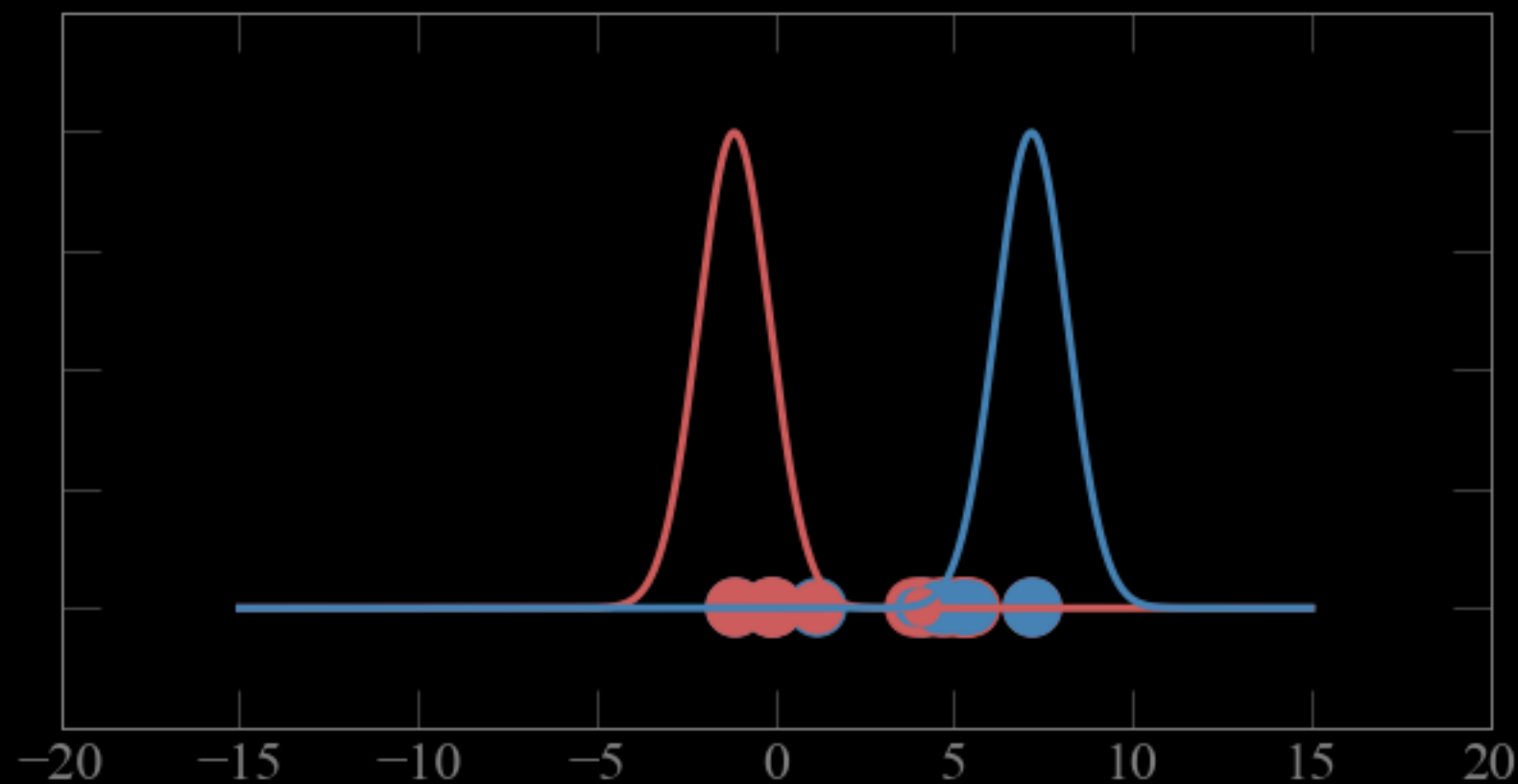


calculate the weighted mean of the cluster,
weighted by the p_{ji}



EM

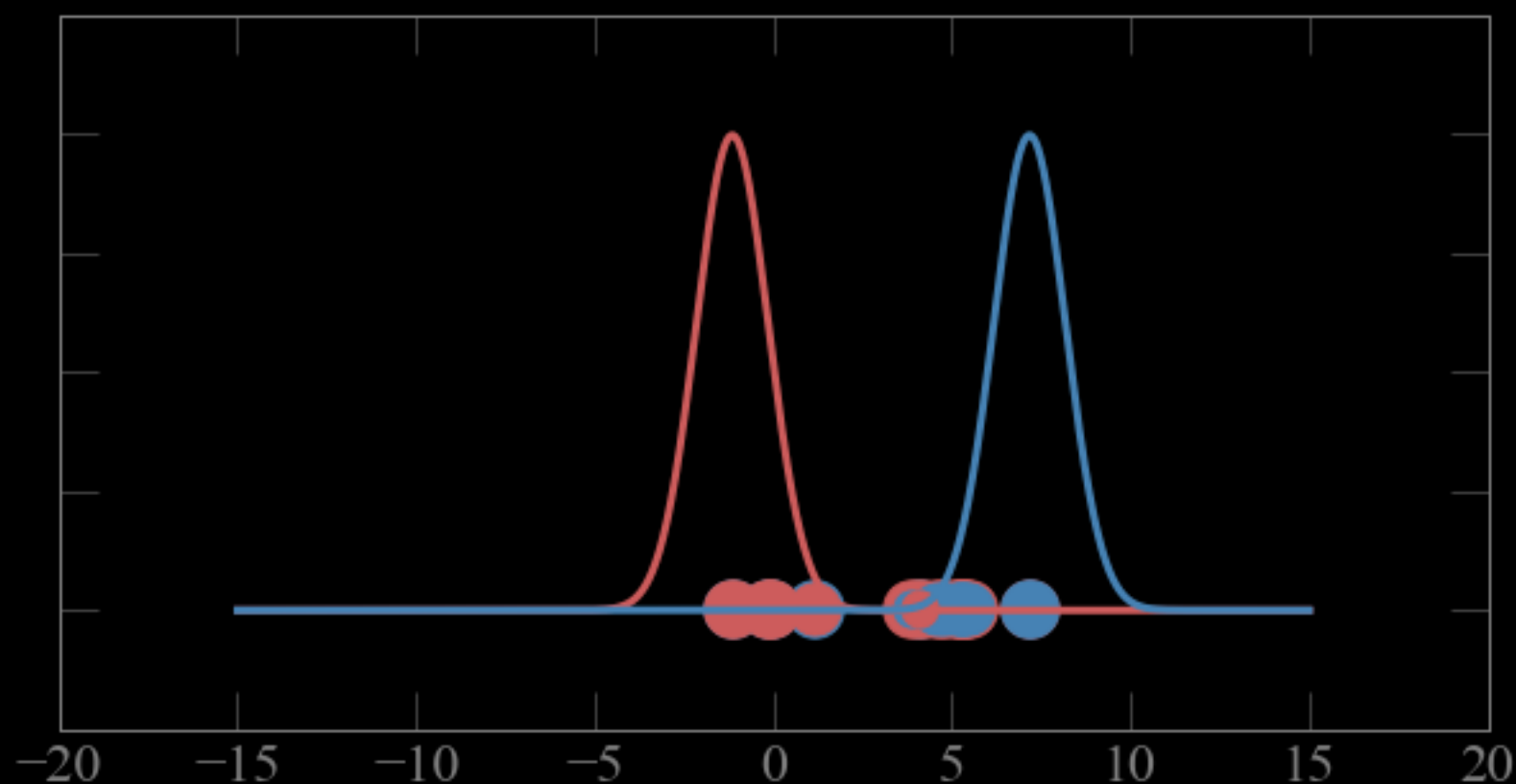
$$\mu_i = \frac{\sum_j P(g_i | x_j) x_j}{\sum_j P(g_i | x_j)} \quad \sigma_j = \frac{\sum_i P(g_j | x_i) (x_i - \mu_j)^2}{\sum_i P(g_j | x_i)}$$



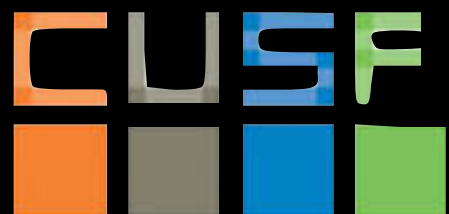
calculate the weighted sigma of the cluster,
weighted by the p_{ji}

EM

$$\mu_i = \frac{\sum_j P(g_i | x_j) x_j}{\sum_j P(g_i | x_j)} \quad \sigma_j = \frac{\sum_i P(g_j | x_i) (x_i - \mu_j)^2}{\sum_i P(g_j | x_i)}$$



$$P(g_1 | x_i) = \frac{P(x_i | g_1) P(g_1)}{P(x_i | g_1) P(g_1) + P(x_i | g_2) P(g_2)}$$

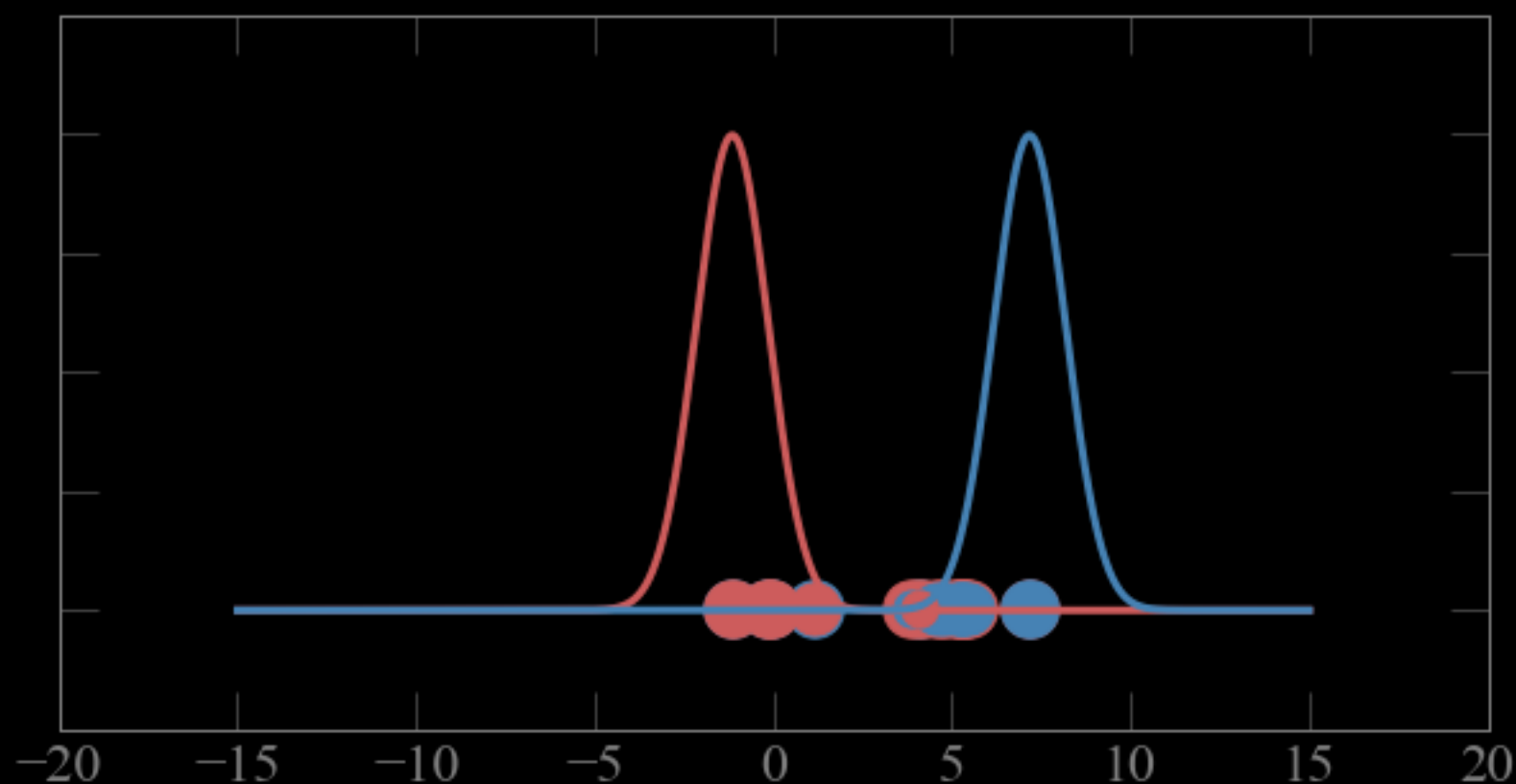


calculate the new p_{ji} ... rinse, repeat

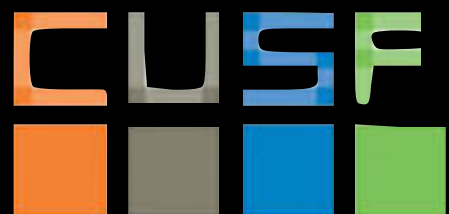
XI: Clustering

EM

$$\mu_i = \frac{\sum_j P(g_i | x_j) x_j}{\sum_j P(g_i | x_j)} \quad \sigma_j = \frac{\sum_i P(g_j | x_i) (x_i - \mu_j)^2}{\sum_i P(g_j | x_i)}$$



$$P(g_1 | x_i) = \frac{P(x_i | g_1) P(g_1)}{P(x_i | g_1) P(g_1) + P(x_i | g_2) P(g_2)}$$

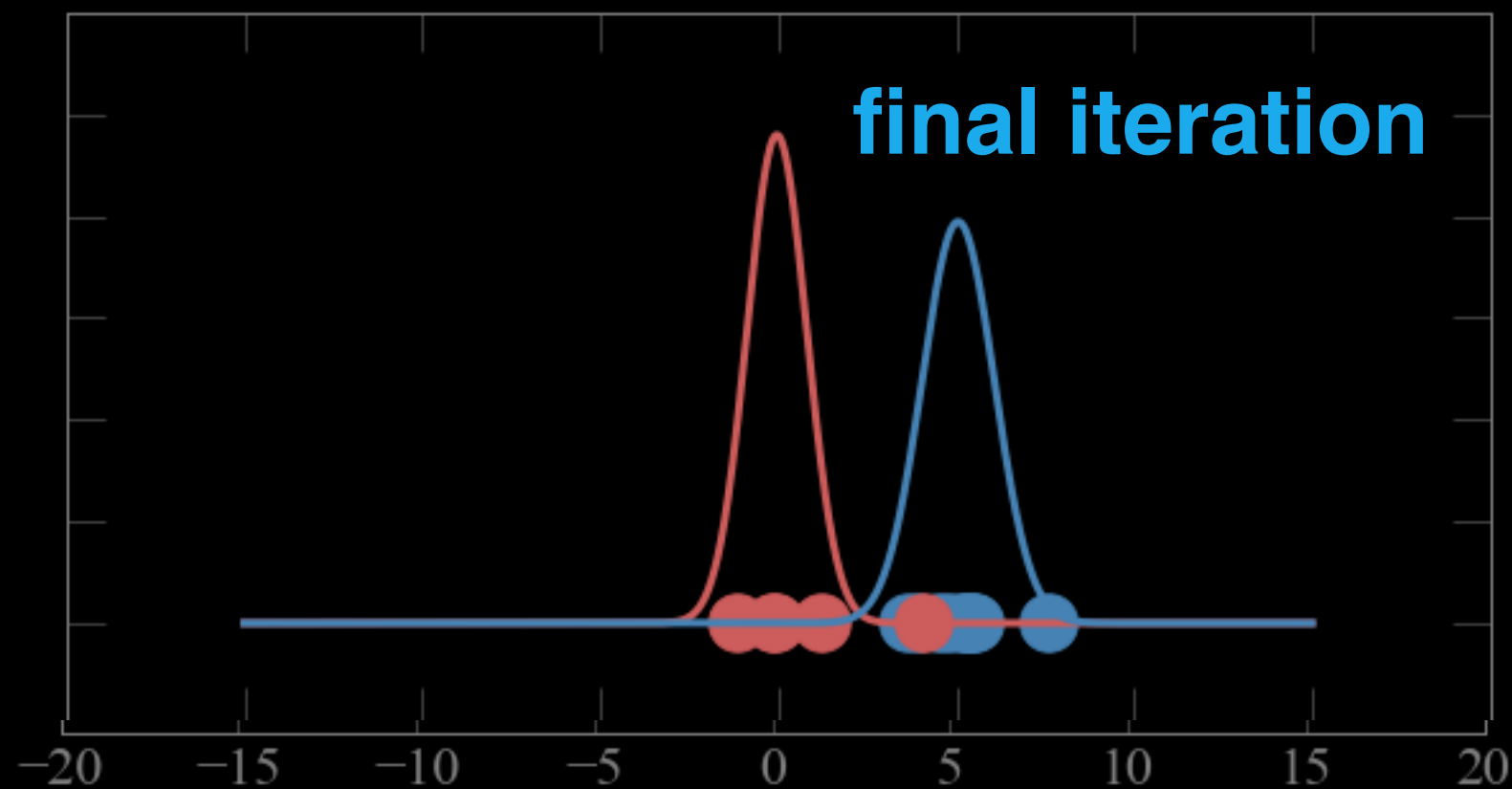


calculate the new p_{ji} ... rinse, repeat

XI: Clustering

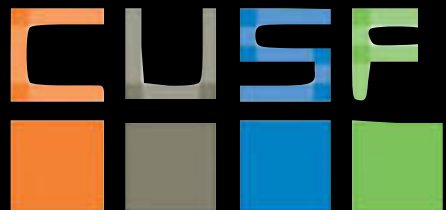
EM

$$\mu_i = \frac{\sum_j P(g_i | x_j) x_j}{\sum_j P(g_i | x_j)} \quad \sigma_j = \frac{\sum_i P(g_j | x_i) (x_i - \mu_j)^2}{\sum_i P(g_j | x_i)}$$



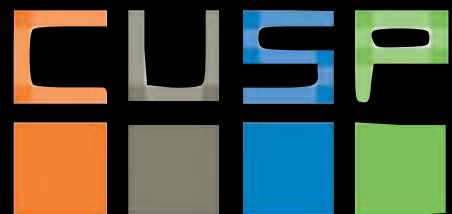
$$P(g_1 | x_i) = \frac{P(x_i | g_1)P(g_1)}{P(x_i | g_1)P(g_1) + P(x_i | g_2)P(g_2)}$$

... till it converges



Expectation Maximization:

1. **Choose N “centers” guesses:** like in K-means
2. **Calculate the probability of each distribution given the point (Expectation step)**
3. **Calculate the new centers and variances as weighted averages of the datapoints, weighted by the probabilities**
4. **Iterate 2&3 till convergence:** when gaussian parameters no longer change



Expectation Maximization:

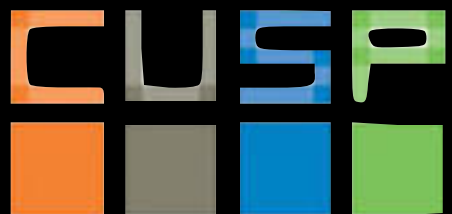
Order: #clusters #dimensions #iterations #datapoints
#parameter $O(KdNp)$

based on Bayes theorem

Its non-deterministic: the result depends on the
(random) starting point

**It only works where a probability distribution for the
data points can be defines (or equivalently a
likelihood)**

**Must declare the number of clusters and the shape of
the pdf upfront**



Clustering methods

- **Partitioning**

- Hard clustering**

- K-means (McQueen '67)

- K-medoids (Kaufman & Rausseeuw '87)

- Soft Clustering**

- Expectation Maximization (Dempster, Laird, Rubin '77)

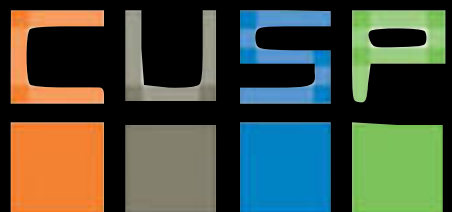
- **Hierarchical**
agglomerative

- divisive

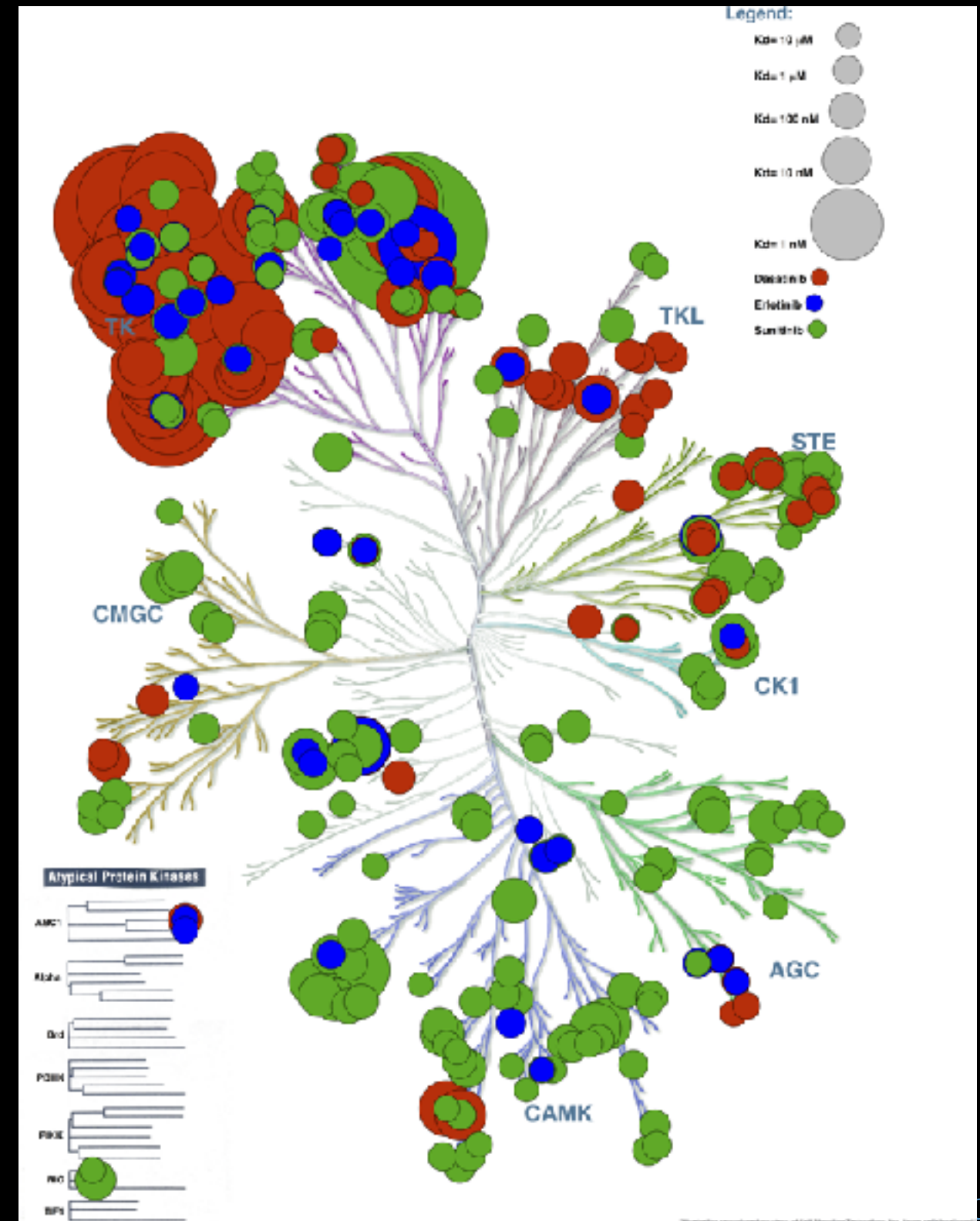
- **also:**
 - **Density based**

- **Grid based**

- **Model based**

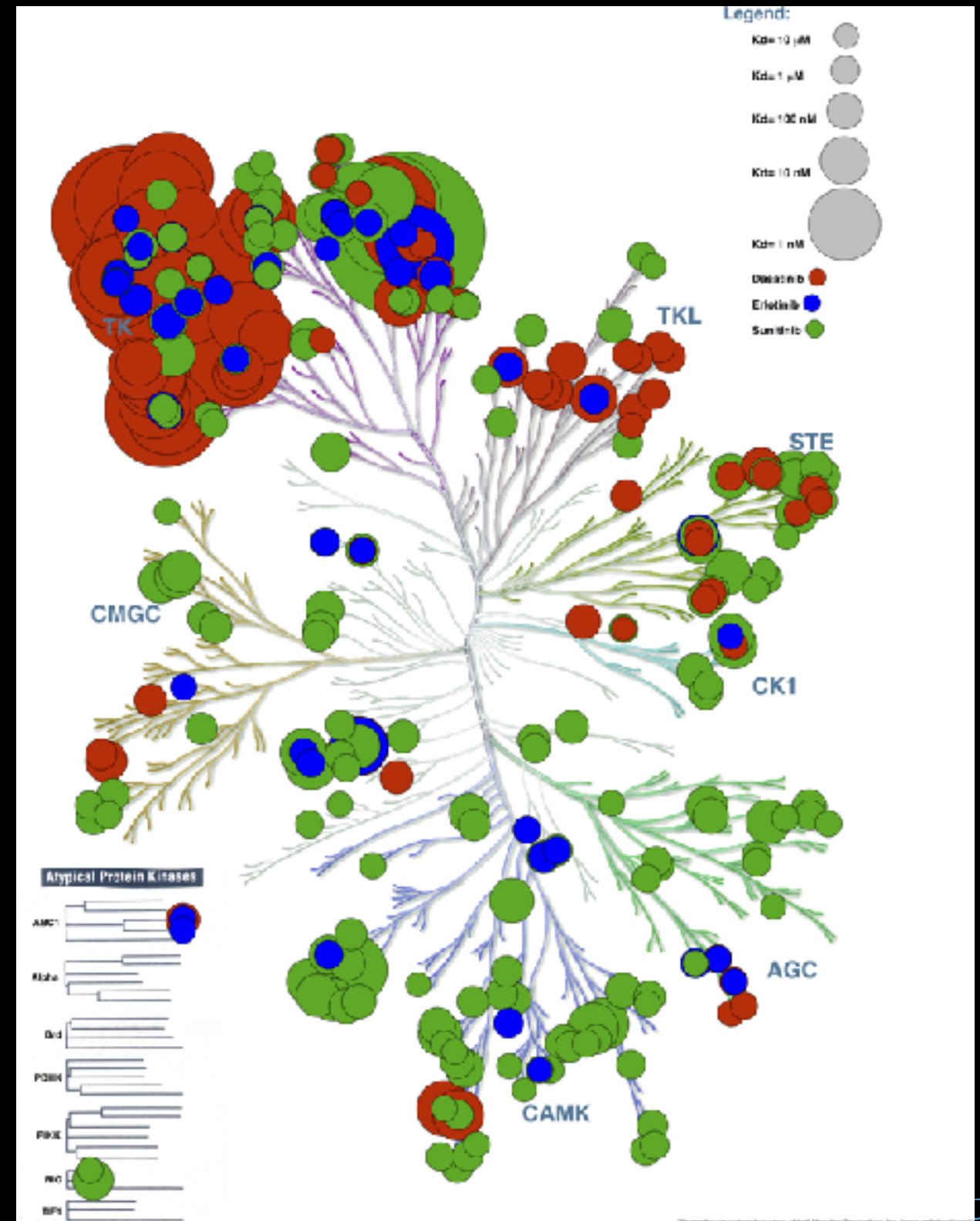


hierarchical clustering



hierarchical clustering

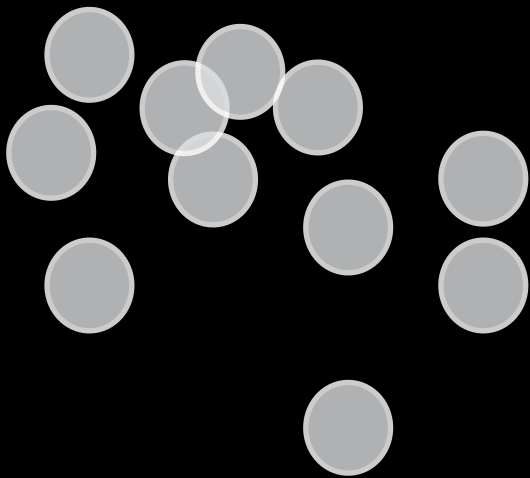
removes the issue of
deciding K (number of
clusters)



hierarchical clustering

devisive (*top-down*):

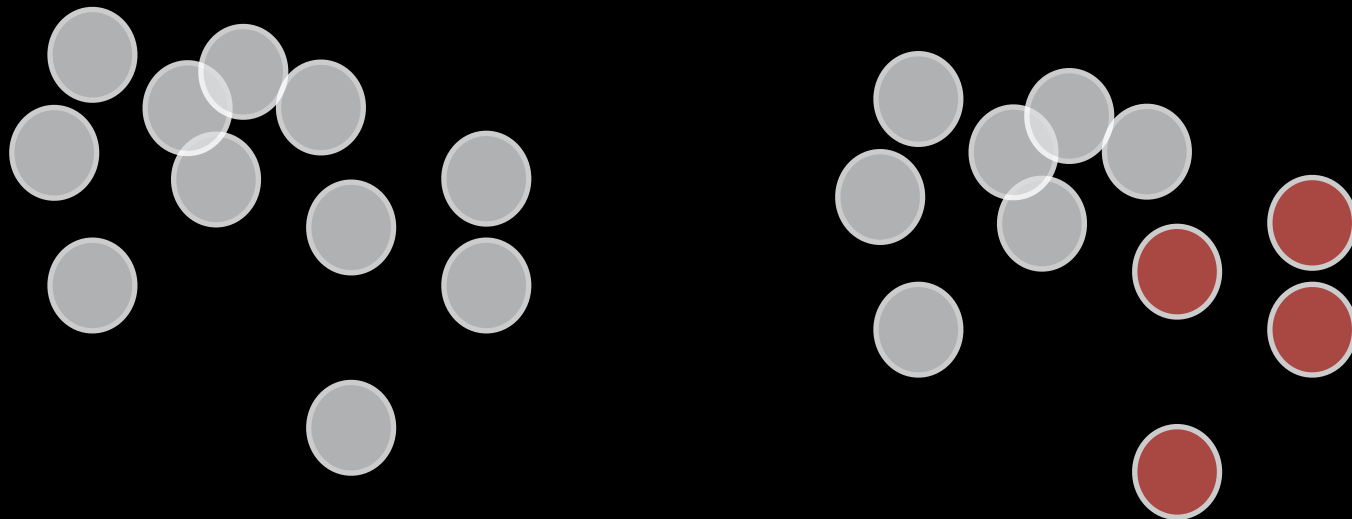
e.g. hierarchical k-mean



hierarchical clustering

devisive (*top-down*):

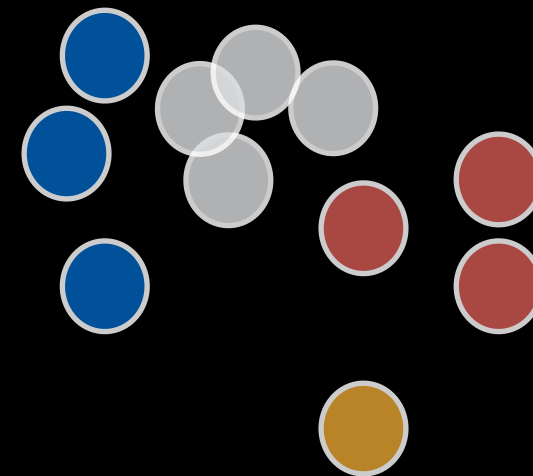
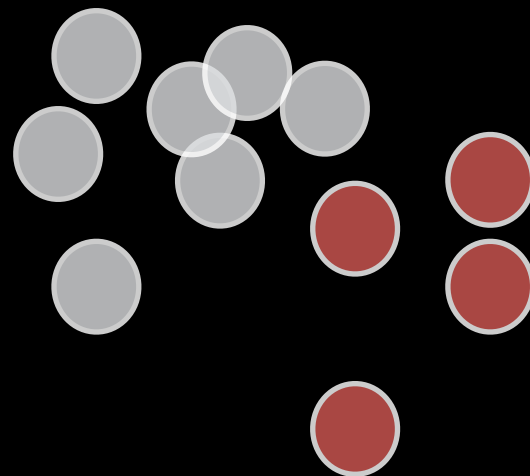
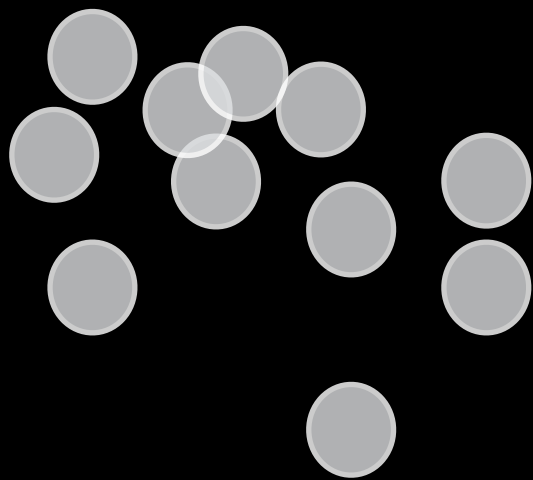
e.g. hierarchical k-mean



hierarchical clustering

devisive (*top-down*):

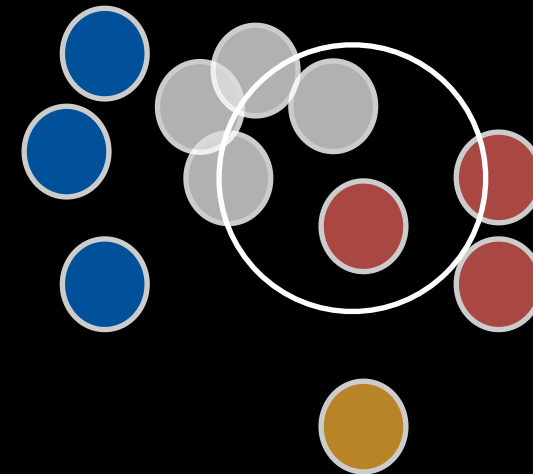
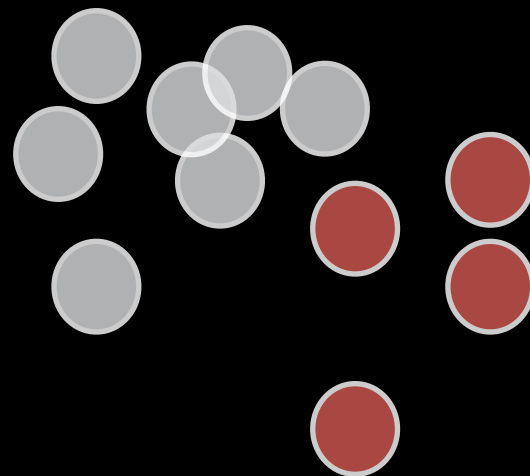
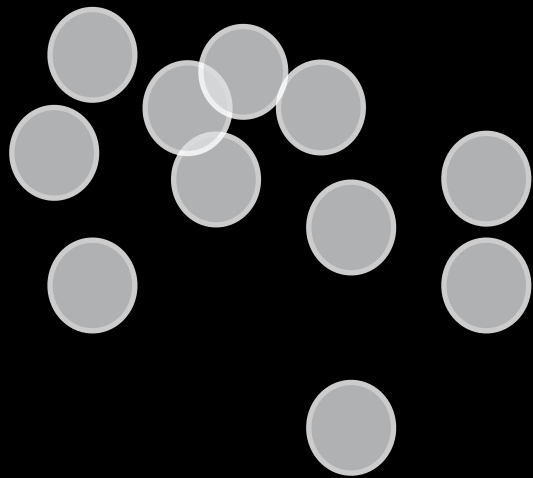
e.g. hierarchical k-mean



hierarchical clustering

devisive (*top-down*):

e.g. hierarchical k-mean



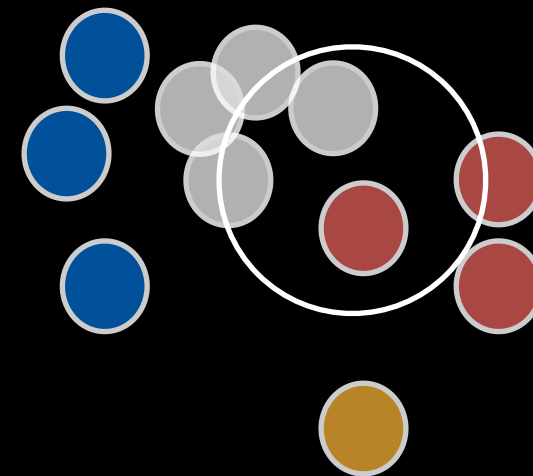
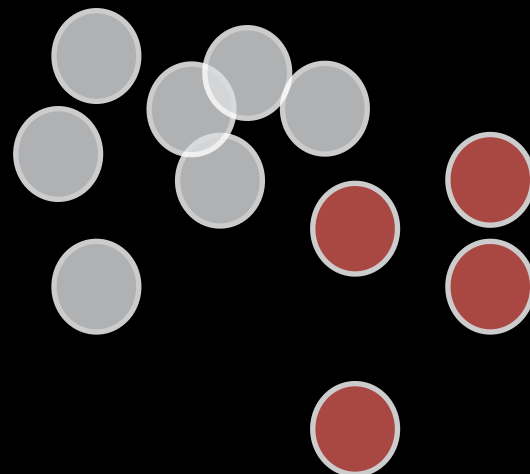
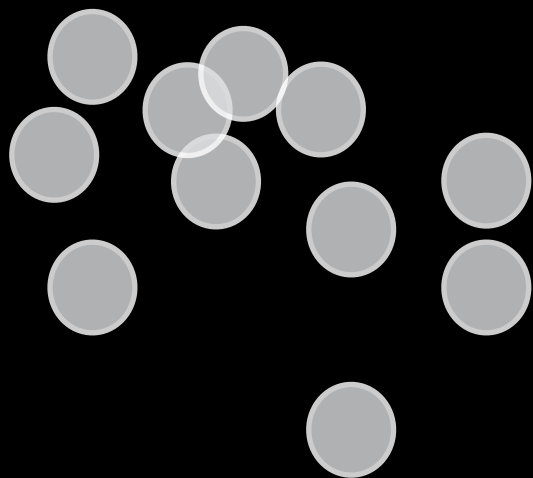
it is
non-deterministic

it is *greedy* -
just as k-means
two nearby points
may end up in
separate clusters

hierarchical clustering

devisive (top-down):

e.g. hierarchical k-mean

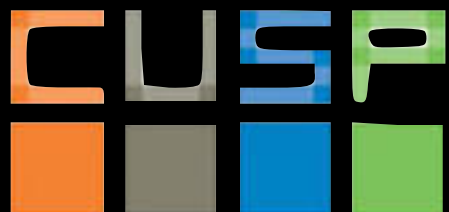


it is
non-deterministic

it is *greedy* -
just as k-means
two nearby points
may end up in
separate clusters

it is simple and
fast:
complexity

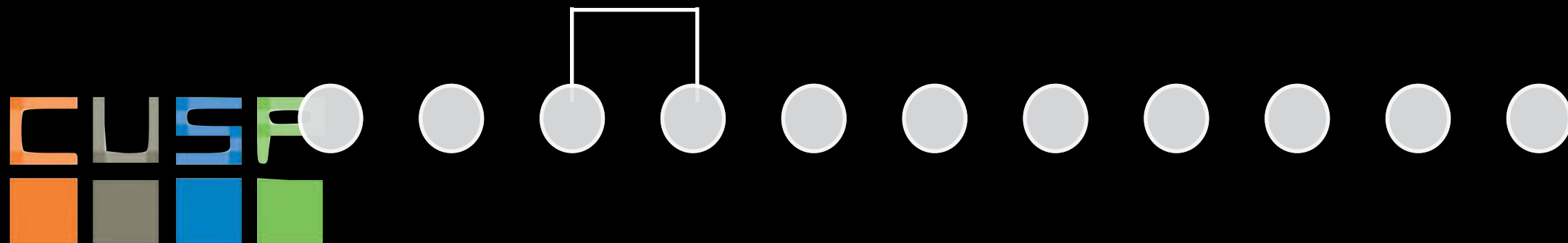
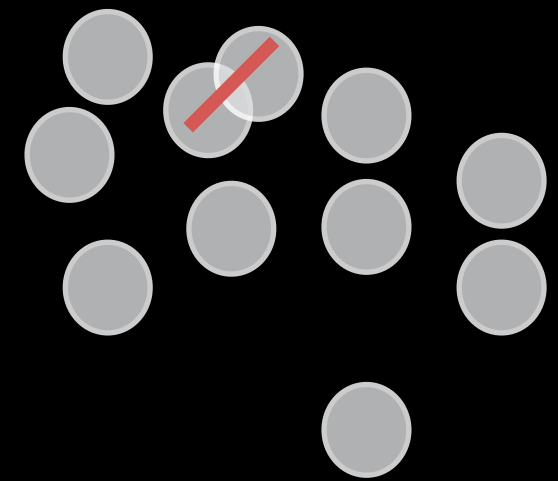
$O(NdK \log_k N)$



hierarchical clustering

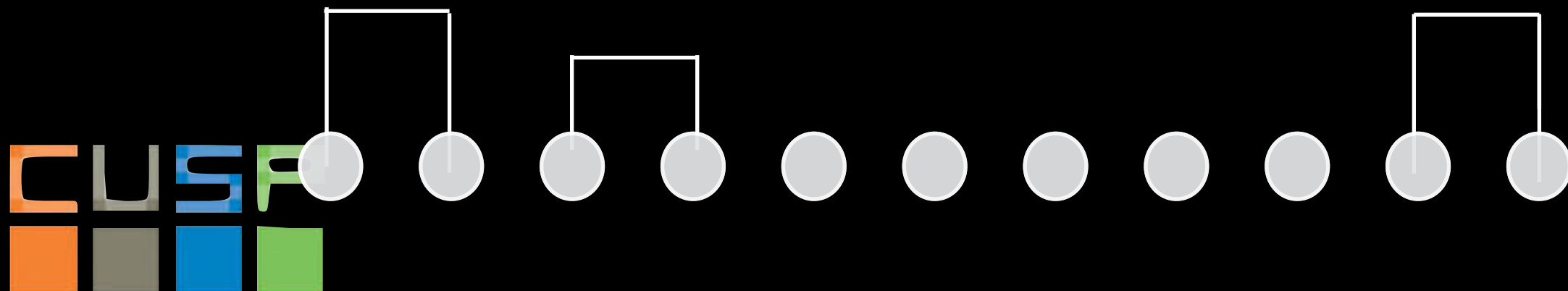
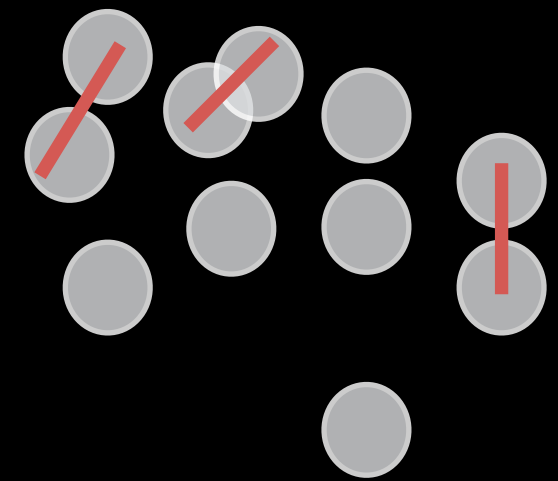
agglomerative

bottom-up



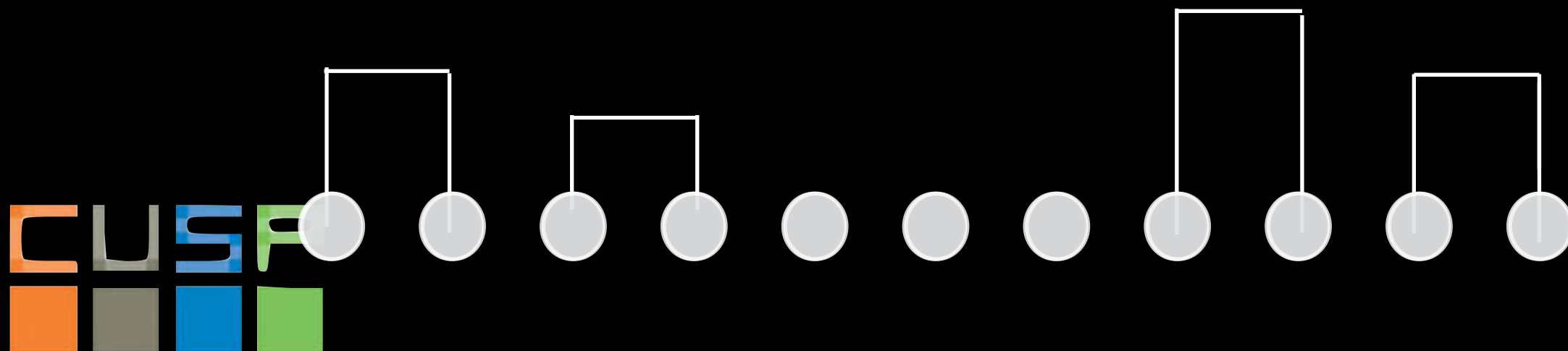
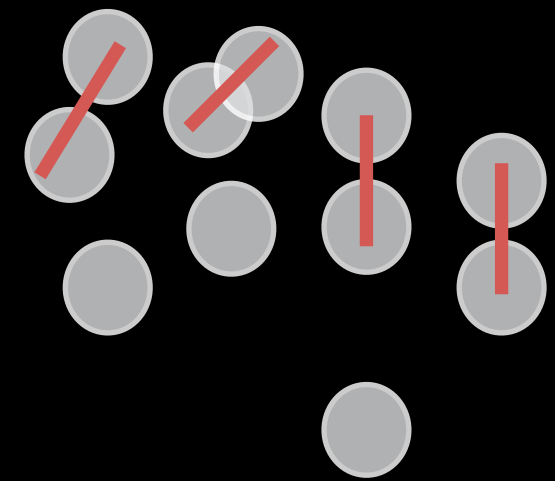
hierarchical clustering

agglomerative
bottom-up



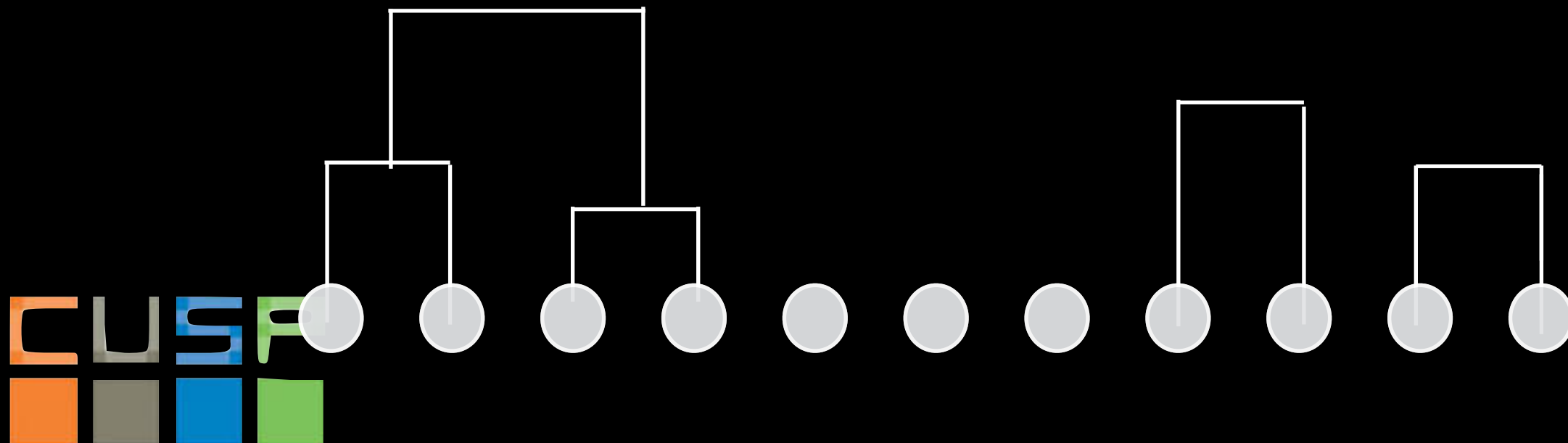
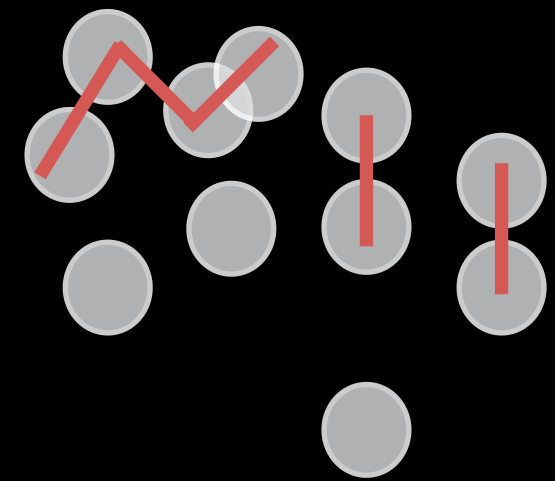
hierarchical clustering

agglomerative
bottom-up



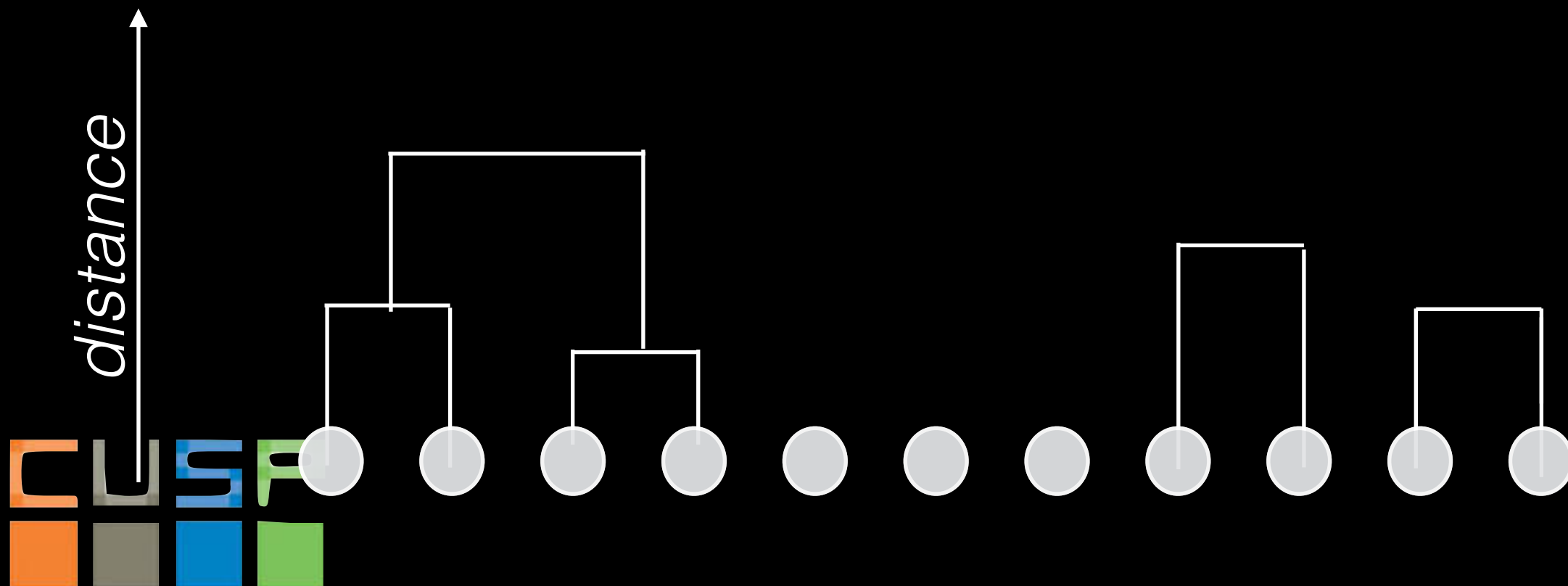
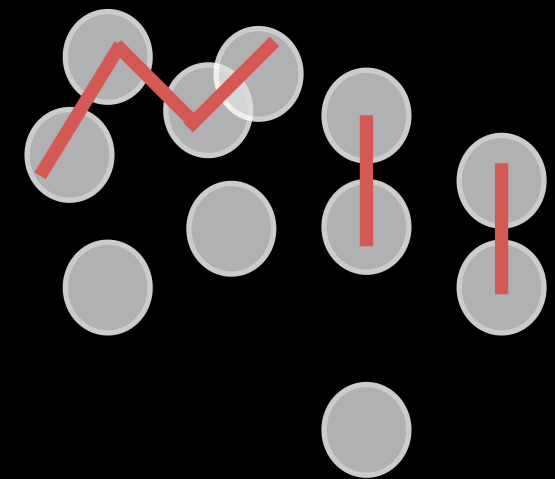
hierarchical clustering

agglomerative
bottom-up



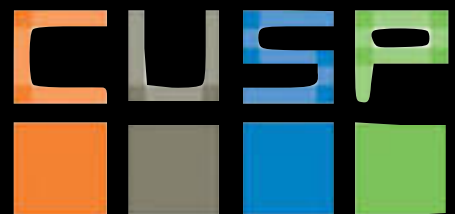
hierarchical clustering

agglomerative
bottom-up





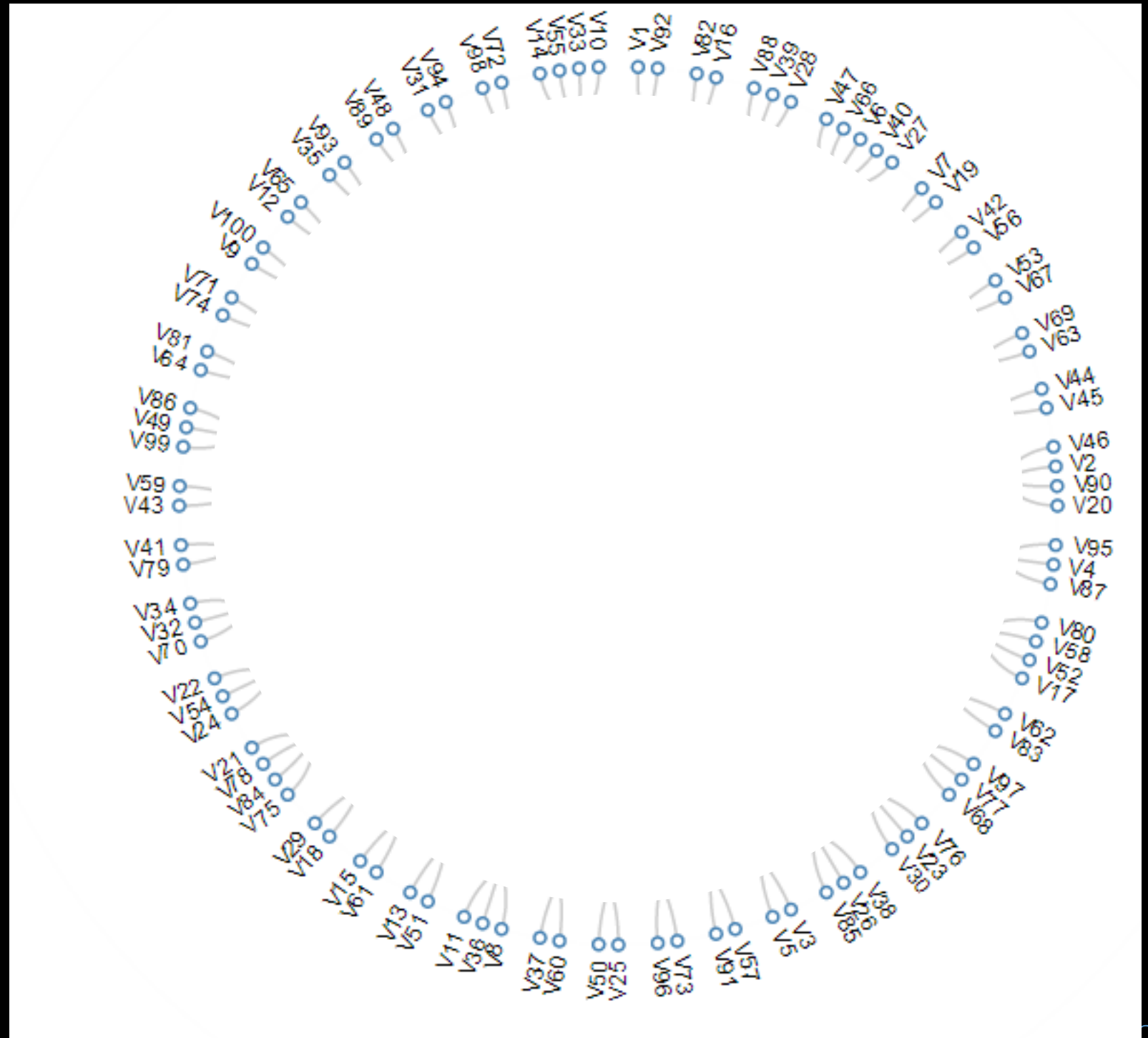
[https://github.com/fedhere/Ulnotebooks/blob/master/cluster/
thanksgivingClustering.ipynb](https://github.com/fedhere/Ulnotebooks/blob/master/cluster/thanksgivingClustering.ipynb)



hierarchical clustering

agglomerative
bottom-up

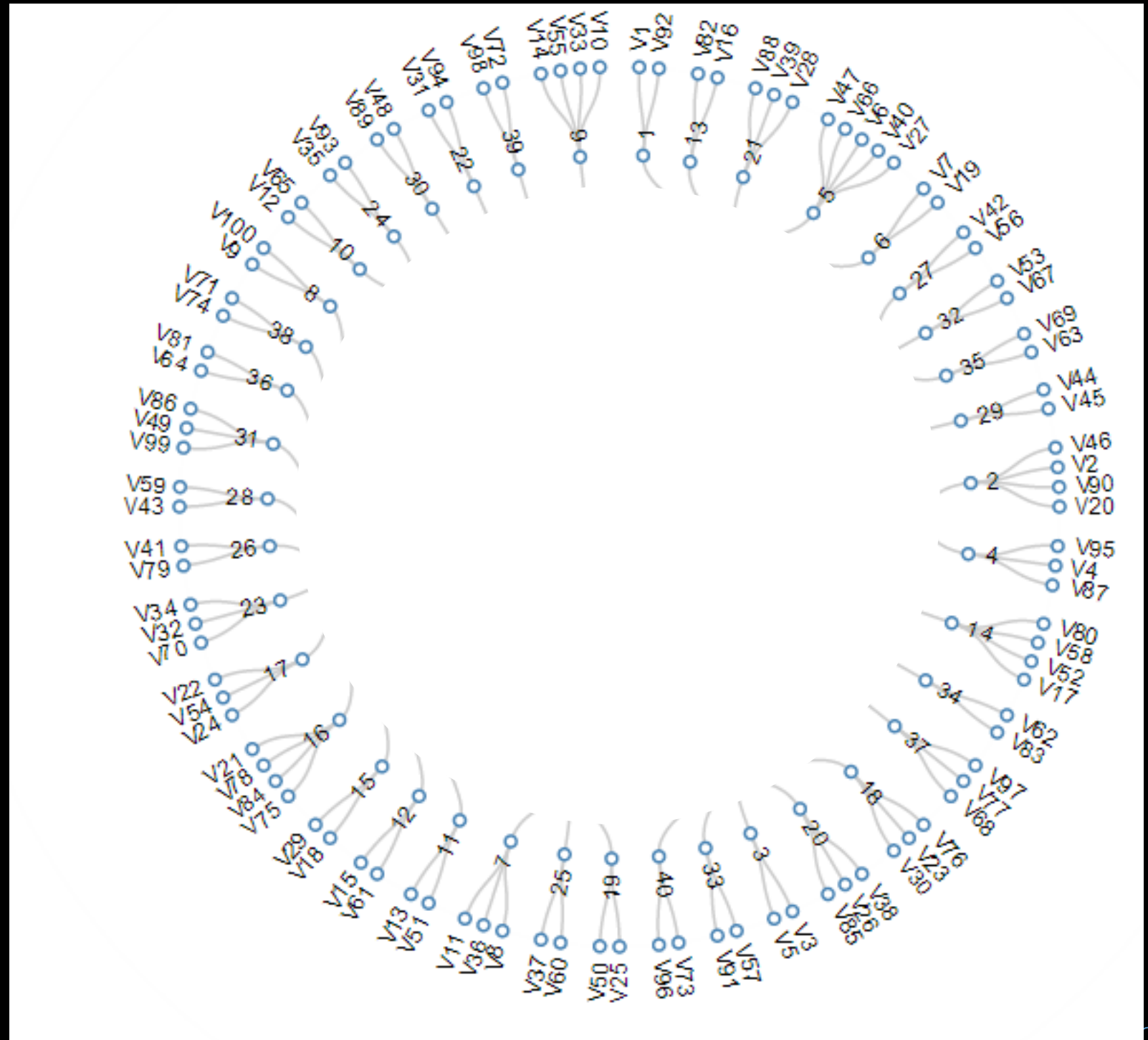
computationally
intense because
every **cluster pair**
distance has to be
calculate



hierarchical clustering

agglomerative
bottom-up

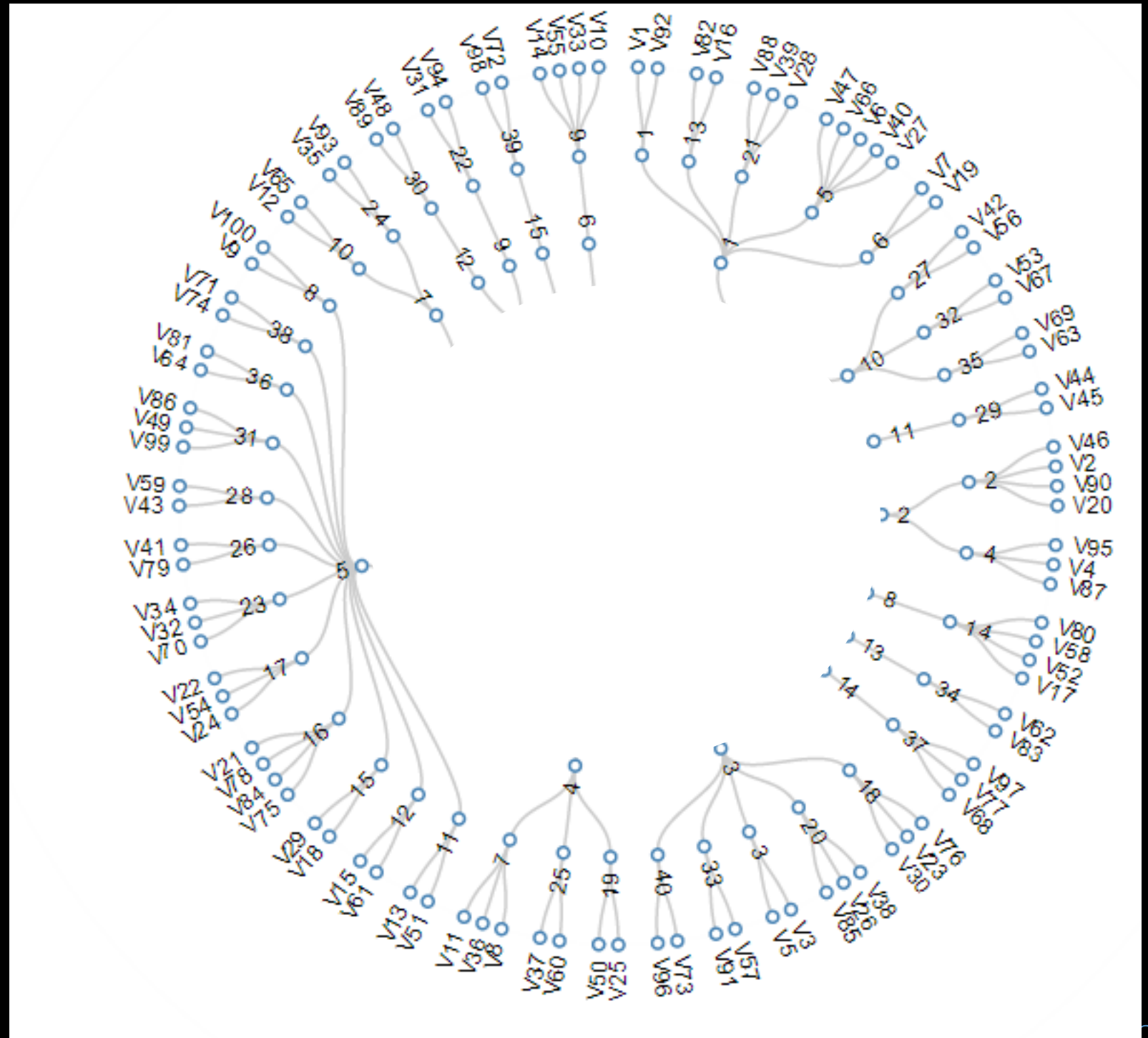
computationally
intense because
every **cluster pair**
distance has to be
calculate



hierarchical clustering

agglomerative
bottom-up

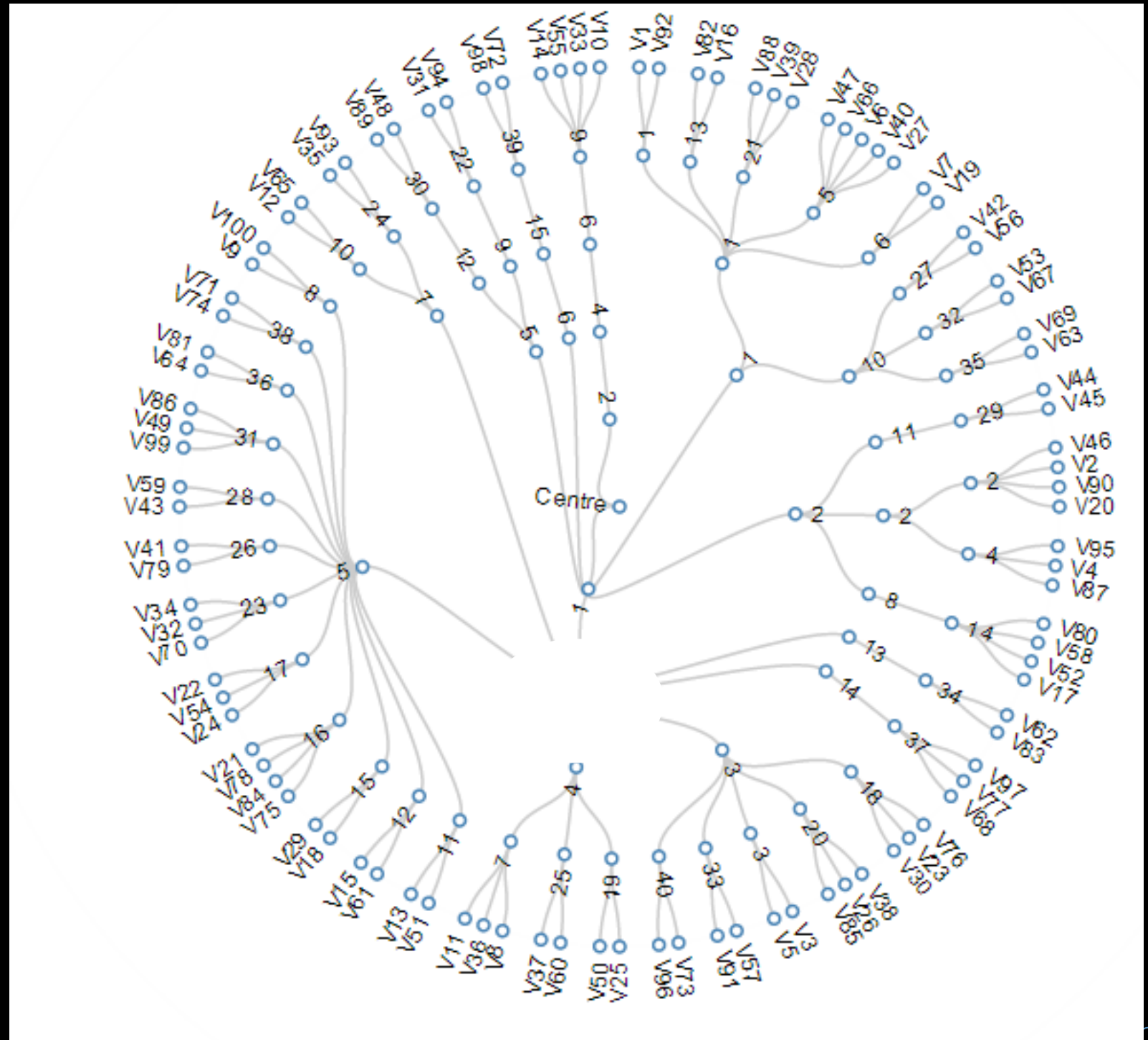
computationally
intense because
every **cluster pair**
distance has to be
calculate



hierarchical clustering

agglomerative
bottom-up

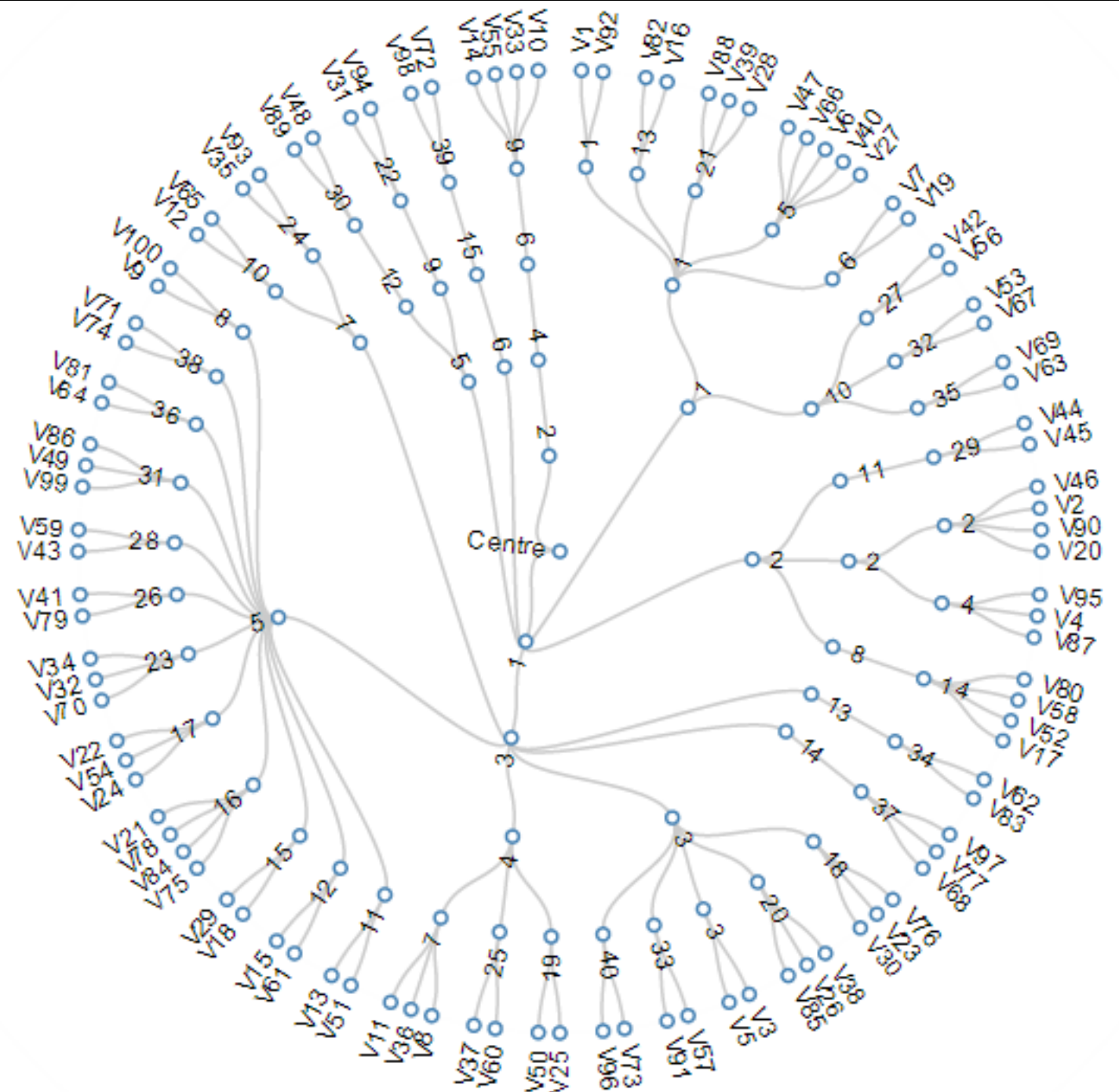
computationally
intense because
every **cluster pair**
distance has to be
calculate



hierarchical clustering

agglomerative
bottom-up

computationally
intense because
every **cluster pair**
distance has to be
calculate



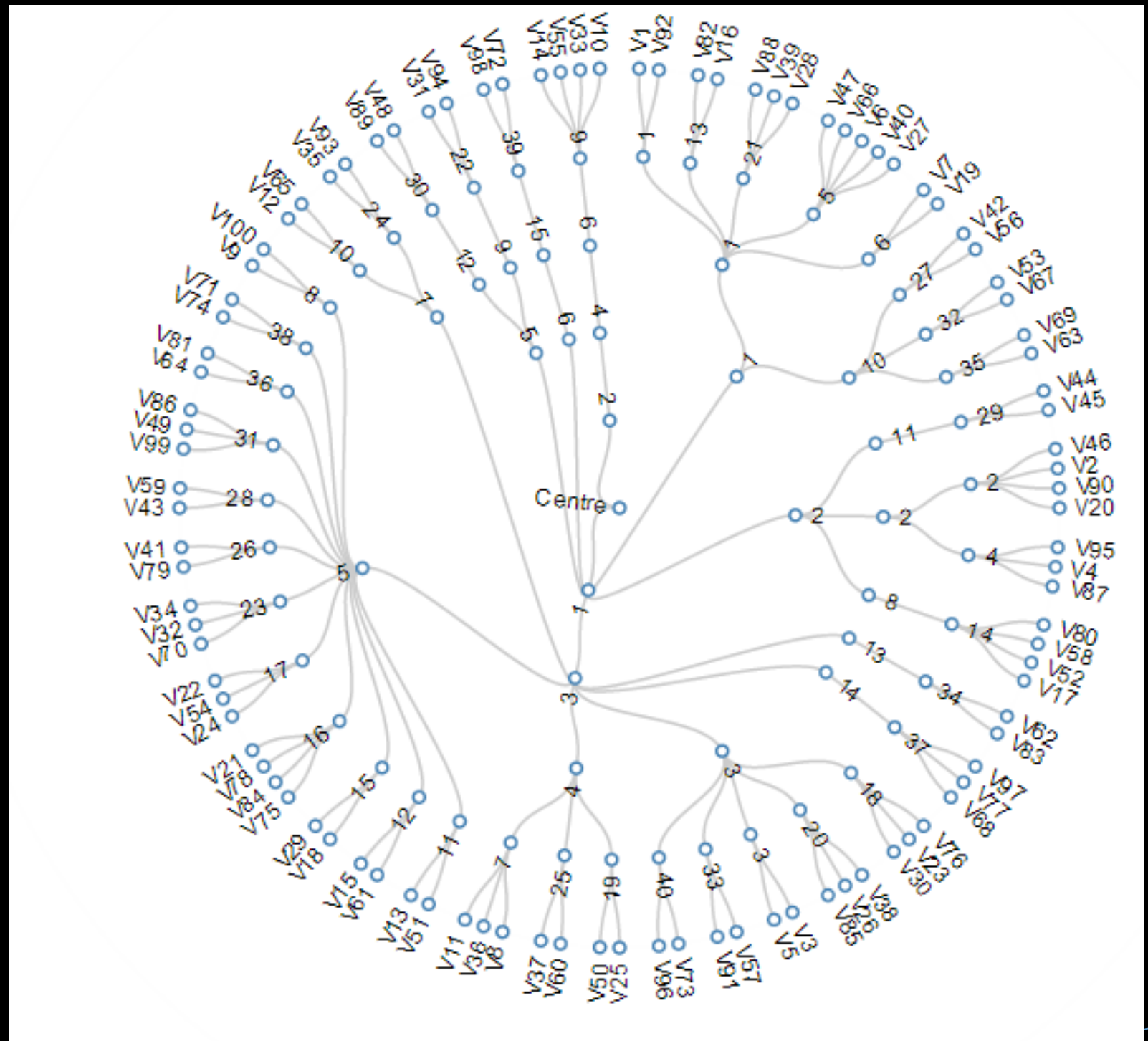
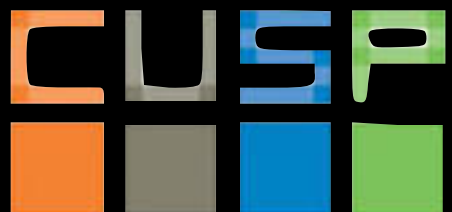
hierarchical clustering

agglomerative
bottom-up

computationally
intense because
every **cluster pair**
distance has to be
calculate

it is slow, though it
can be optimize:
complexity

$$O(N^2d + N^3)$$



hierarchical clustering

agglomerative

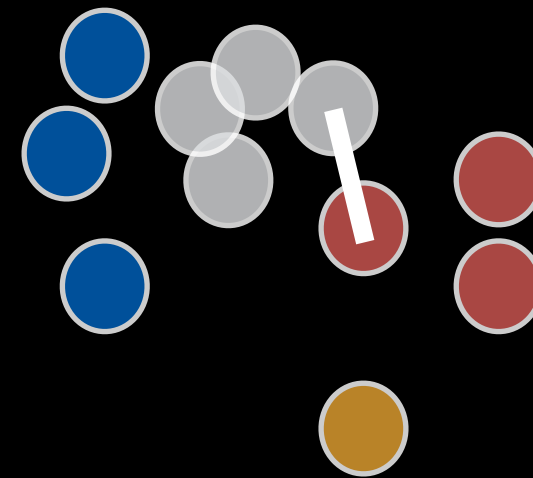
bottom-up

single link distance

$$D(c1, c2) = \min(D(x_{c1}, x_{c2}))$$

complete link distance

$$D(c1, c2) = \max(D(x_{c1}, x_{c2}))$$



hierarchical clustering

agglomerative

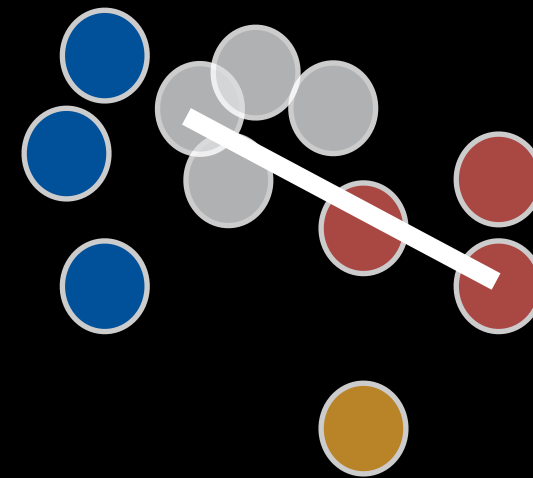
bottom-up

single link distance

$$D(c1, c2) = \min(D(x_{c1}, x_{c2}))$$

complete link distance

$$D(c1, c2) = \max(D(x_{c1}, x_{c2}))$$



hierarchical clustering

agglomerative

bottom-up

single link distance

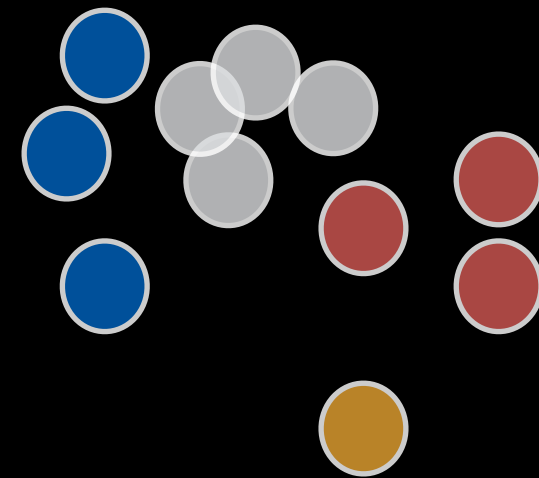
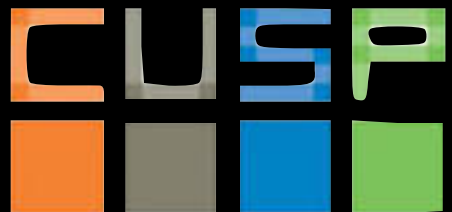
$$D(c1, c2) = \min(D(x_{c1}, x_{c2}))$$

complete link distance

$$D(c1, c2) = \max(D(x_{c1}, x_{c2}))$$

centroid distance

$$D(c1, c2) = \text{mean}(D(x_{c1}, x_{c2}))$$



ward distance

minimizes variance

$$D_{tot} = \sum_j \sum_{i, x_i \in C_j} (x_i - \mu_j)^2$$

Summary and Key concepts

clustering is easy, but interpreting results is tricky

Distance metrics:

- Euclidean and other Minkowski metrics

- geospatial distances

- metrics for non continuous data

Partitioning methods: inexpensive, typically non deterministic

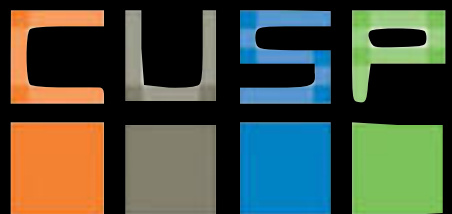
- Hard methods: *K-means, K-medoids*

- Soft (or fuzzy) methods: (i.e. probabilistic approach)

 - Expectation Maximization Mixture models*

Hierarchical methods:

- divisive vs agglomerative, dendrograms



RESOURCES:

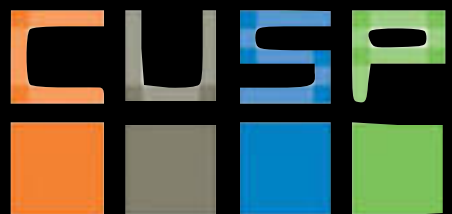
a comprehensive review of clustering methods

Data Clustering: A Review, Jain, Murty, Flynn 1999

<https://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf>

a blog post on how to generate and interpret a scipy dendrogram by Jörn Hees

<https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/>



READING:

your data aint that big...

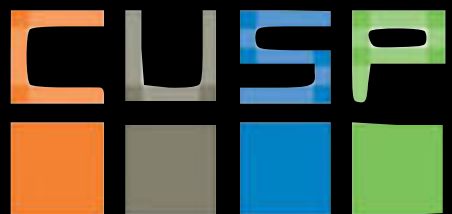
https://www.chrisstucchio.com/blog/2013/hadoop_hatred.html

HW: cluster NYC business history

1. cluster the economic trends in NYC using 2 methods: use K-Means and another method of your choice (e.g. DBscan, agglomerative clustering): use the time behavior of the number of establishments per zip code as your feature space
2. see if the clusters based on the time behavior also form spatial clusters.

map the time-based clusters

(e.g. with geopanda as a heat map).
attempt an interpretation.



HW: cluster NYC business history

Use census data for NYC businesses:

number of establishments per zip code for ~20 years since 1994

you can get the zip code info (list of NYC zip codes and shape files for plotting) here:

<http://data.nycprepared.org/dataset/nyc-zip-code-tabulation-areas/resource/0c0e14e9-78e1-404e-97b0-c2fabceb3981>

this is the link to the census business data

<http://www.census.gov/econ/cbp/download/>

you can download manually, which is labour intensive, or on the terminal via ftp, which requires some wrangling, but I did that for you! (see below)

```
for ((y=93; y<=99; y+=1)); do wget zbp$y\totals.zip; done
for ((y=0; y<=9; y+=1)); do wget ftp://ftp.census.gov/econ200$yV
CBP_CSV/zbp0$y\totals.zip; done
for ((y=10; y<=15; y+=1)); do wget ftp://ftp.census.gov/econ20$yV
CBP_CSV/zbp$y\totals.zip; done
```

