Clustering in astronomy

dr. federica bianco fb55@nyu.edu

 @fedhere 

Recap:

- Good scientific practices: Falsifiability, Reproducibility
- Basic statistics: distributions and their moments
- Hypothesis testing: *p*-value, statistical significance
- Model fitting: Regression, OLS, Optimization
- Goodness of fit tests, Likelihood
- Bayesian concepts

machine learning

algorithms that can learn from and make predictions on data.

machine learning

algorithms that can learn from and make predictions on data.

machine learning

machine learning

algorithms that can learn from and make predictions on data.

machine learning

clustering

machine learning

algorithms that can learn from and make predictions on data.

machine learning

clustering

distances

machine learning

algorithms that can learn from and make predictions on data.

machine learning

clustering

distances

k-means

machine learning

algorithms that can learn from and make predictions on data.

machine learning

clustering

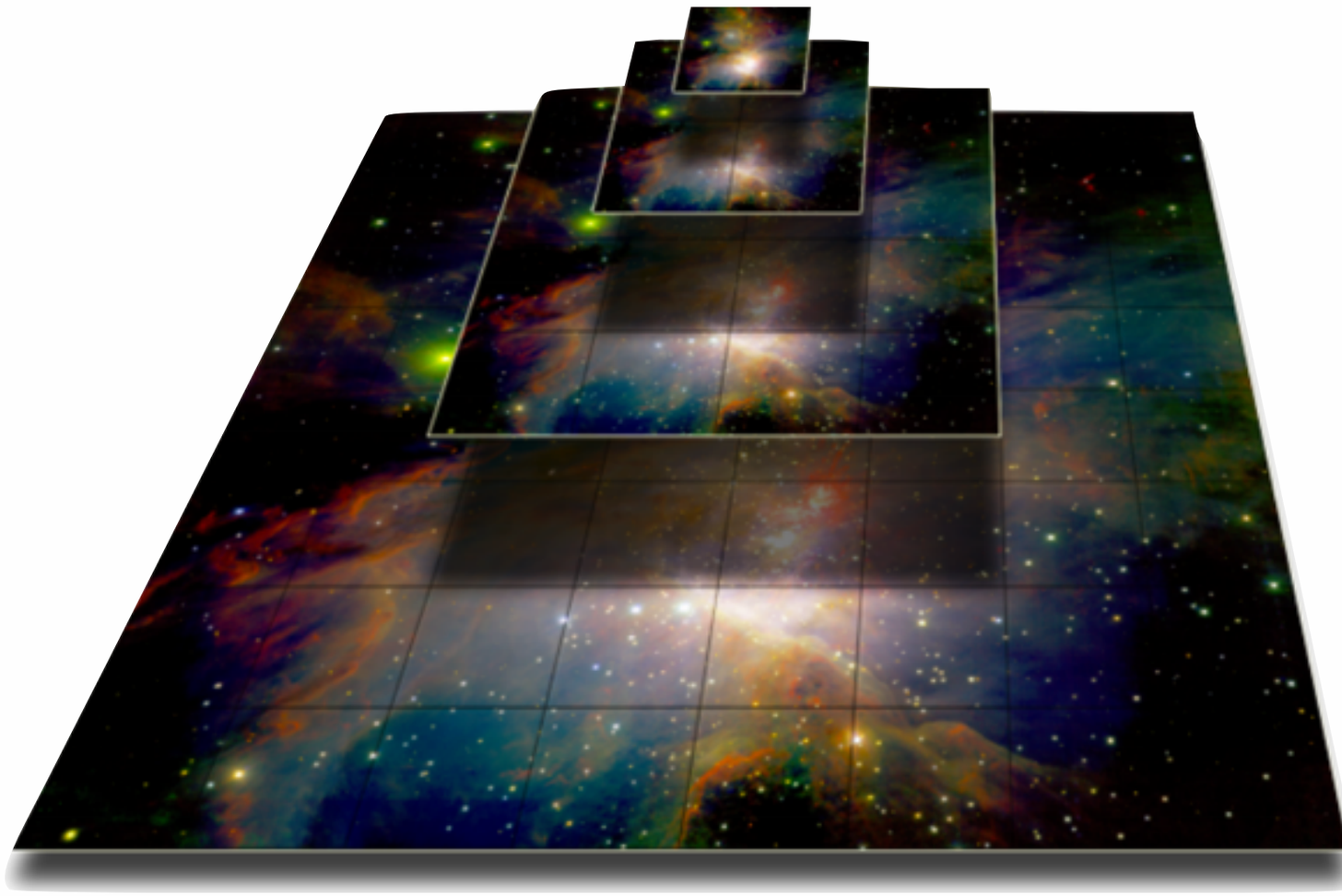
distances

k-means

probabilistic clustering

hierarchical

first CCD in astronomy: 1975 100x100 pixel x 16bits= 0.02MB



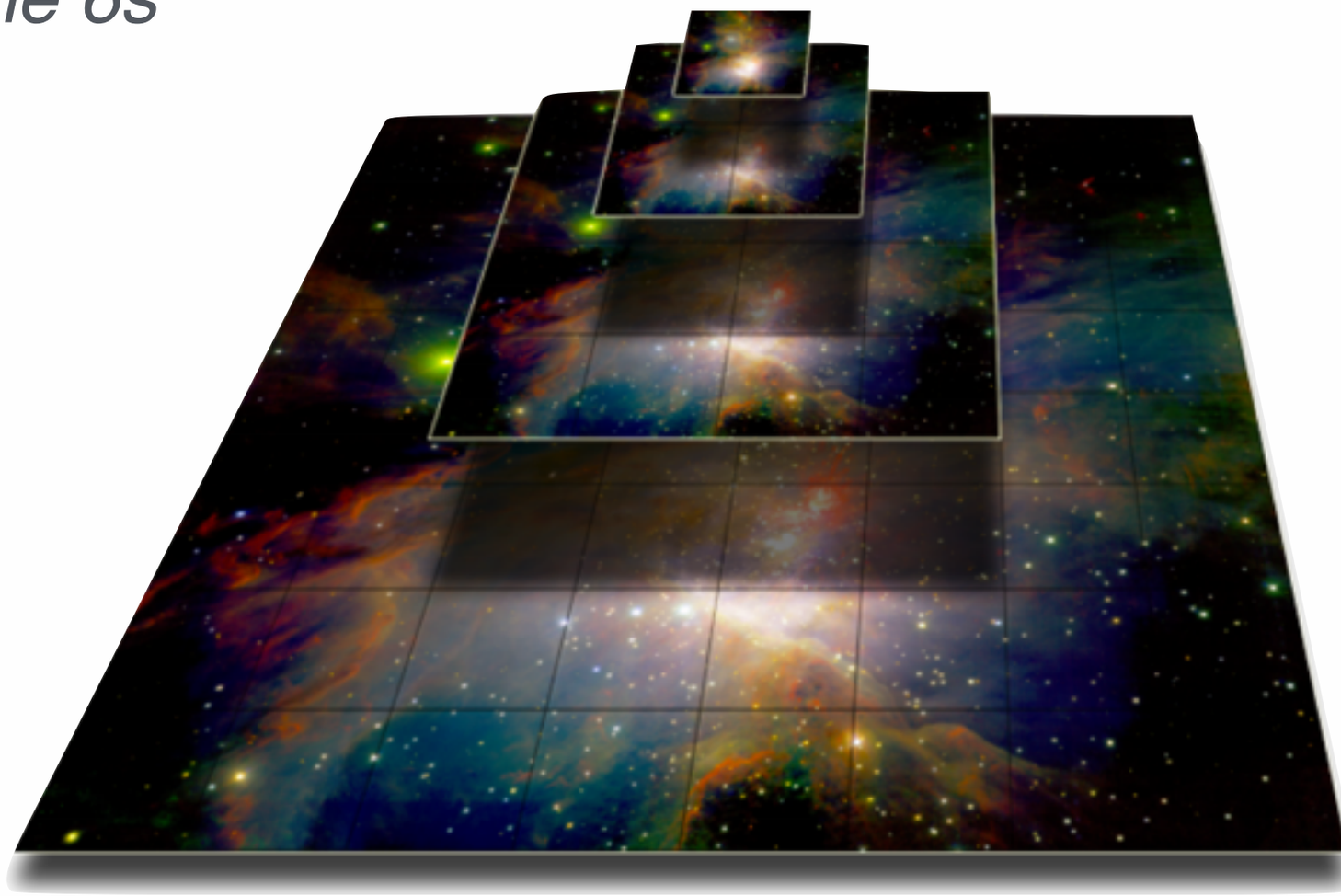
log10(data)

first CCD in astronomy: 1975 100x100 pixel x 16bits= 0.02MB

modern CCD (’90 KeplerCam FLWO) 2k x 2k pix x 8bits = 4MB

traditional telescope throughput: 0.5GB/night

A year of data fits in 1 iPhone 6s



log10(data)

first CCD in astronomy: 1975 100x100 pixel x 16bits= 0.02MB

modern CCD (’90 KeplerCam FLWO) 2k x 2k pix x 8bits = 4MB
traditional telescope throughput: 0.5GB/night

first massive astronomical survey:
MACHO: 7TB over 5y (2000)

a bit more than 2 Apple Time Capsules

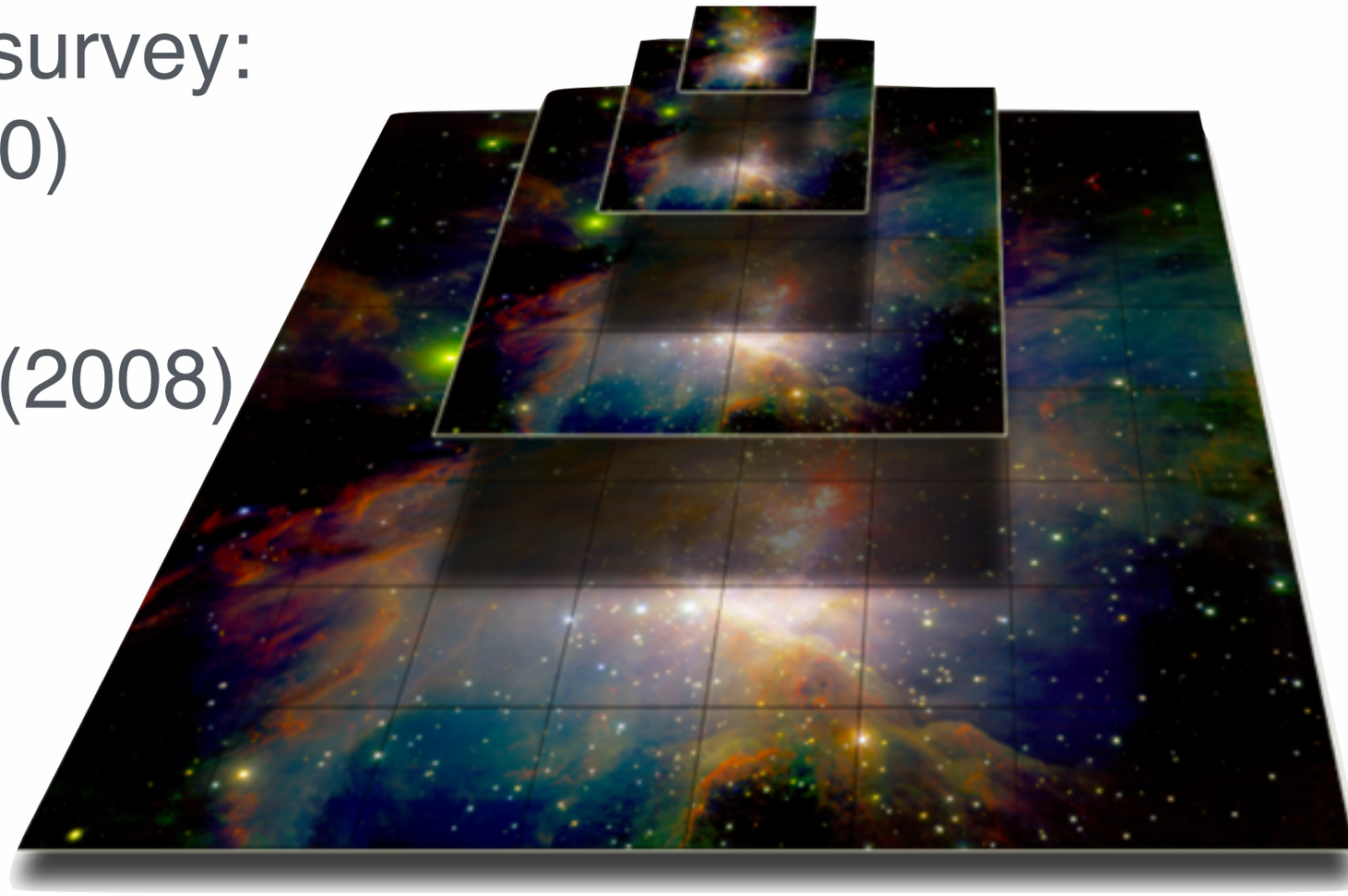
log10(data)

first CCD in astronomy: 1975 100x100 pixel x 16bits= 0.02MB

modern CCD (’90 KeplerCam FLWO) 2k x 2k pix x 8bits = 4MB
traditional telescope throughput: 0.5GB/night

first massive astronomical survey:
MACHO: 7TB over 5y (2000)

SDSS imaging data: 25TB (2008)
sources: 1,231,051,050



log10(data)

first CCD in astronomy: 1975 100x100 pixel x 16bits= 0.02MB

modern CCD (’90 KeplerCam FLWO) 2k x 2k pix x 8bits = 4MB
traditional telescope throughput: 0.5GB/night

first massive astronomical survey:
MACHO: 7TB over 5y (2000)

SDSS imaging data: 25TB (2008)
sources: 1,231,051,050

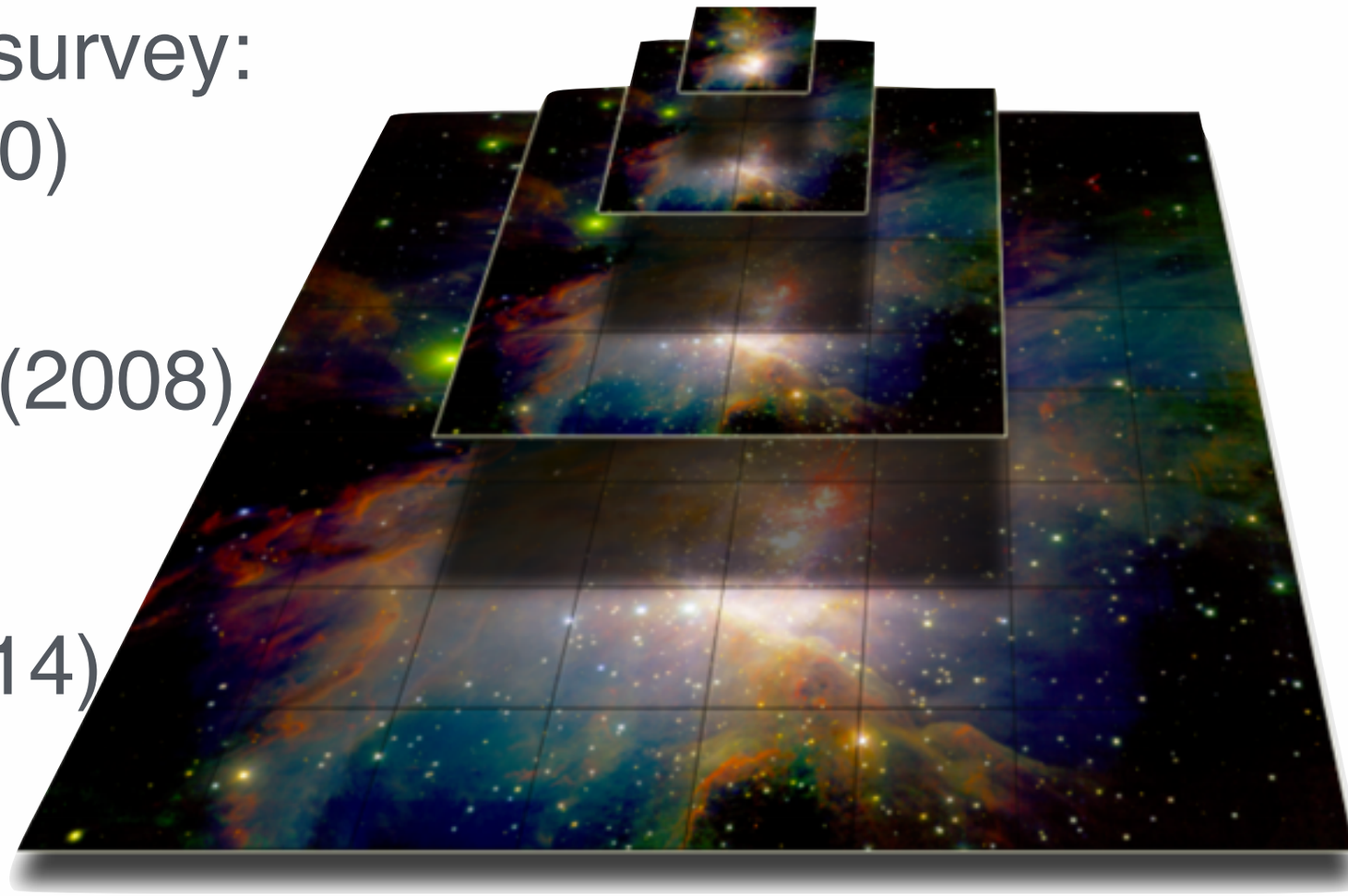
PanSTARRS img: 2PB (2014)
sources: 1.10×10^9

log10(data)

first CCD in astronomy: 1975 100x100 pixel x 16bits= 0.02MB

modern CCD (’90 KeplerCam FLWO) 2k x 2k pix x 8bits = 4MB
traditional telescope throughput: 0.5GB/night

first massive astronomical survey:
MACHO: 7TB over 5y (2000)



SDSS imaging data: 25TB (2008)
sources: 1,231,051,050

PanSTARRS img: 2PB (2014)
sources: 1.10×10^9

LSST: 200PB (2030, 15TB/night)

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*

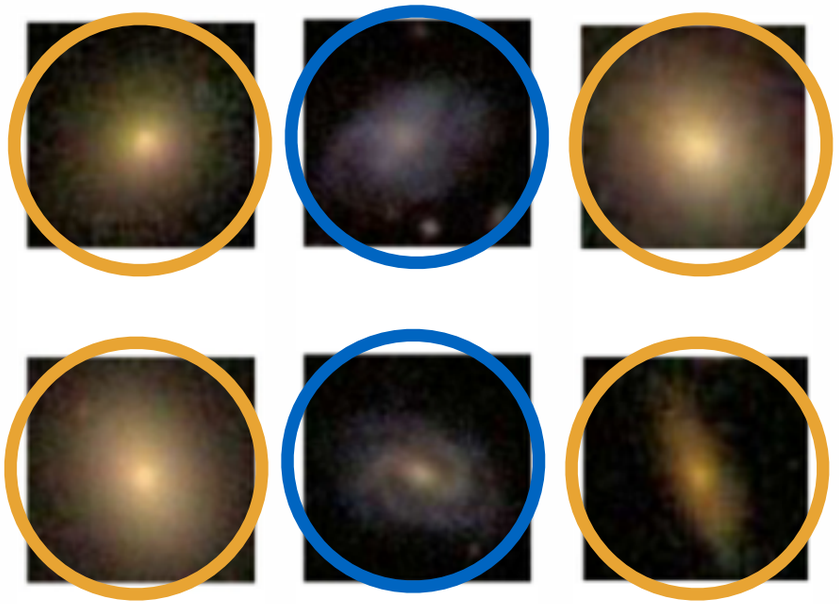
machine learning

algorithms that can learn from and make predictions on data.

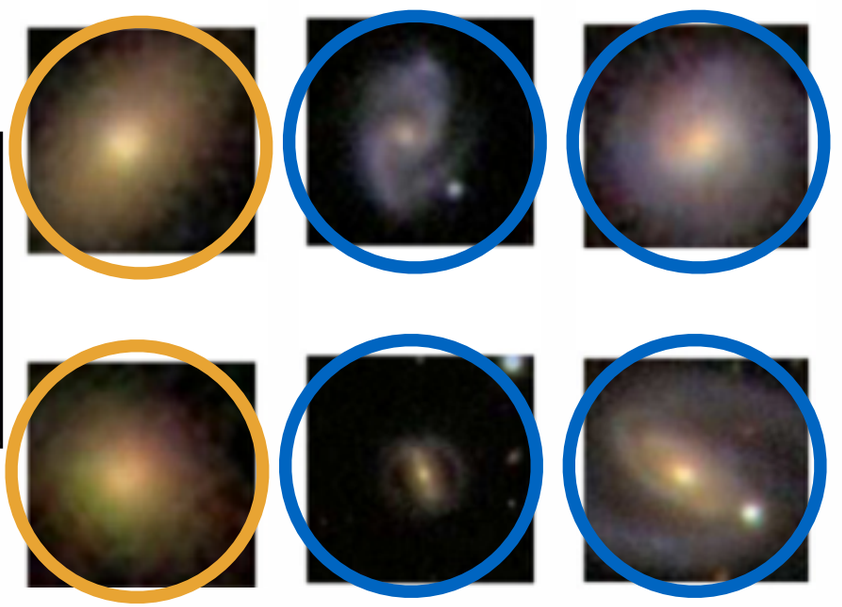
supervised learning

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

Elliptical



Spiral



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 structure in data

machine learning

supervised methods

classification

prediction

unsupervised methods

understanding structure

organizing + compressing data

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

Astronomical Catalogs

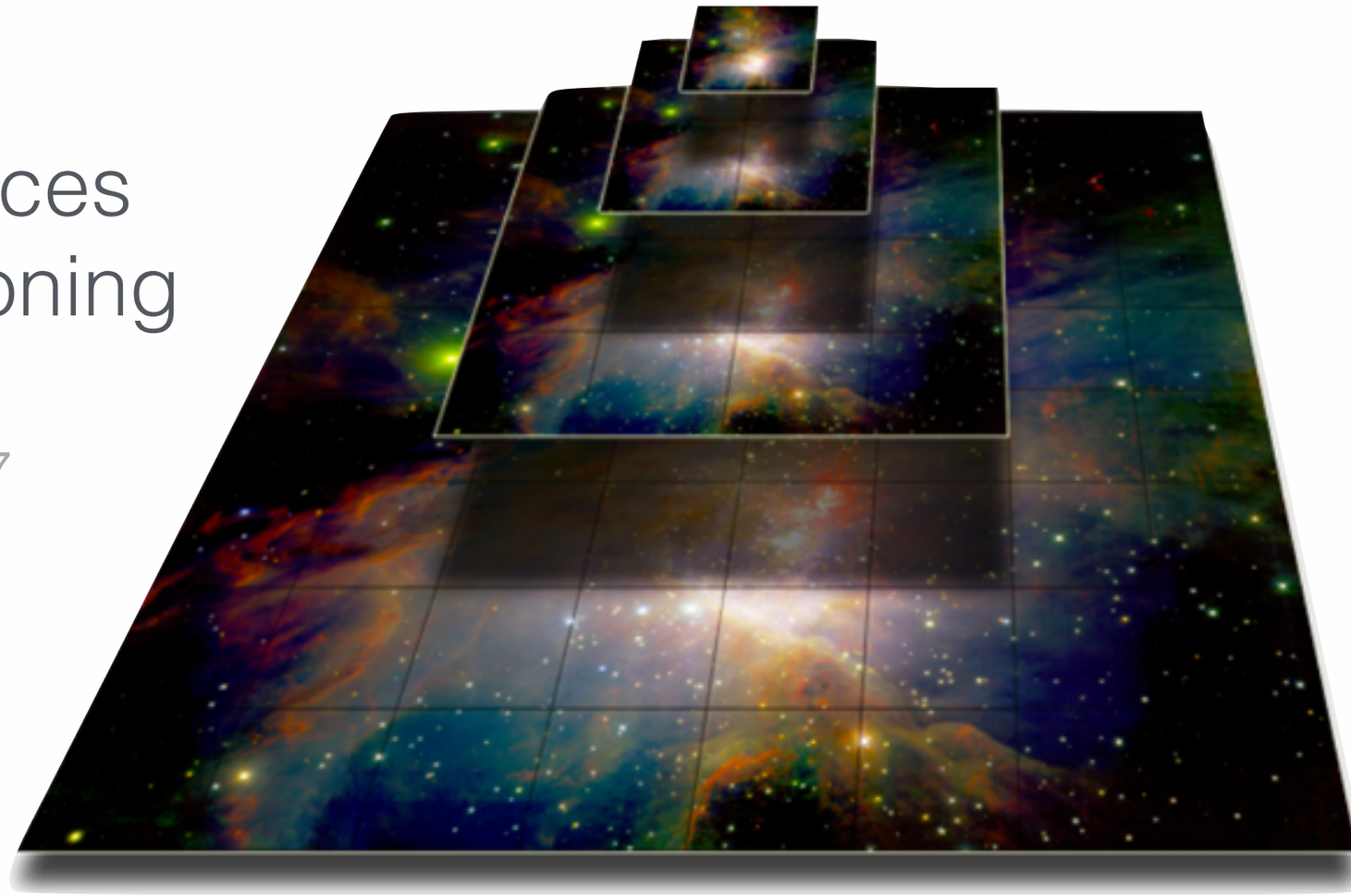
Organizing billions of sources

e.g. ***SkyCat***:

10^9 photometric sources organized by partitioning a 7D space

Albrecht et al., 1997

galaxies,
stars,
QSO
…



Bertin+14 http://arxiv.org/pdf/1403.6025.pdf

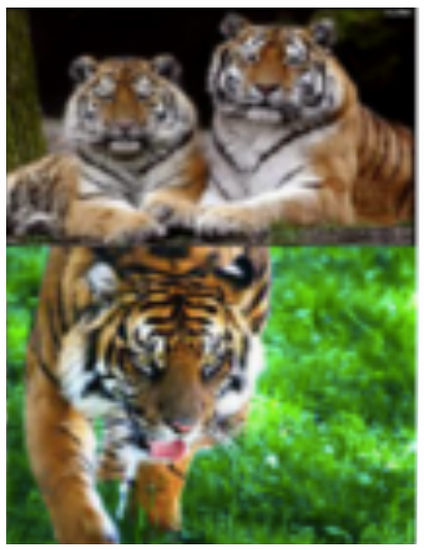
what is a cluster?

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

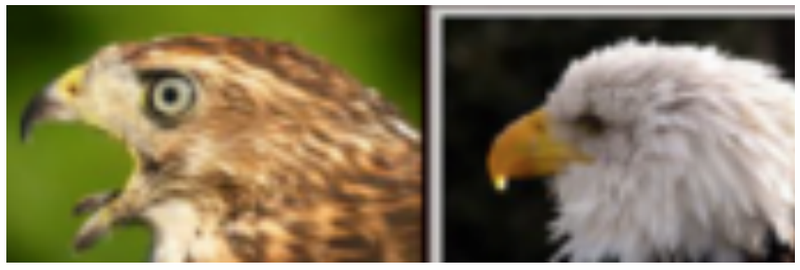
what is a cluster?

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

tigers



whales

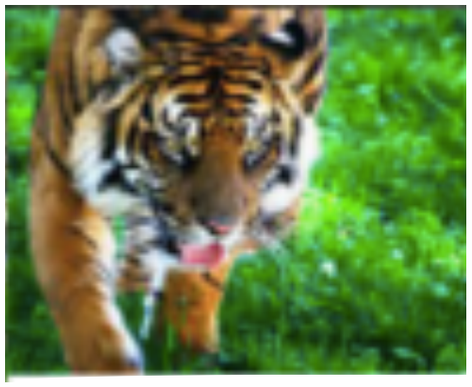


raptors

what is a cluster?

- **internal criterion:** members of the cluster should be similar to each other

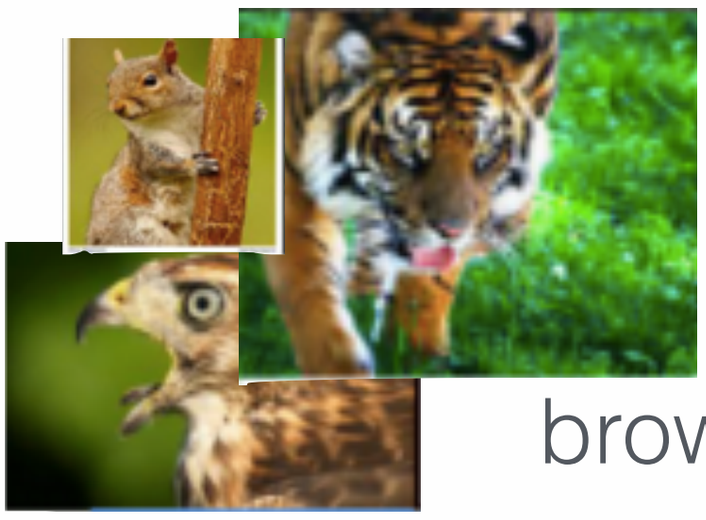
- **external criterion:** objects outside the cluster should be dissimilar from the objects inside the cluster **(intra-cluster distance)**


what is a cluster?

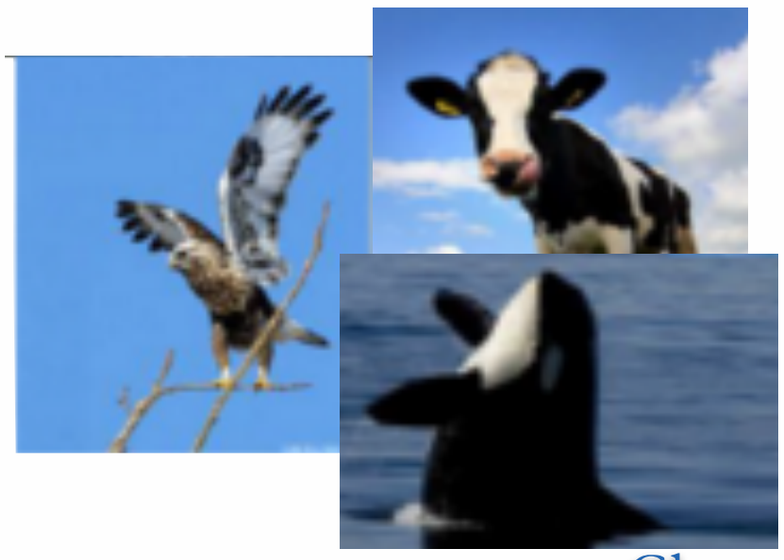
- **internal criterion:** members of the cluster should be similar to each other

- **external criterion:** objects outside the cluster should be dissimilar from the objects inside the cluster



green
brown & white

blue
black
&
white

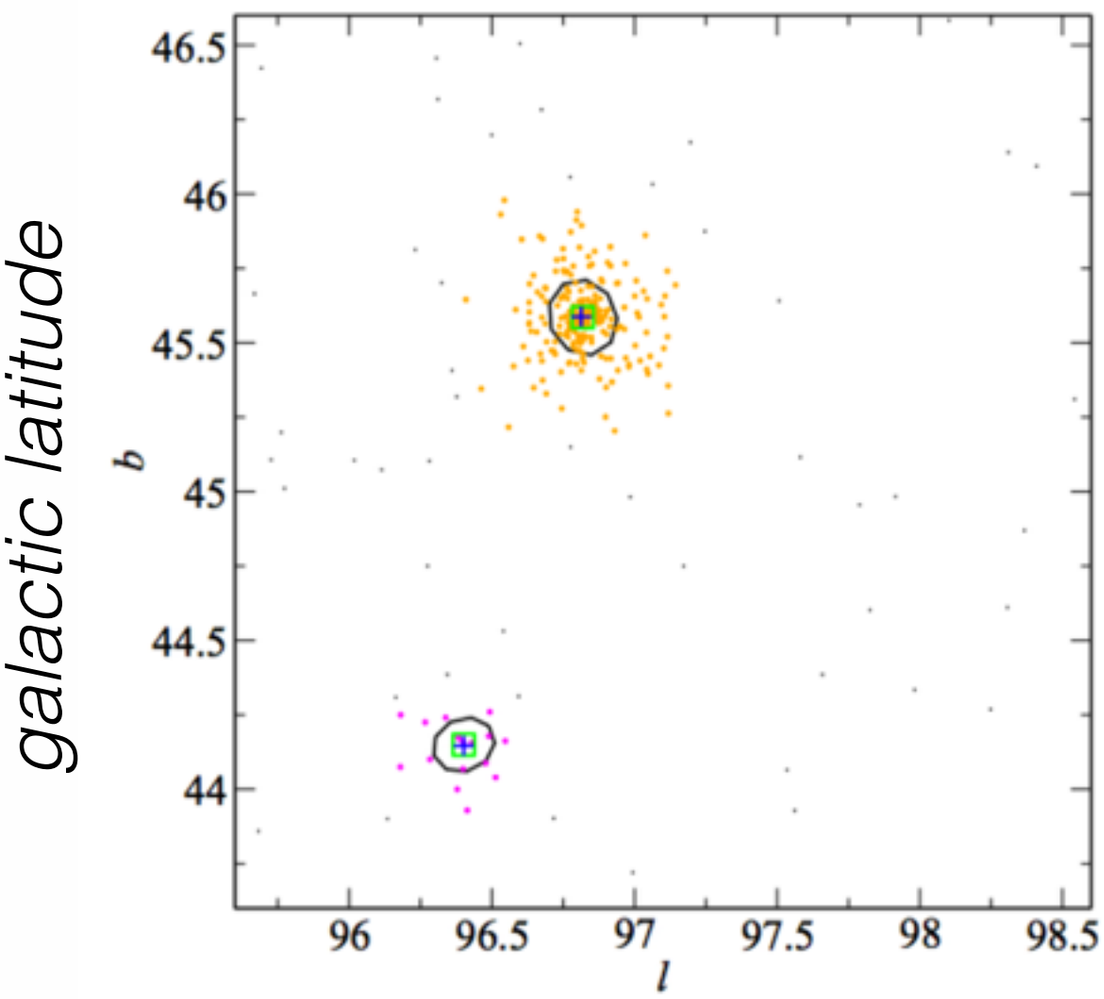


Defining the feature space

Spatial clustering

γ-ray DBSCAN: a clustering algorithm applied to *Fermi*-LAT γ-ray data

Context. The density based spatial clustering of applications with noise (DBSCAN) is a topometric algorithm used to cluster spatial data that are affected by background noise. For the first time, we propose this method to detect sources in γ-ray astrophysical images obtained from the *Fermi*-LAT data, where each point corresponds to the arrival direction of a photon.



2013A&A...549A.138T

Discovery of structure: How many classes?

THREE TYPES OF GAMMA-RAY BURSTS

SOMA MUKHERJEE,[1,2,3] ERIC D. FEIGELSON,[4] GUTTI JOGESH BABU,[5] FIONN MURTAGH,[6,7]
CHRIS FRALEY,[8] AND ADRIAN RAFTERY[8]

Received 1998 February 9; accepted 1998 June 25

A multivariate analysis of gamma-ray burst (GRB) bulk properties is presented to discriminate between distinct classes of GRBs. Several variables representing burst duration, fluence, and spectral

decline rate (2 parameters),
total emitted flux (at peak)
hardness of the spectrum (2 parameters)

$$logT_{50}, logT_{90},$$

$$F_{tot} = F_1 + F_2 + F_3 + F_4,$$

$$H_{32} = F_3/F_2, \quad H_{321} = F_3/(F_2 + F_1)$$

Automatic classification

AUTOMATED UNSUPERVISED CLASSIFICATION OF THE SLOAN DIGITAL SKY SURVEY STELLAR SPECTRA USING *k*-MEANS CLUSTERING

J. SÁNCHEZ ALMEIDA[1,2] AND C. ALLENDE PRIETO[1,2]
[1] Instituto de Astrofísica de Canarias, E-38205 La Laguna, Tenerife, Spain
[2] Departamento de Astrofísica, Universidad de La Laguna, Tenerife, Spain; jos@iac.es, callende@iac.es
Received 2012 September 10; accepted 2012 November 23; published 2013 January 8

Large spectroscopic surveys require automated methods of analysis. This paper explores the use of *k*-means clustering as a tool for automated unsupervised classification of massive stellar spectral catalogs. The classification

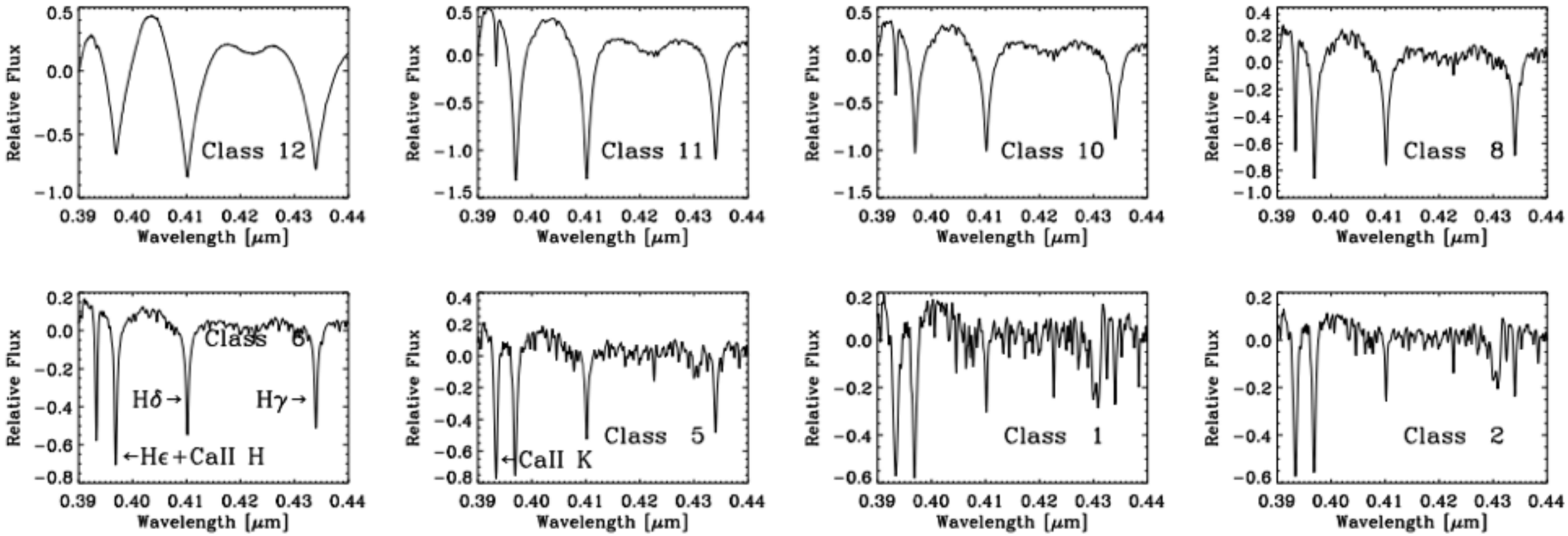
emission intensity
at designated wavelengths
(e.g. *H-alpha, Helium, Calcium absorption*)

$$f_{H\alpha}, f_{He}, f_{CaII}$$

Automatic classification

AUTOMATED UNSUPERVISED CLASSIFICATION OF THE SLOAN DIGITAL SKY SURVEY STELLAR SPECTRA USING *k*-MEANS CLUSTERING

J. SÁNCHEZ ALMEIDA[1,2] AND C. ALLENDE PRIETO[1,2]
[1] Instituto de Astrofísica de Canarias, E-38205 La Laguna, Tenerife, Spain
[2] Departamento de Astrofísica, Universidad de La Laguna, Tenerife, Spain; jos@iac.es, callende@iac.es
Received 2012 September 10; accepted 2012 November 23; published 2013 January 8



$$f_{H\alpha}, f_{He}, f_{CaII}$$

Defining the distance or similarity

Distance Metrics

Continuous variables

Numerical attributes

Minkowski family of distances

N features (dimensions)

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

Distance Metrics

Continuous variables

Numerical attributes

Minkowski family of distances

N features (dimensions)

$$D(i,j) = \sqrt[p]{\sum_{k=0}^{N} |x_{ik} - x_{jk}|^p}$$

$$D(i,j) \geq 0$$

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

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

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

Distance Metrics

Continuous variables

Numerical attributes

Minkowski family of distances

N features (dimensions)

$$D(i,j) = \sqrt[p]{\sum_{k=0}^{N} |x_{ik} - x_{jk}|^p}$$

Manhattan: p = 1

Distance Metrics

Continuous variables

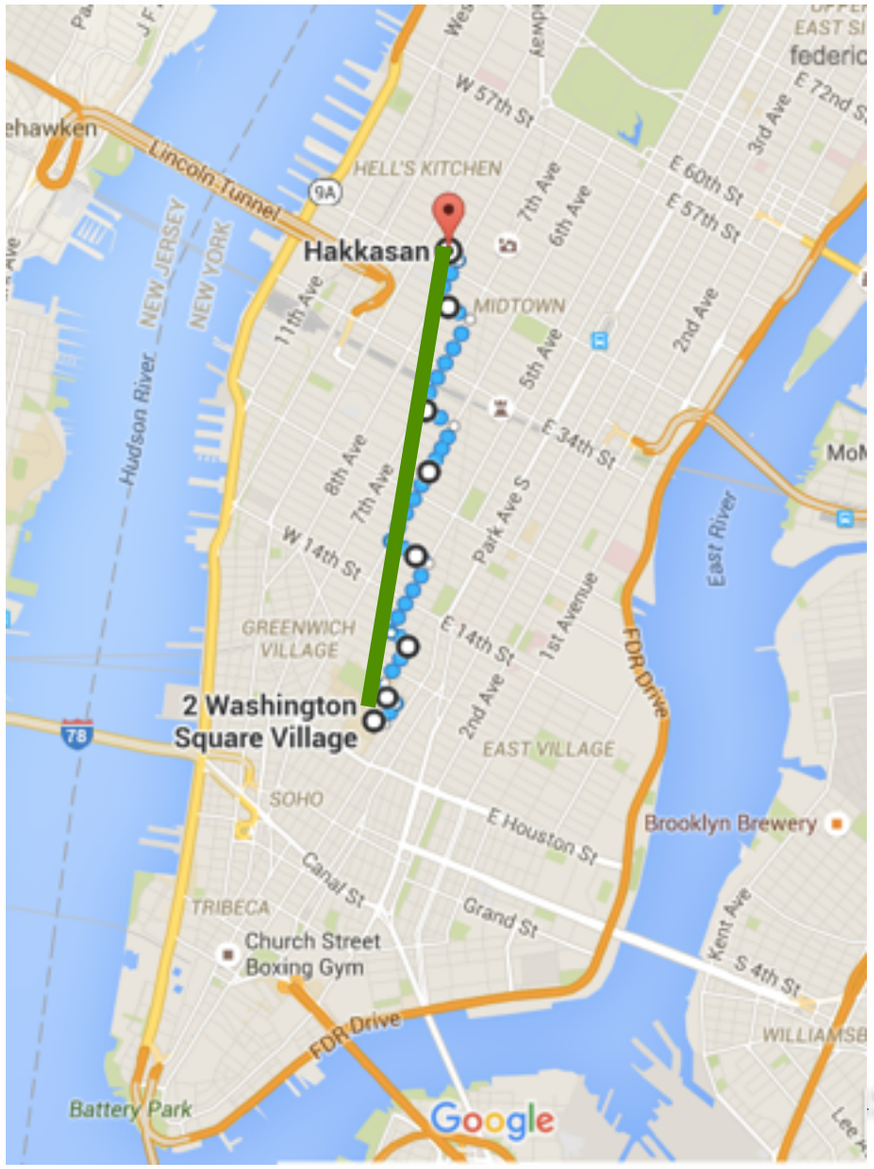
Numerical attributes

Minkowski family of distances

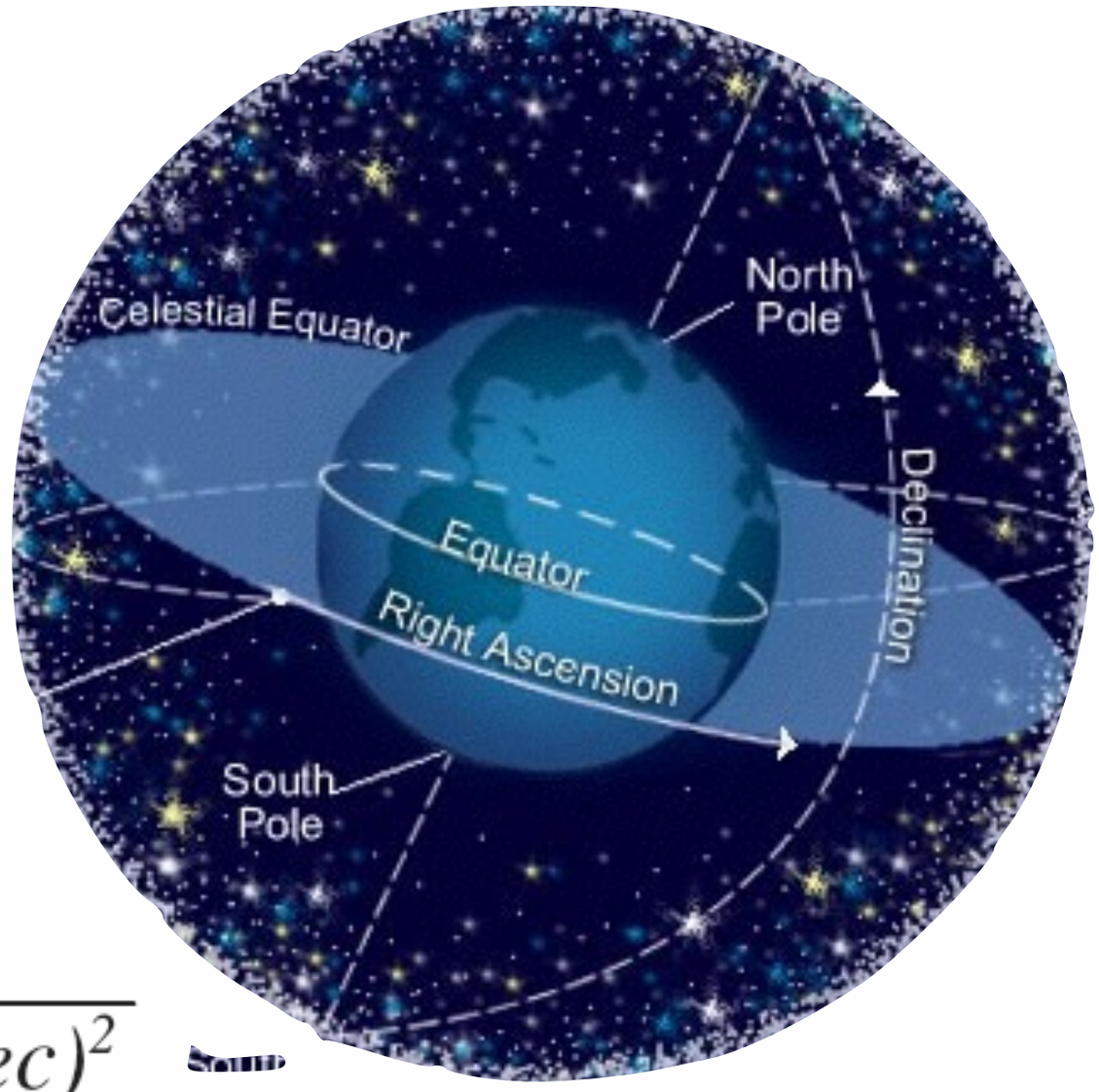
N features (dimensions)

$$D(i,j) = \sqrt[p]{\sum_{k=0}^{N} |x_{ik} - x_{jk}|^p}$$

Euclidean: p = 2



Distance Metrics Continuous variables



Great Circle distances:

$$\theta(ij) = \sqrt{(\Delta RA \cos(Dec))^2 + (\Delta Dec)^2}$$

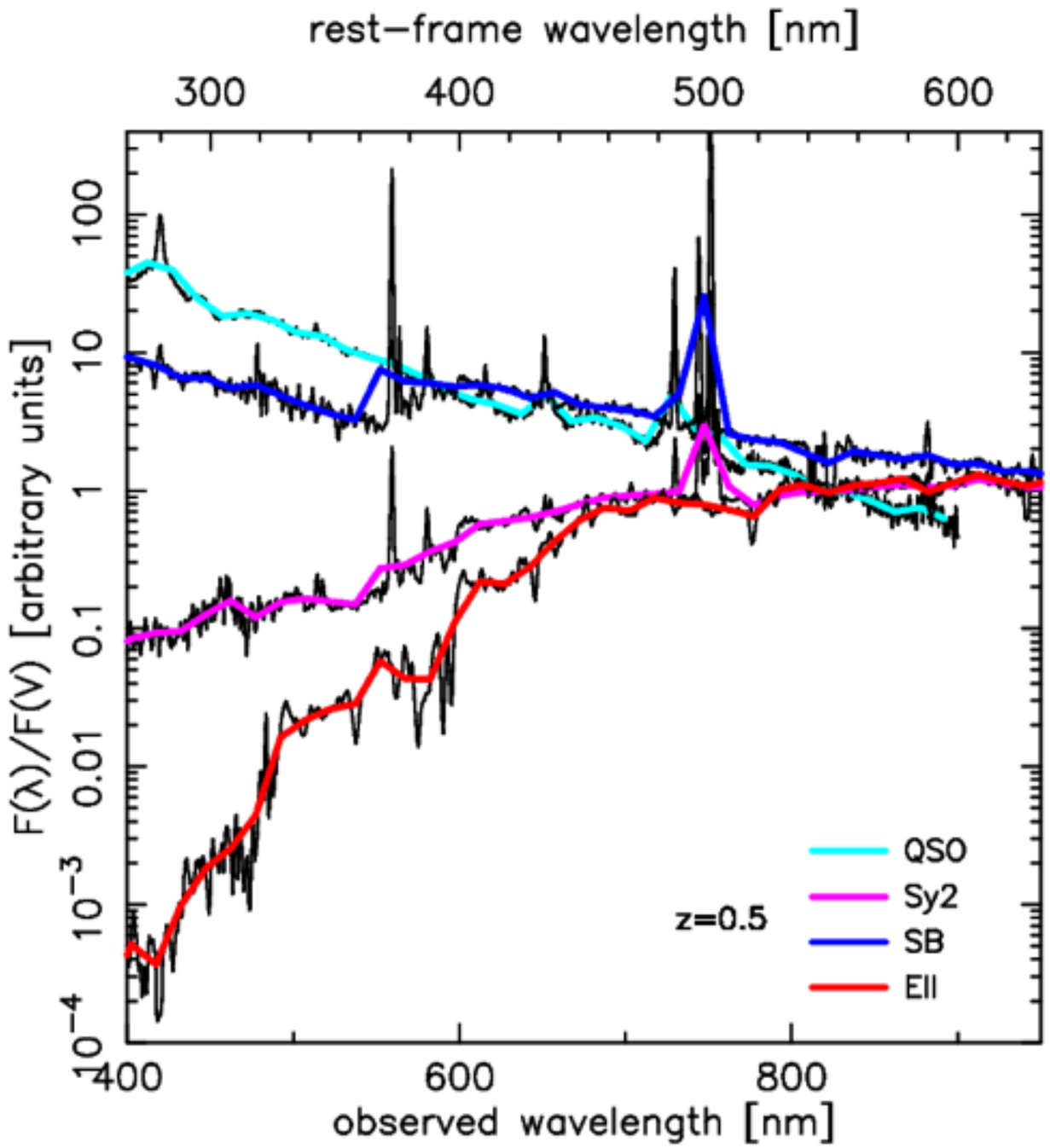
latitude and longitude ***δ, λ***

$$\theta(ij) = \cos^{-1}(\sin\delta_i \cdot \sin\delta_j + \cos\delta_i \cdot \cos\delta_j \cdot \cos(\lambda_i - \lambda_j))$$

Distance Metrics High dimensional vectors

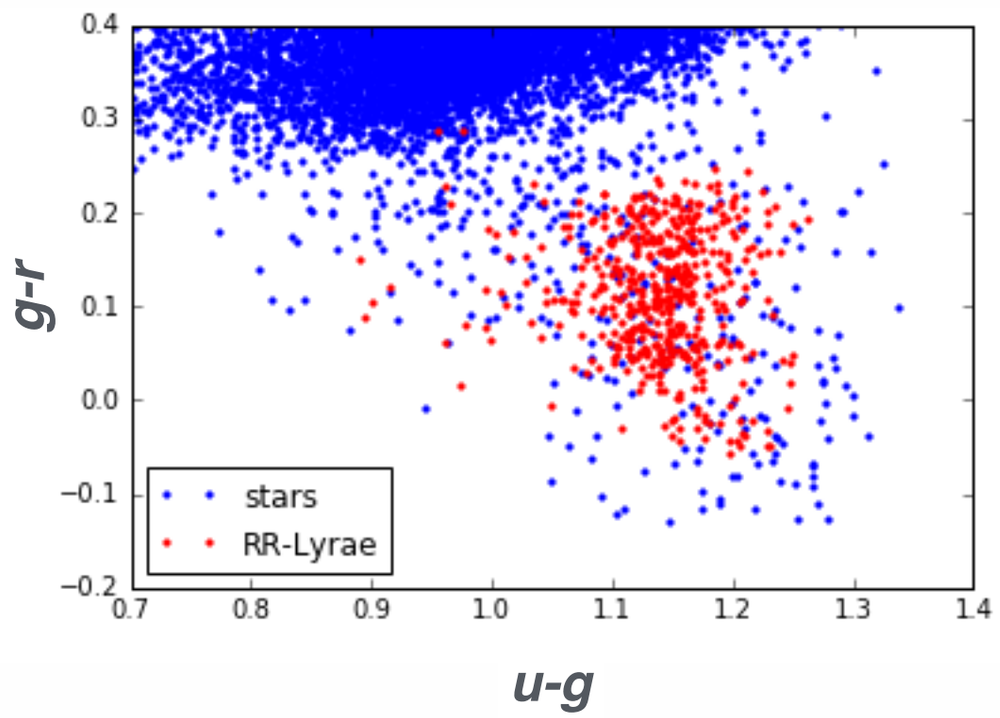
Pearson's correlation

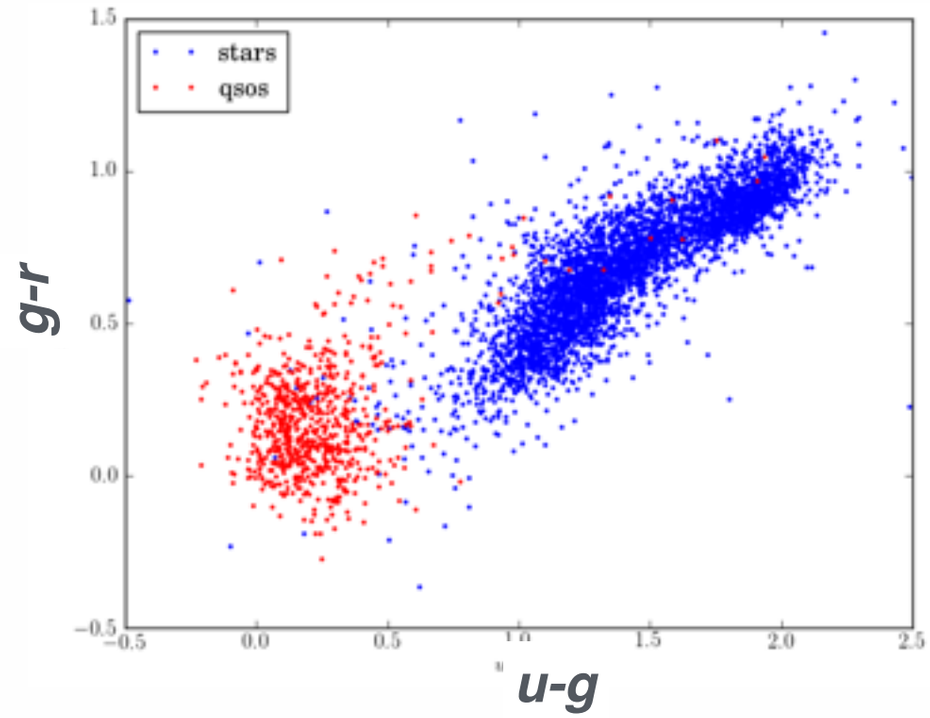
$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$



Distance Metrics Continuous variables

Euclidean distances in color space

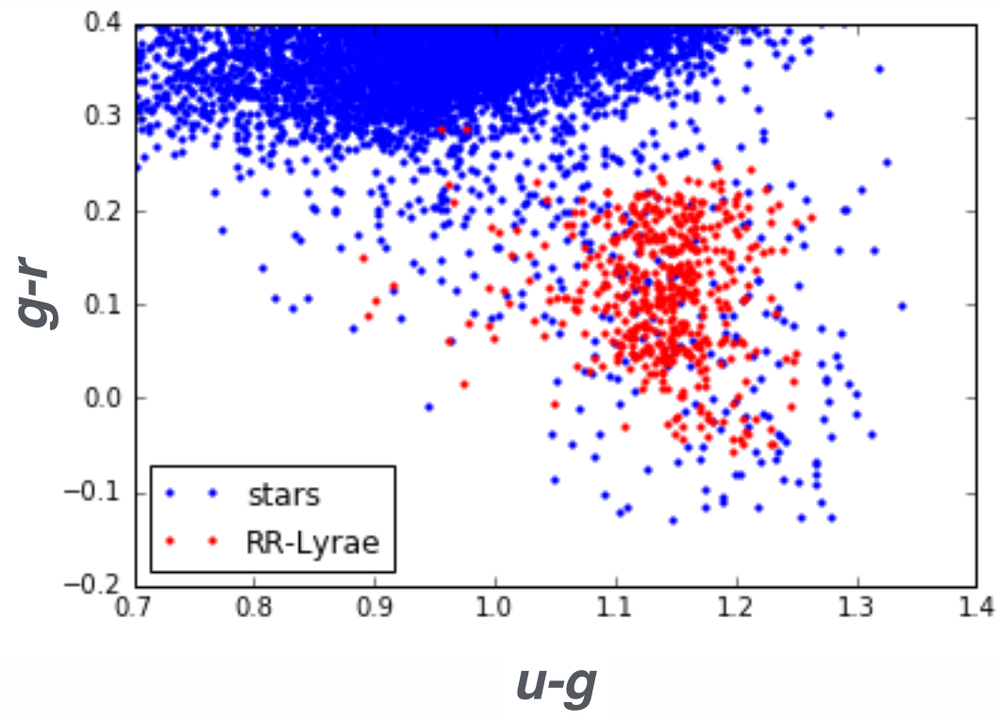


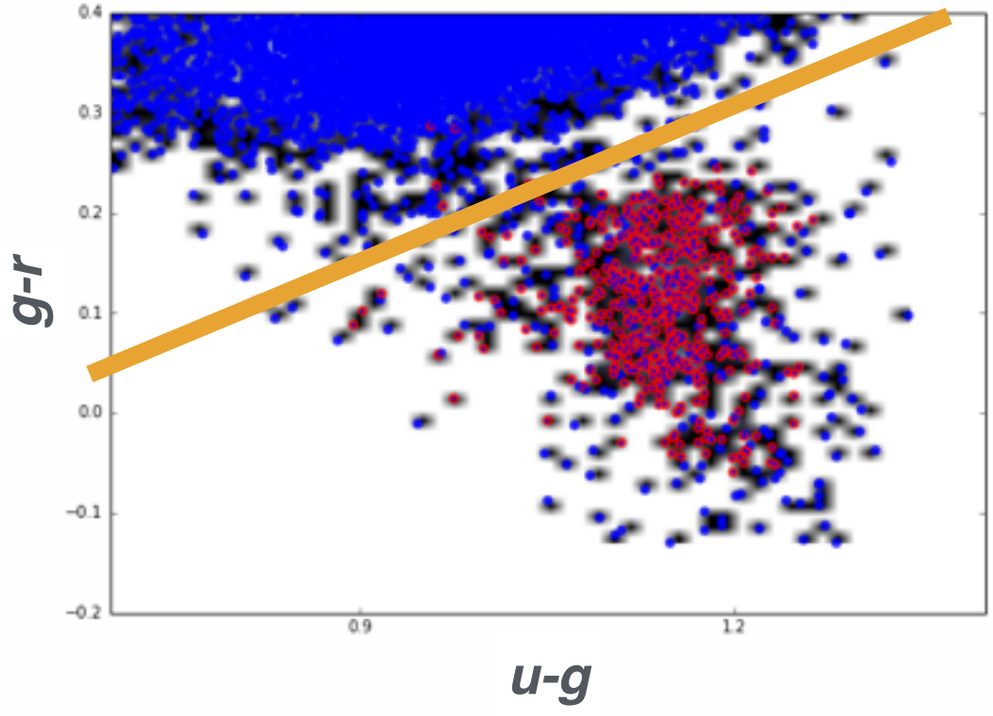


jupyter

Distance Metrics Continuous variables

Euclidean distances in color space

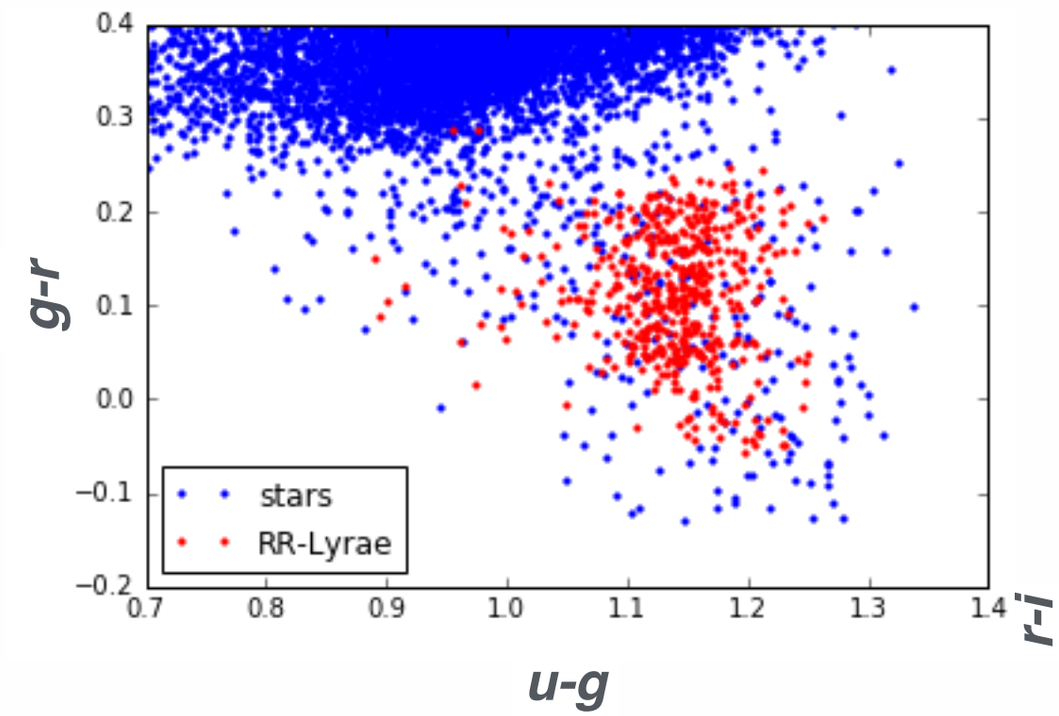




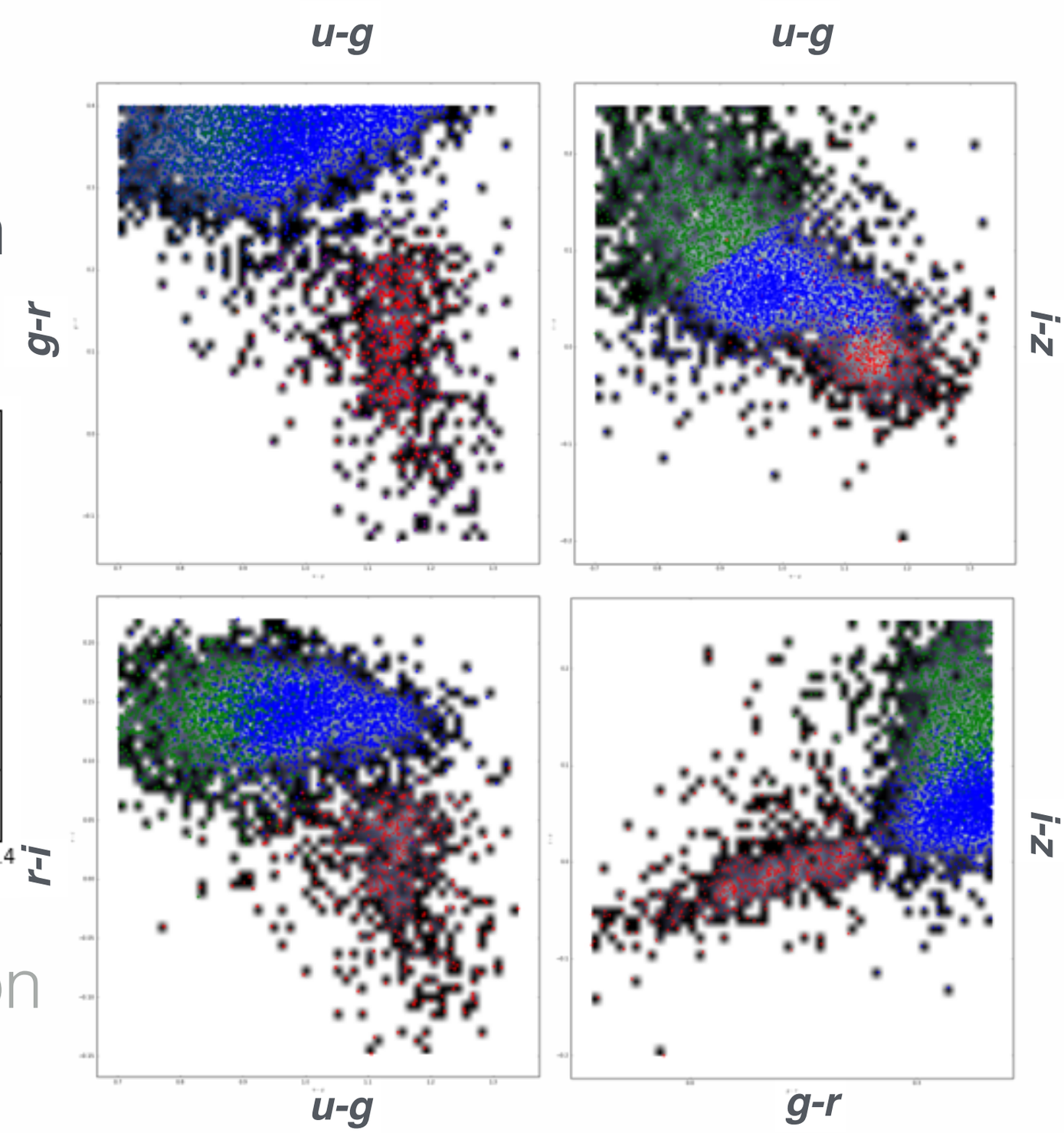
jupyter

Distance Metrics

Euclidean distances in 4D color space



time-domain classification



http://www.astroml.org/examples/datasets/plot_sdss_galaxy_colors.html

Distance Metrics Binary variables (classes)

(e.g. the detection of a GRB in conjunction with a SN of a certain type)

	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

contingency table

‘simple’ $$D_S = \frac{M_{10}+M_{01}}{M_{00}+M_{10}+M_{01}+M_{11}}$$

Jaccard $$D_J = \frac{M_{10}+M_{01}}{M_{10}+M_{01}+M_{11}}$$

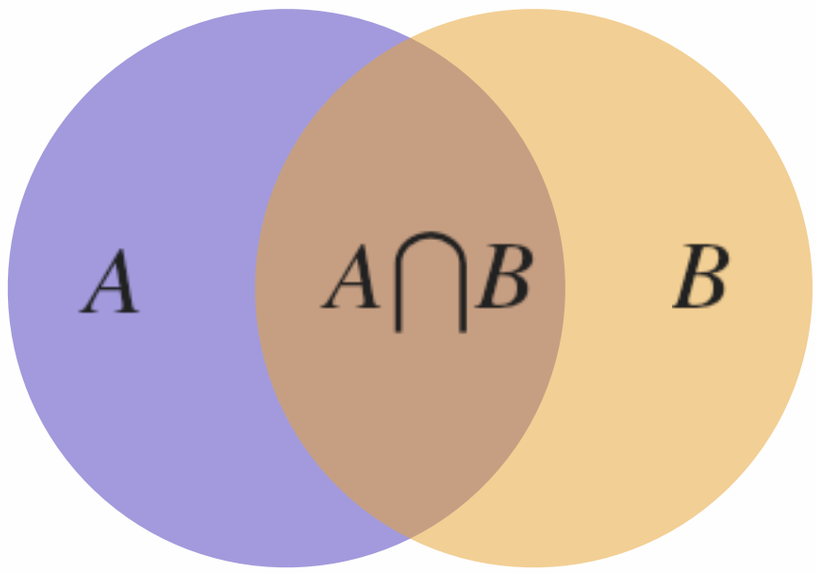
Distance Metrics Binary variables (classes)

	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

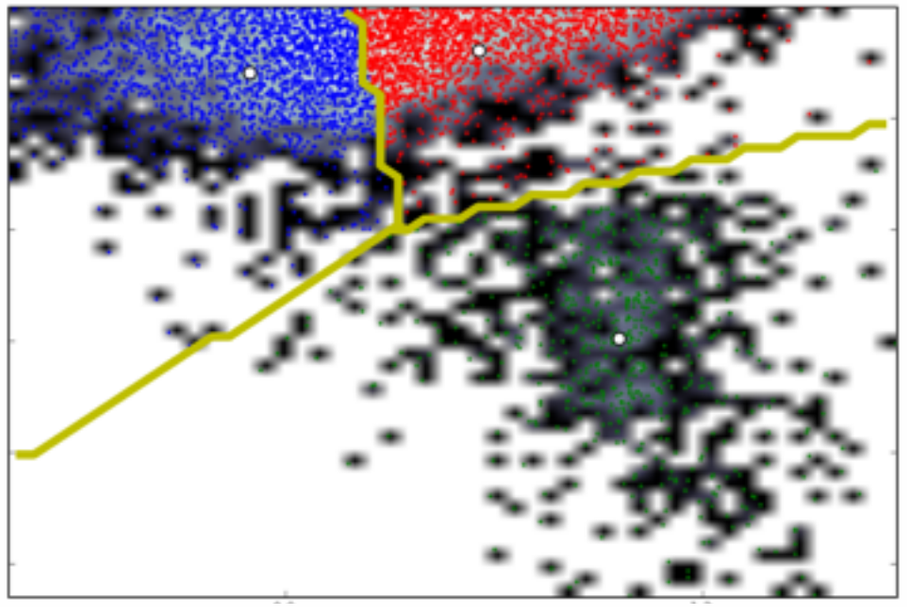
contingency table

‘simple’ $$D_S = \frac{M_{10}+M_{01}}{M_{00}+M_{10}+M_{01}+M_{11}}$$

Jaccard $$D_J = \frac{M_{10}+M_{01}}{M_{10}+M_{01}+M_{11}}$$



Clustering techniques



Partitioning

Hard clustering

K-means (McQueen ’67)

K-medoids (Kaufman & Rausseeuw ’87)

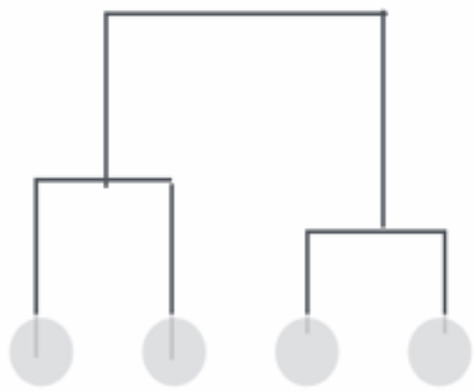
Soft Clustering

Expectation Maximization (Dempster,Laird,Rubin ’77)

Hierarchical

agglomerative

divisive



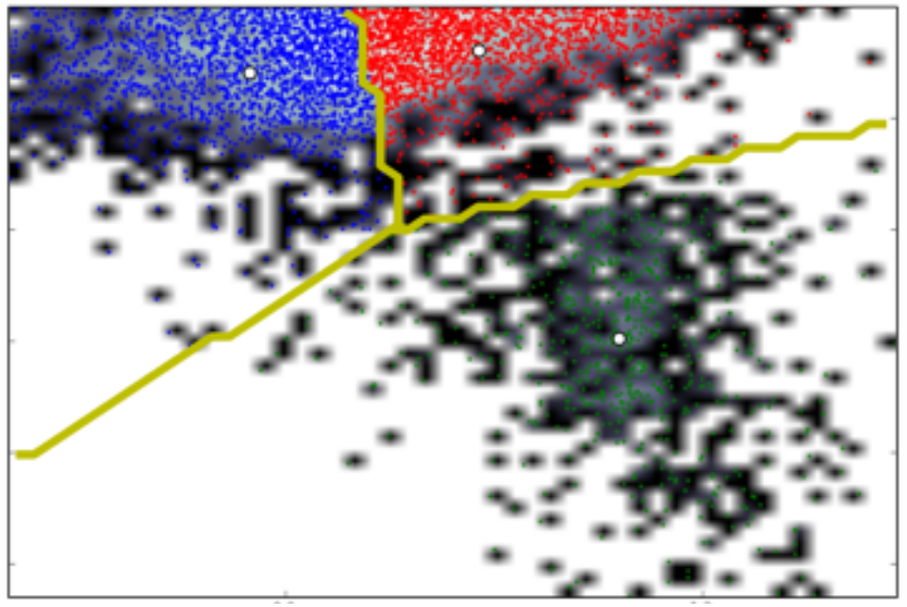
also: Density based, Grid based, Model based

DBSCAN, SPECTRAL

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,

DBSCAN, SPECTRAL


jupyter

K-means Algorithm

1. **Choose N "centers" guesses:** random points in the data space

2. **Calculate to which center each datapoint is closest and assign to the cluster with that center**

3. **Calculate the new centers as means of the assigned clusters**: these are the new N centers

4. **Iterate 2&3 until convergence**: when clusters no longer change

K-means Pros-Cons

Scalability:

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

K-means Pros-Cons

Scalability:

O(KdN): #clusters #dimensions #iterations #datapoints +

Minimizes the inter-cluster variance +

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

K-means Pros-Cons

Scalability:

O(KdN): #clusters #dimensions #iterations #datapoints +

Minimizes the inter-cluster variance +

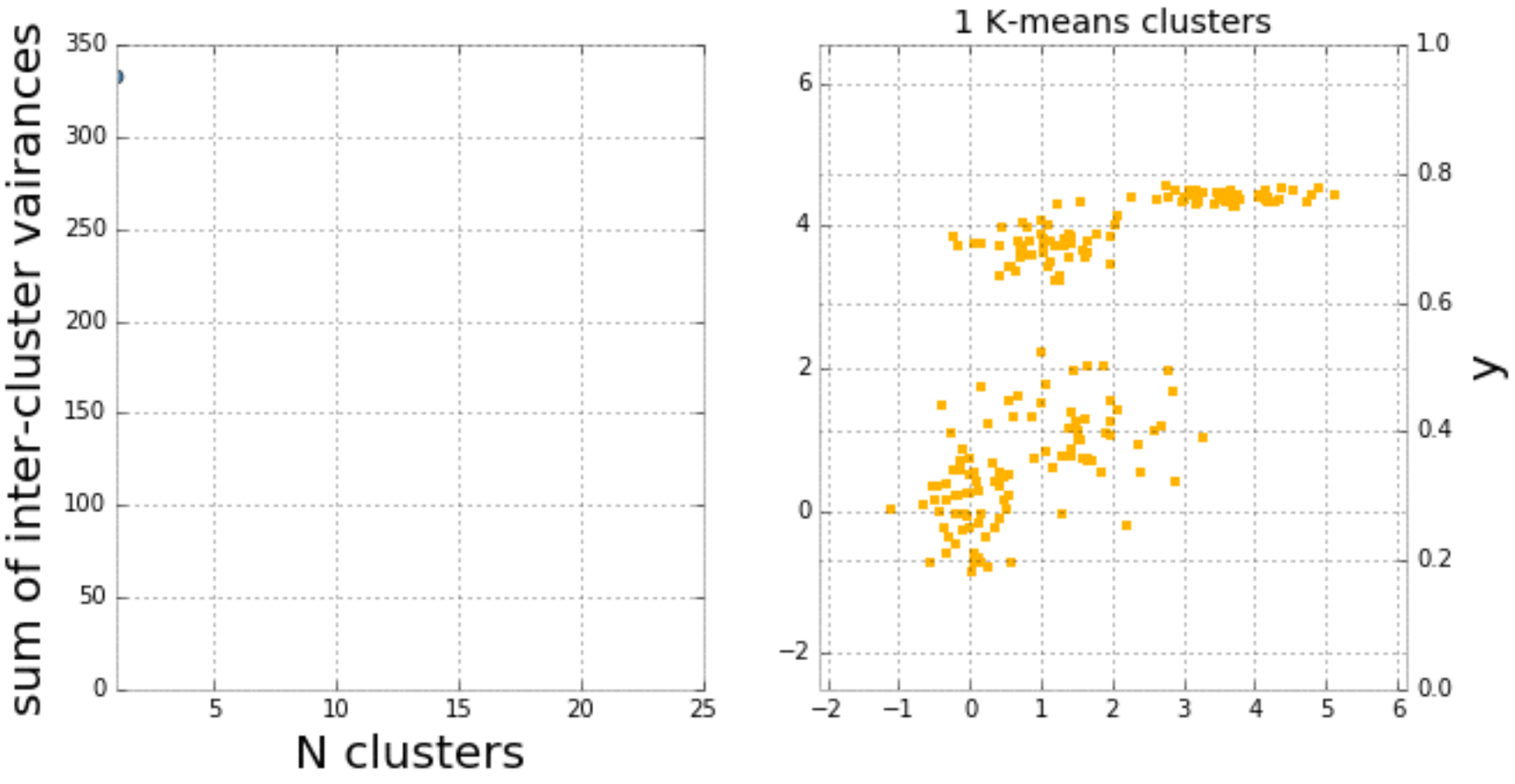
works on minimizing the aggregate distance within the cluster. *if the distance is Euclidean* this is the same as minimizing the variance

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 −

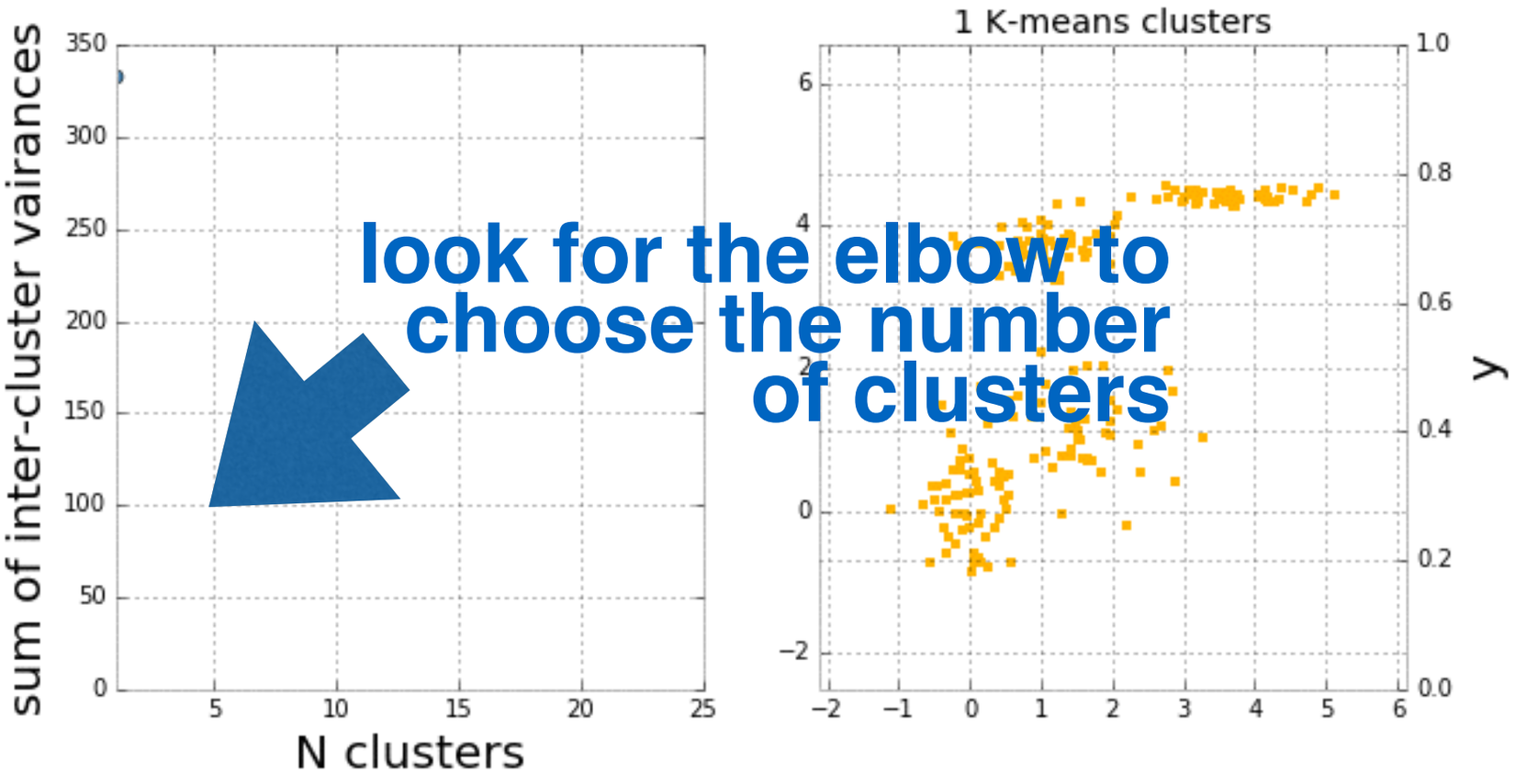
K-means

Must declare the number of clusters upfront −

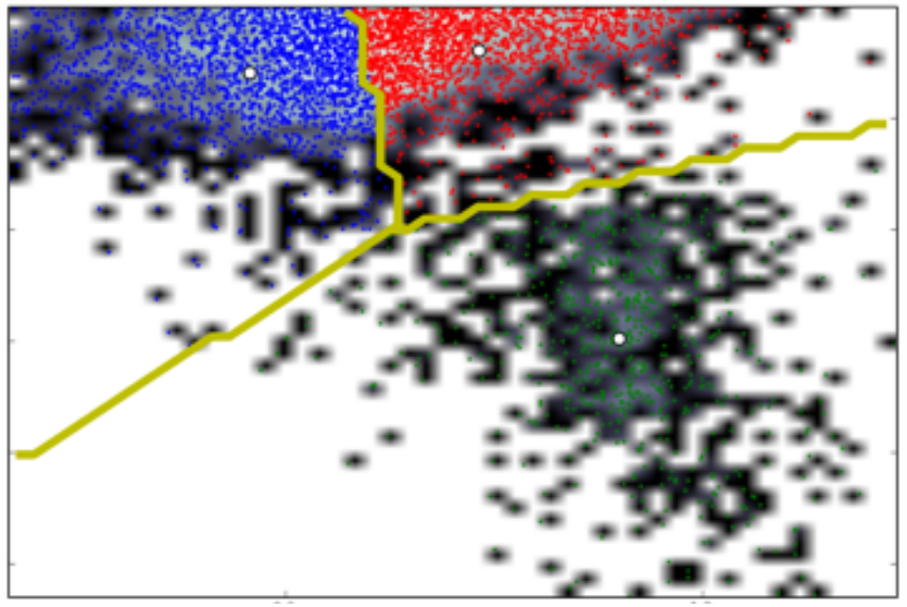


K-means

Must declare the number of clusters upfront



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:

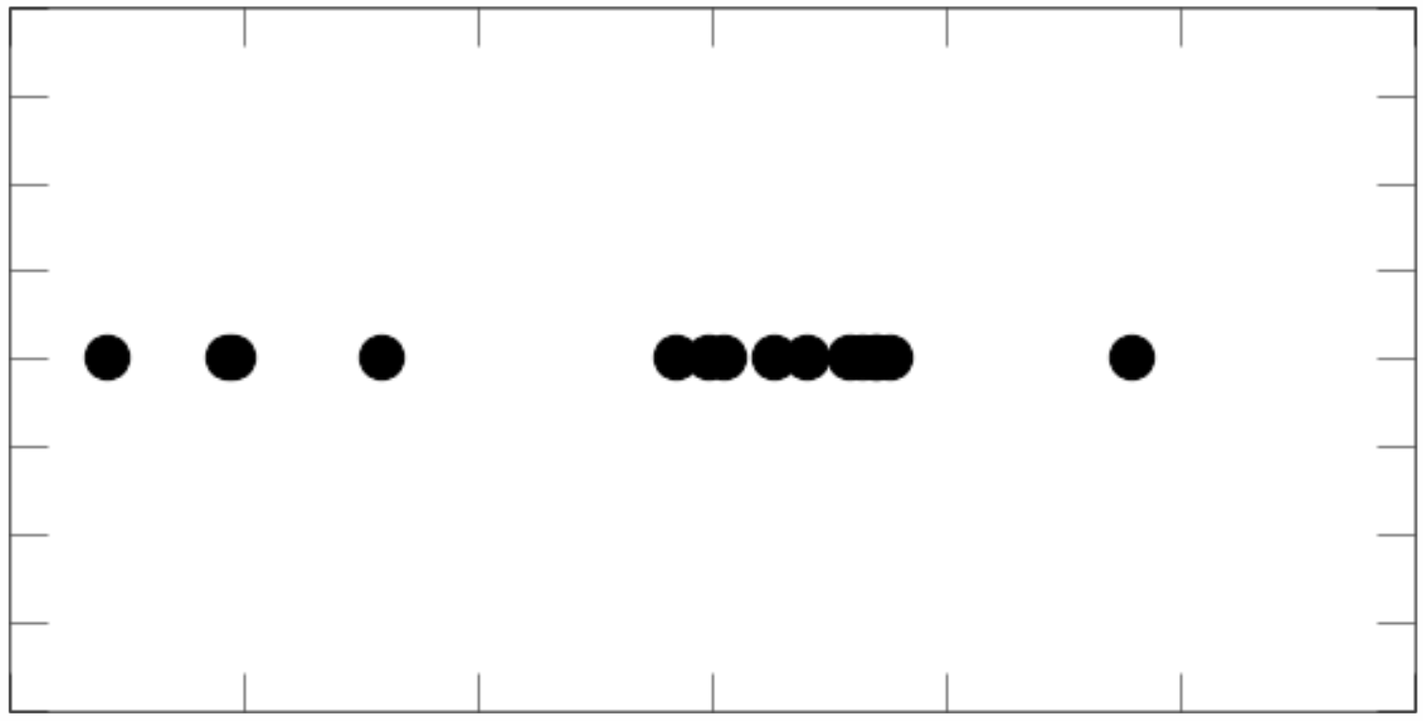
each object in the sample belongs to only 1 cluster

Soft Clustering:

to each object in the sample we assign a probability of belonging to each cluster

Mixture models

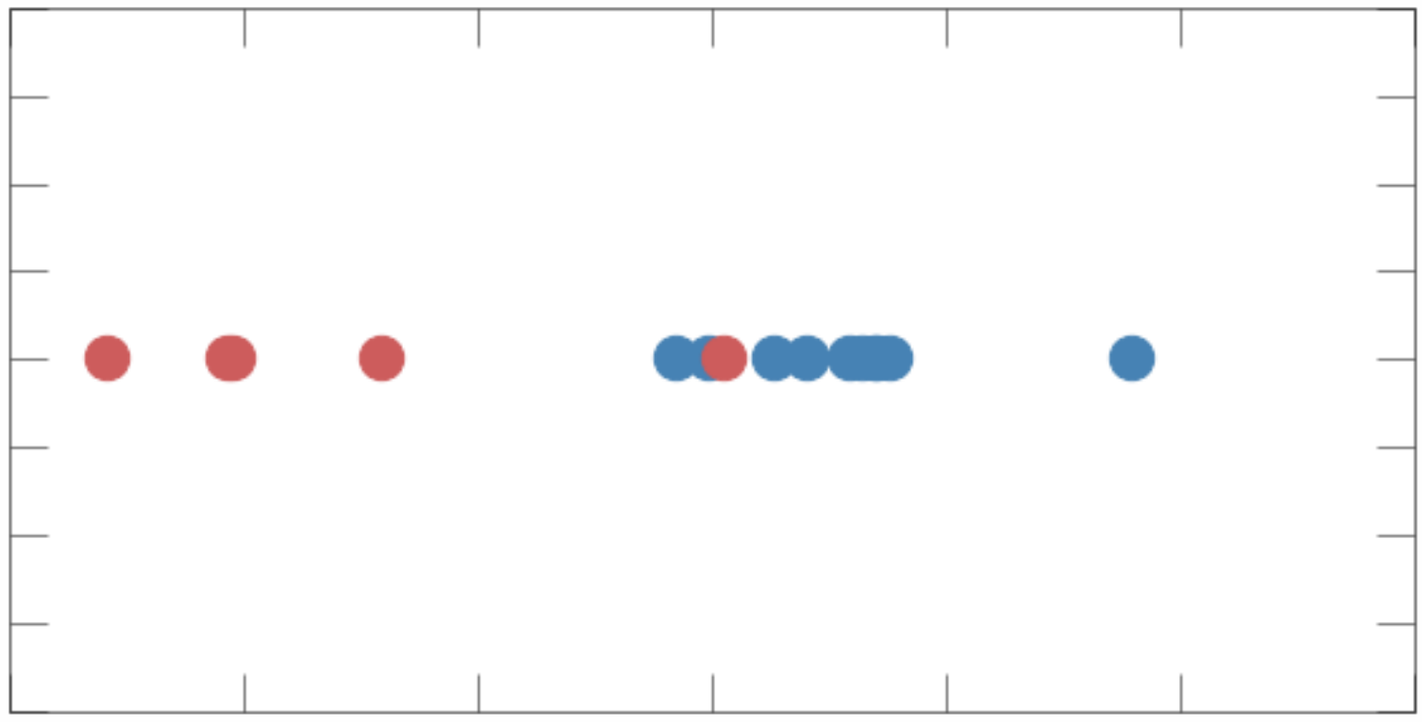
A probabilistic way to do clustering



These points come from 2 gaussian distributions.
Which point comes from which gaussian?

Mixture models

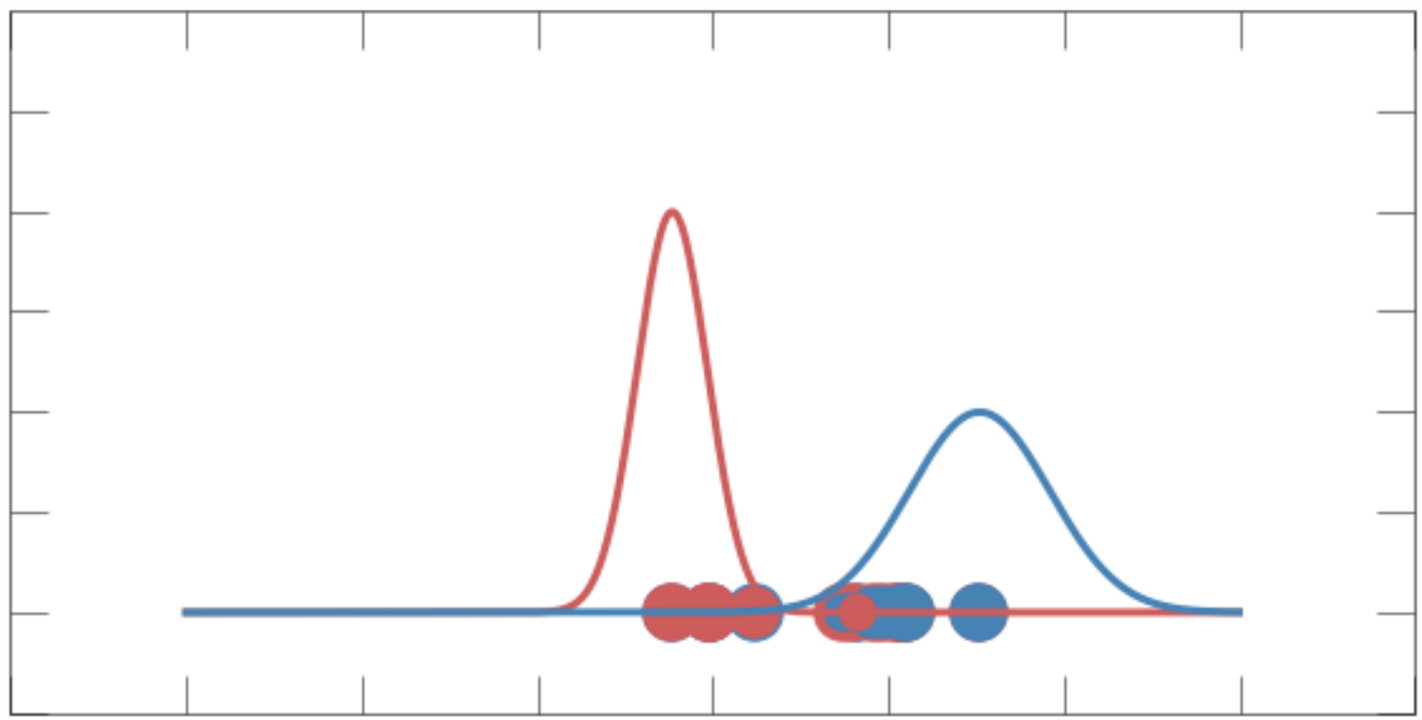
A probabilistic way to do clustering



If I know which point comes from which gaussian
I can solve for the parameters of the gaussians
(e.g. maximizing likelihood)

Mixture models

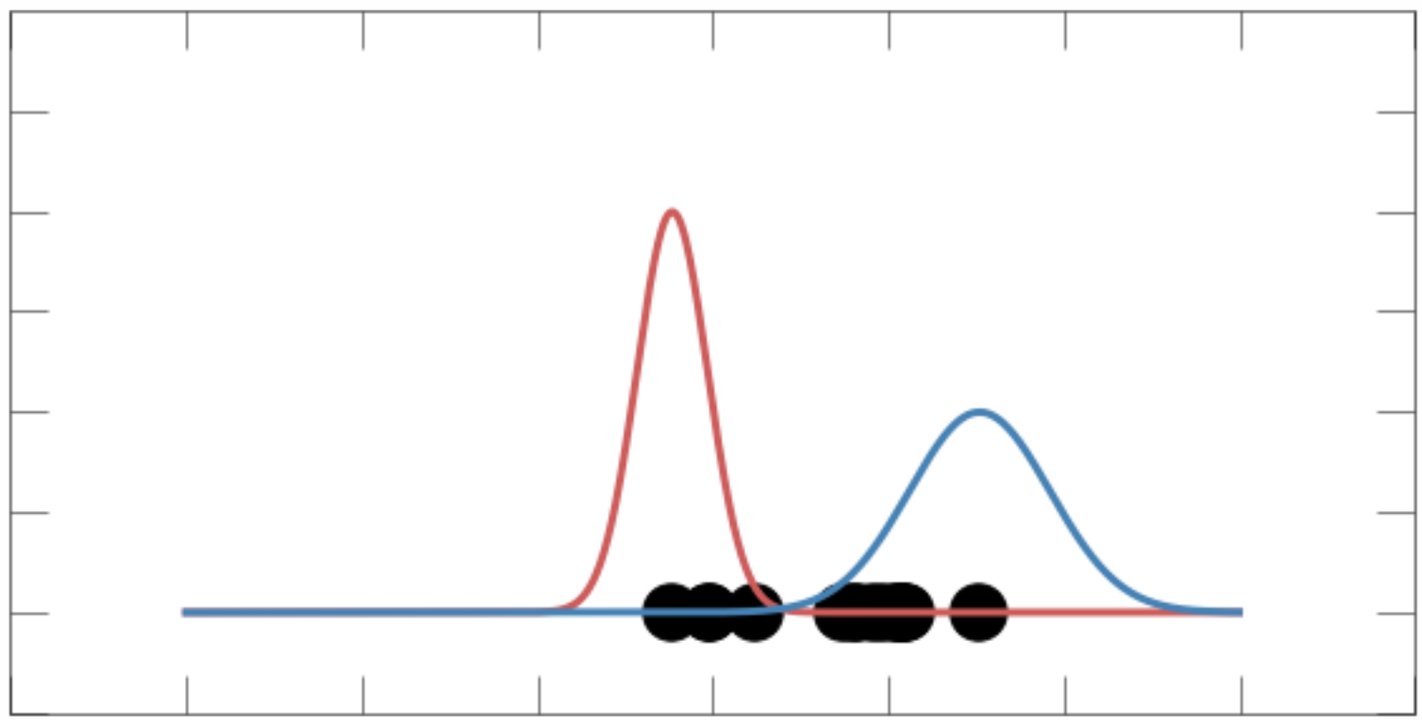
A probabilistic way to do clustering



If I know which point comes from which gaussian
I can solve for the parameters of the gaussians
(e.g. maximizing likelihood)

Mixture models

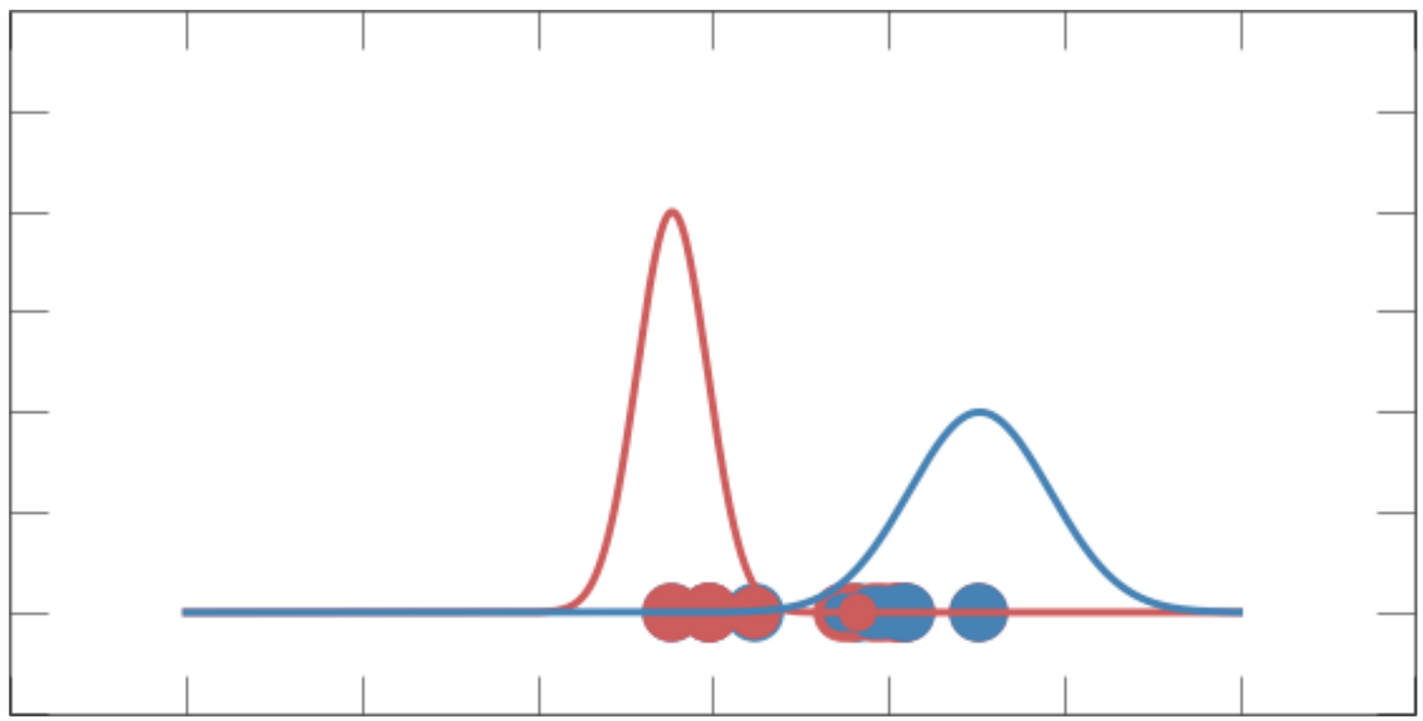
A probabilistic way to do clustering



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

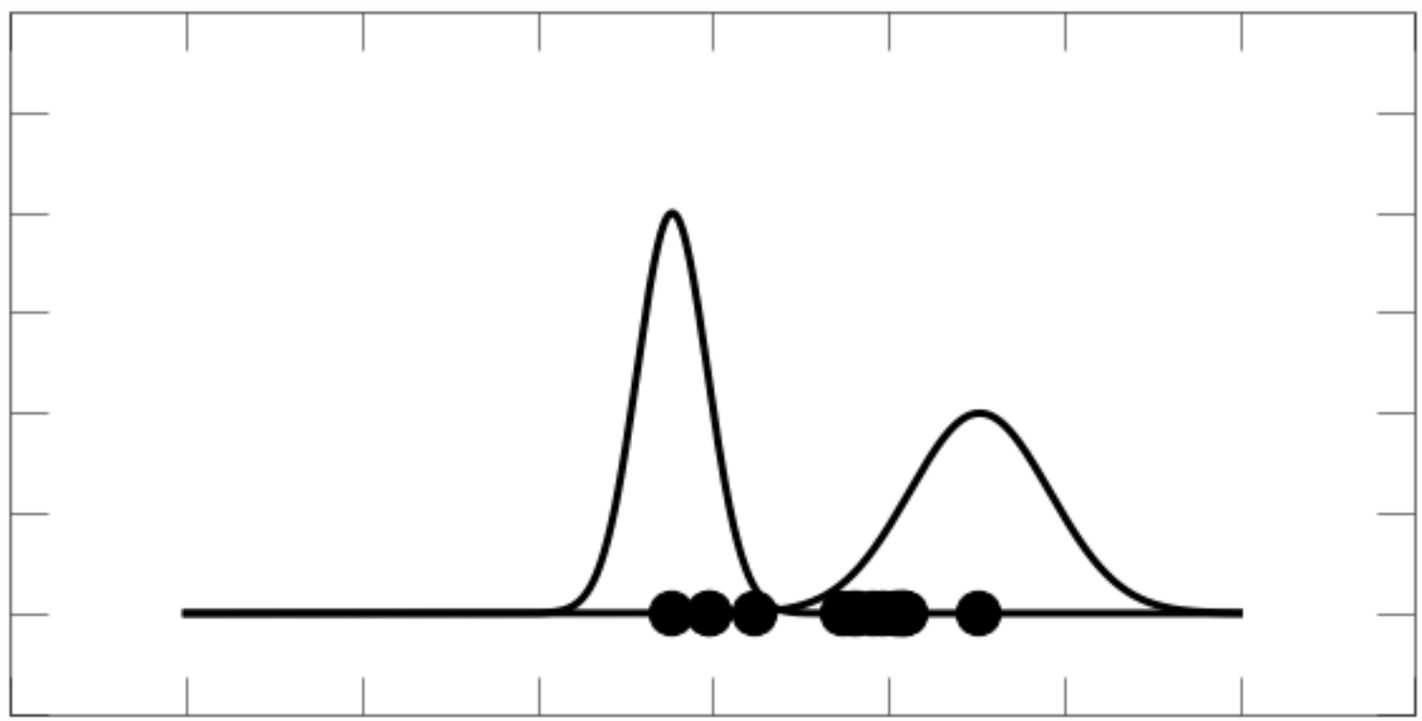
Mixture models

A probabilistic way to do clustering



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

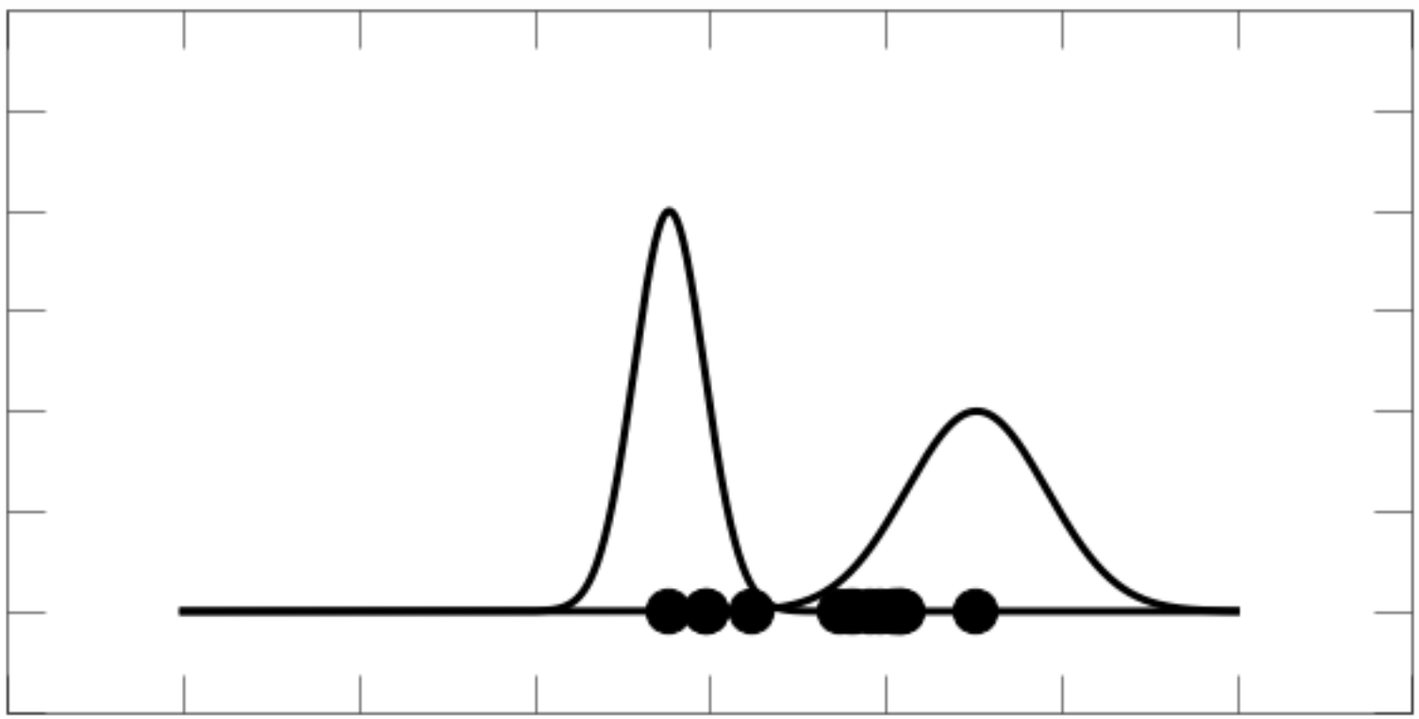
Expectation Maximization



1. guess parameters for 2 gaussian: $\mu_1,\sigma_1; \mu_2,\sigma_2$

Expectation Maximization

1. guess parameters for 2 gaussian: $\mu_1,\sigma_1;\ \mu_2,\sigma_2$

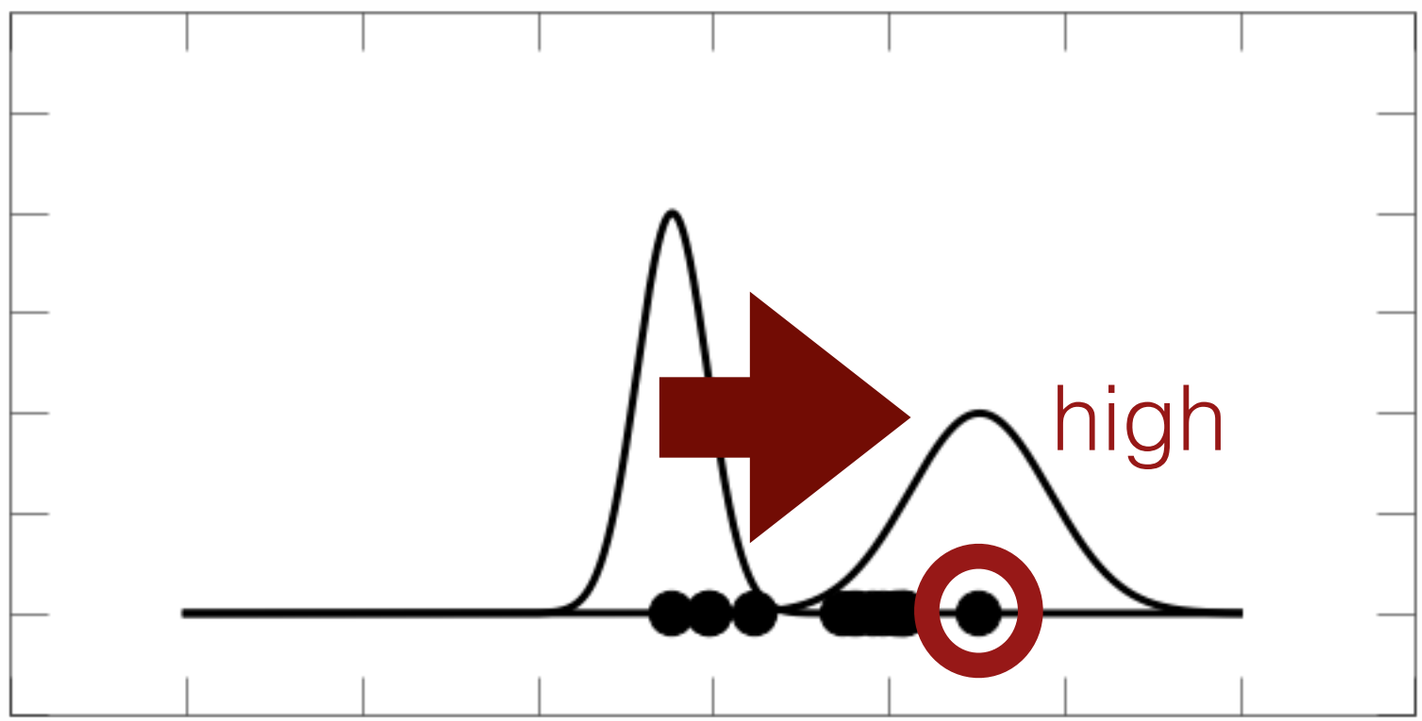


2. for every point calculate probability it comes from either gaussian

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

Expectation Maximization

1. guess parameters for 2 gaussian: $\mu_1,\sigma_1;\ \mu_2,\sigma_2$

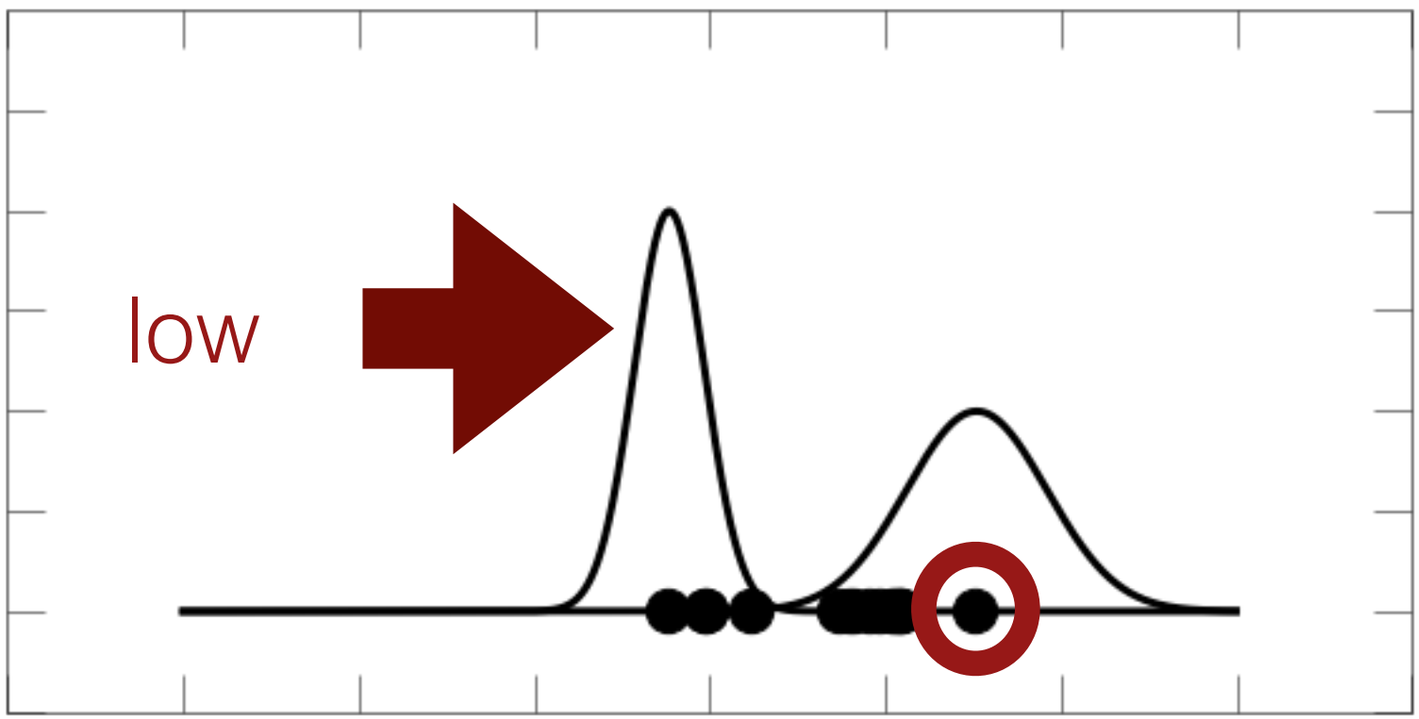


2. for every point calculate probability it comes from either gaussian

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

Expectation Maximization

1. guess parameters for 2 gaussian: $\mu_1,\sigma_1;\ \mu_2,\sigma_2$

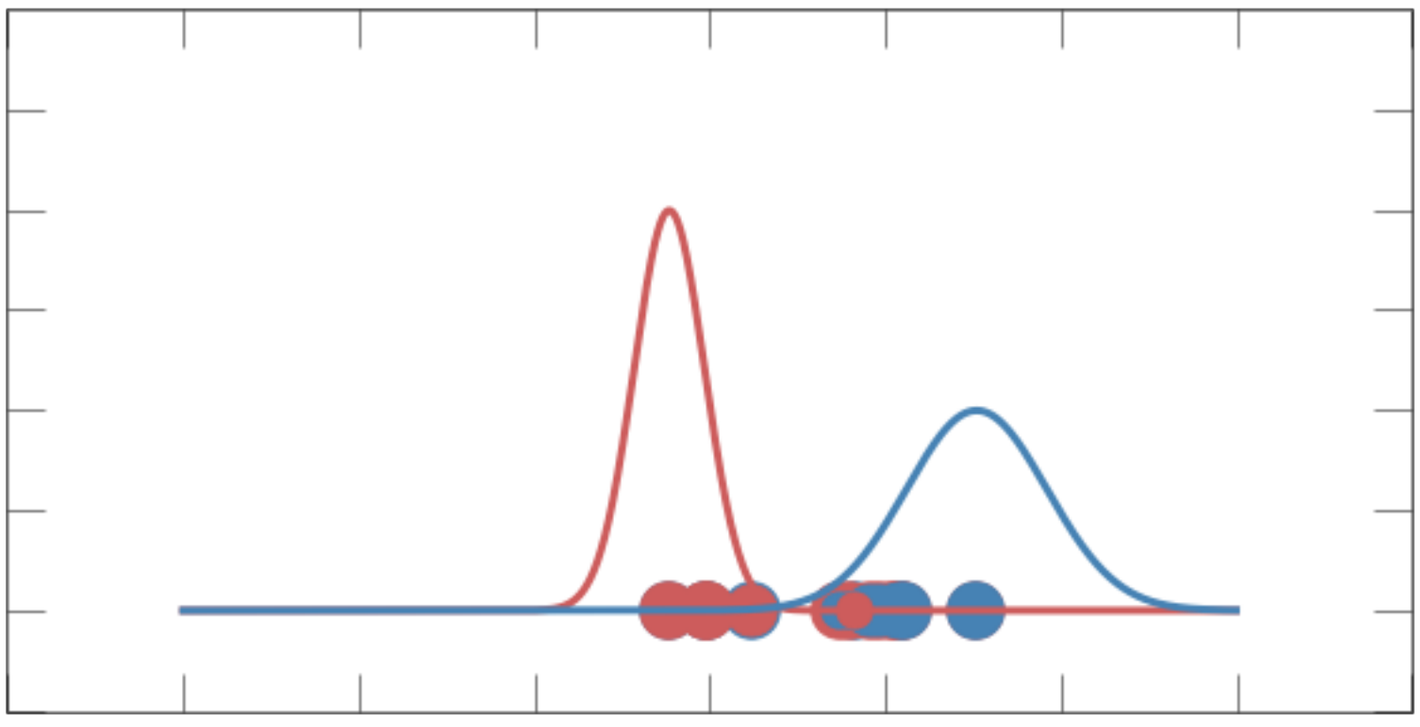


2. for every point calculate probability it comes from either gaussian

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

Expectation Maximization

1. guess parameters for 2 gaussian: $\mu_1,\sigma_1;\ \mu_2,\sigma_2$
2. for every point calculate the probability it comes from either gaussian

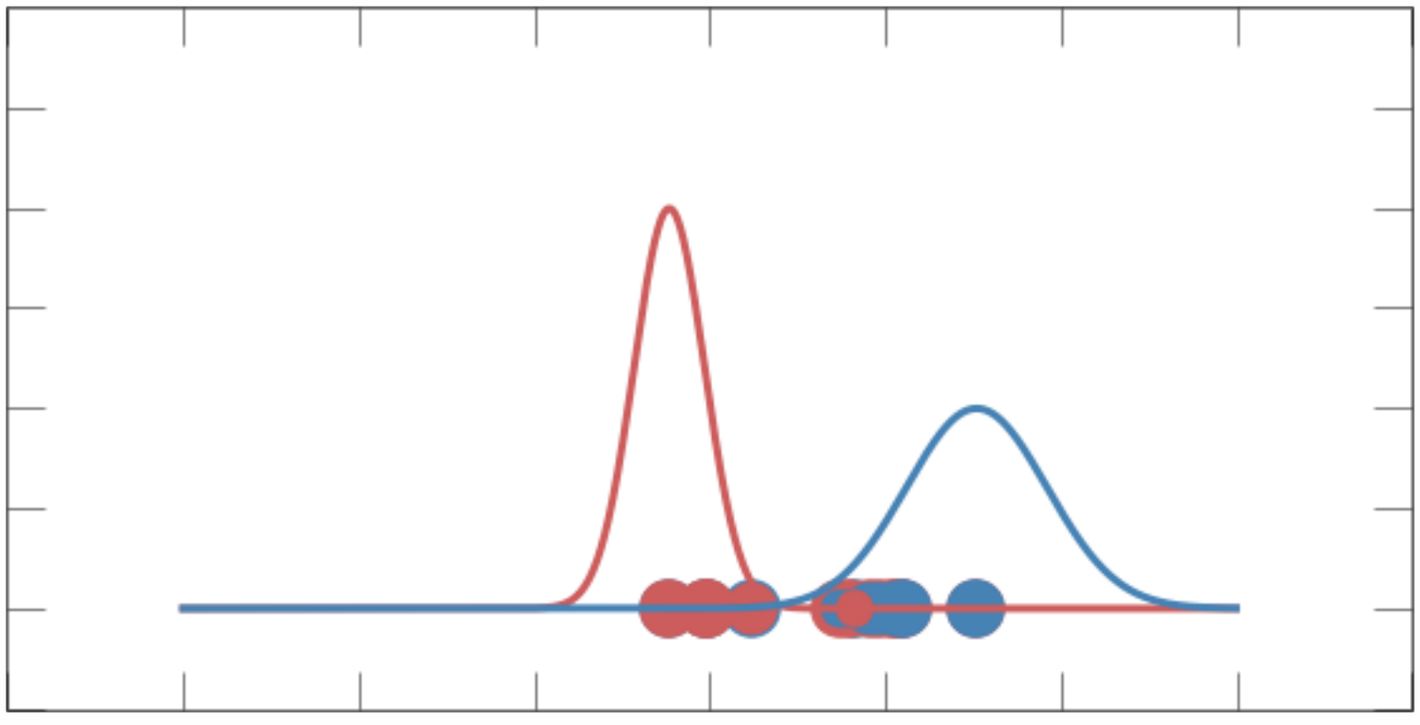


3. with Bayes theorem calculate the probability of model given data

$$P(\mu_j,\sigma_j|x_i) = \frac{P(x_i|\mu_j,\sigma_j^2)P(\mu_j,\sigma_j)}{\sum_j P(x_i|\mu_j,\sigma_j^2)P(\mu_j,\sigma_j)}$$

Expectation Maximization

1. guess parameters for 2 gaussian: $\mu_1,\sigma_1;\ \mu_2,\sigma_2$
2. for every point calculate the probability it comes from either gaussian
3. with Bayes theorem calculate the probability of model given data

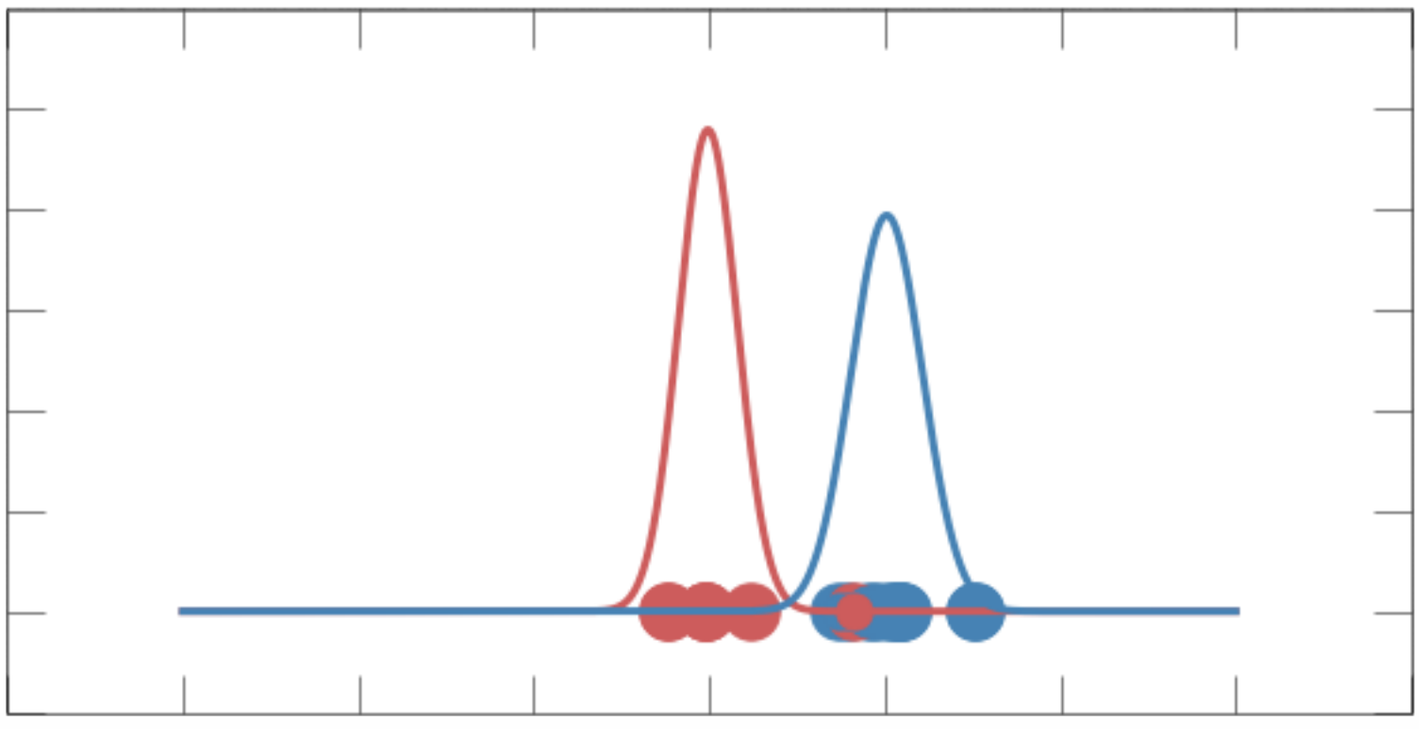


4. use the $P(\mu_j,\sigma_j|x_i)$ as weights to calculate new model parameters

$$\mu_{j,new} = \frac{\sum_j P(\mu_j,\sigma_j|x_i)\cdot x_i}{\sum_j P(\mu_j,\sigma_j|x_i)},\quad \sigma^2_{j,new} = \frac{\sum_j P(\mu_j,\sigma_j|x_i)\cdot (x_i-\mu_j)^2}{\sum_j P(\mu_j,\sigma_j|x_i)}$$

Expectation Maximization

1. guess parameters for 2 gaussian: $\mu_1,\sigma_1;\ \mu_2,\sigma_2$
2. for every point calculate the probability it comes from either gaussian
3. with Bayes theorem calculate the probability of model given data



4. use the $P(\mu_j,\sigma_j|x_i)$ as weights to calculate new model parameters
5. repeat steps 2-4 till convergence

Expectation Maximization Algorithm

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 data-points, weighted by the probabilities**

4. **Iterate 2&3 till convergence**: when gaussian parameters no longer change

Expectation Maximization Pros-Cons

Scalability: #clusters #dimensions #iterations #datapoints #parameter ***O(KdNp)***

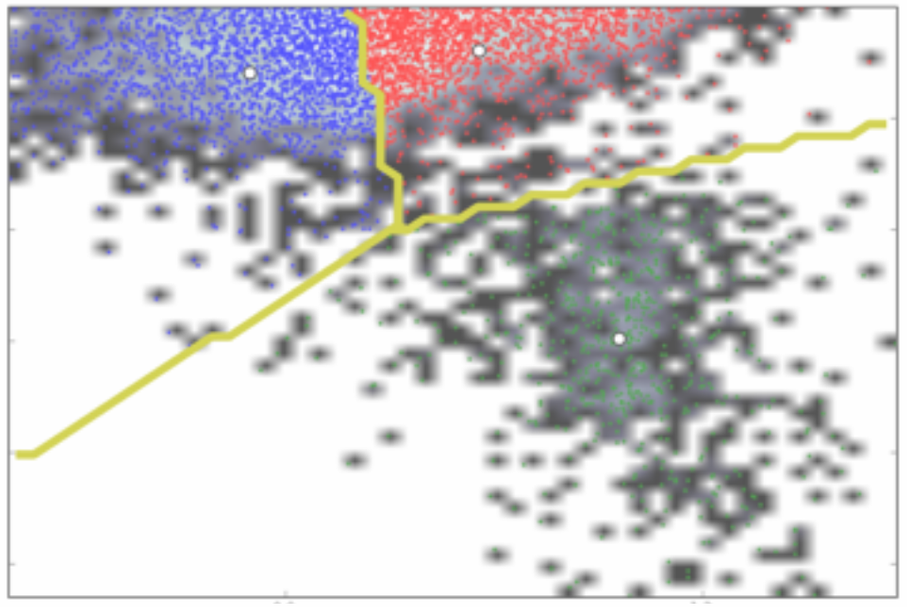
based on Bayes theorem

It's non-deterministic: the result depends on the (random) starting point

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

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

Partitioning



Hard clustering

K-means (McQueen ’67)

K-medoids (Kaufman & Rausseeuw ’87)

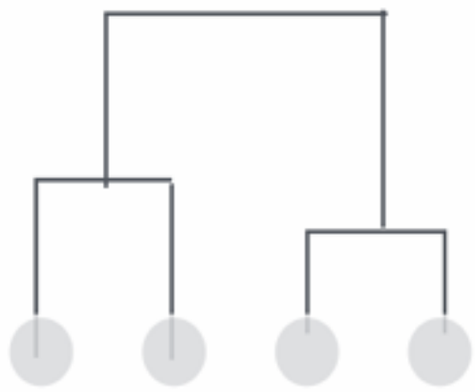
Soft Clustering

Expectation Maximization (Dempster,Laird,Rubin ’77)

Hierarchical

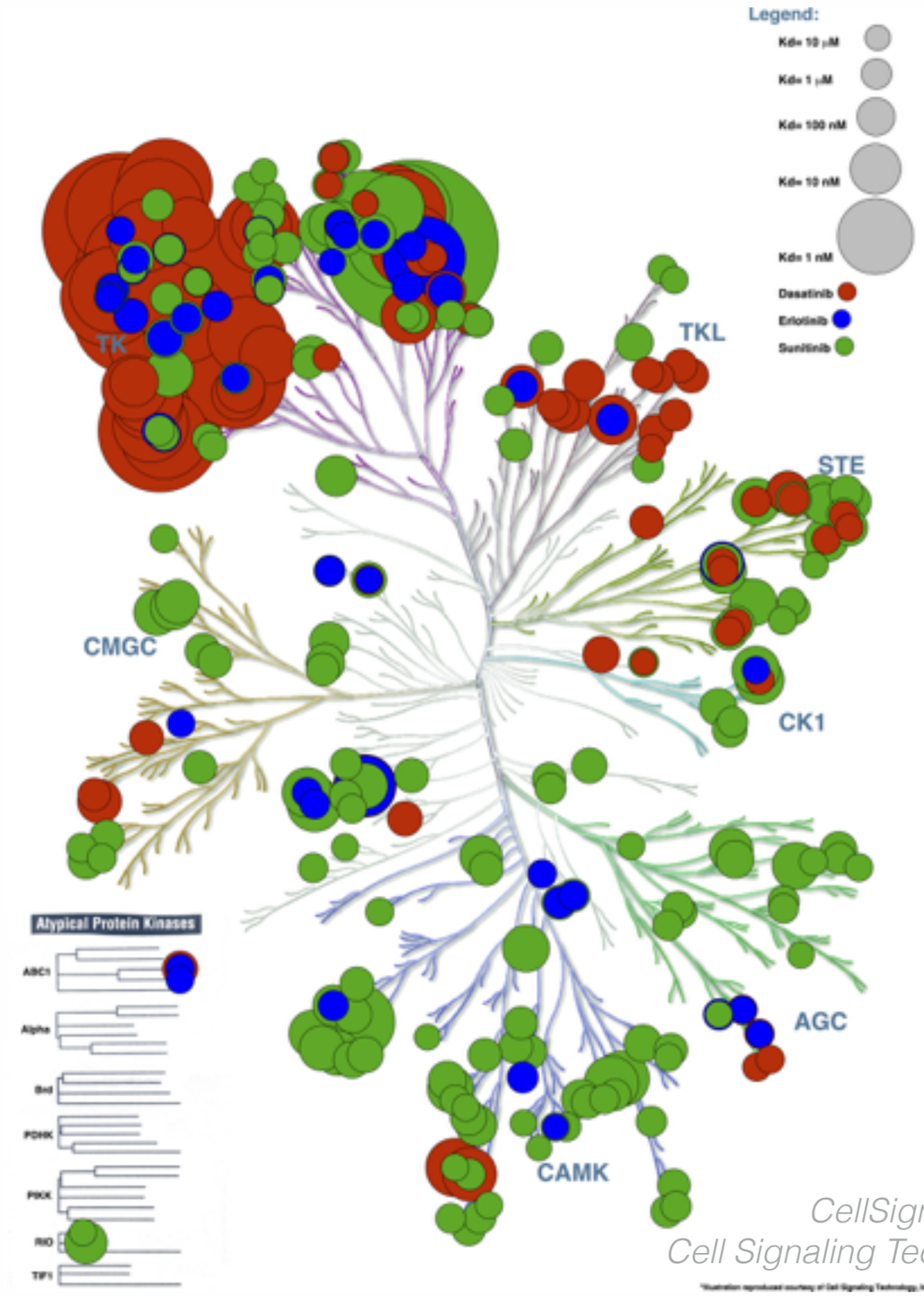
agglomerative

divisive



also: Density based, Grid based, Model based

hierarchical clustering

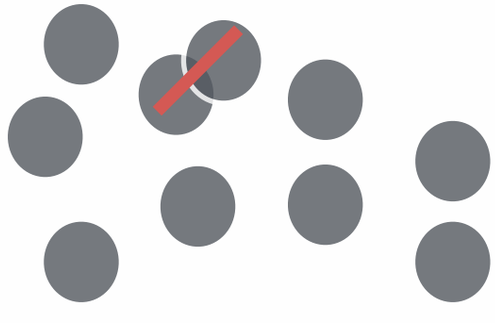


Agglomerative hierarchical clustering

bottom-up:

start with N clusters of 1

end with 1 cluster of N

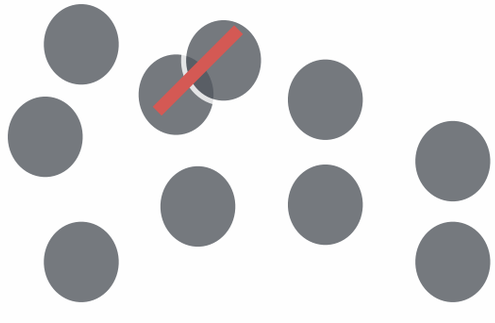


Agglomerative hierarchical clustering

bottom-up:

start with N clusters of 1

end with 1 cluster of N

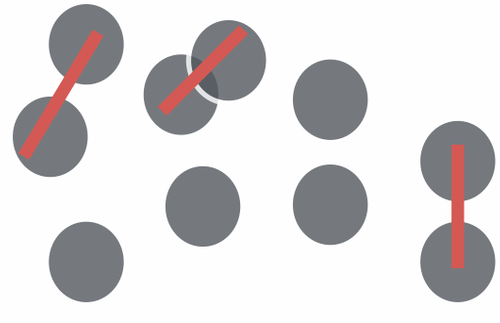


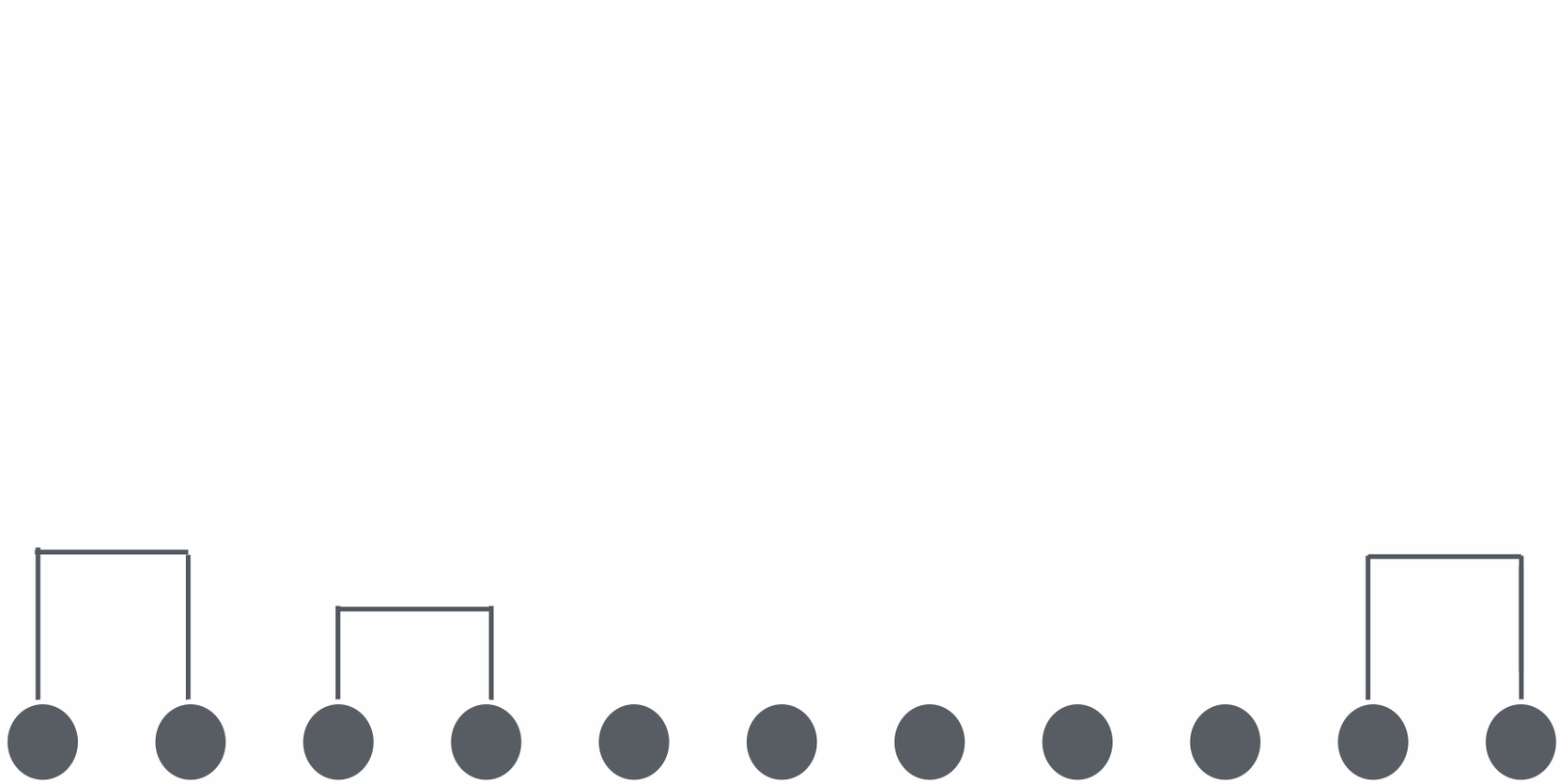
Agglomerative hierarchical clustering

bottom-up:

start with N clusters of 1

end with 1 cluster of N



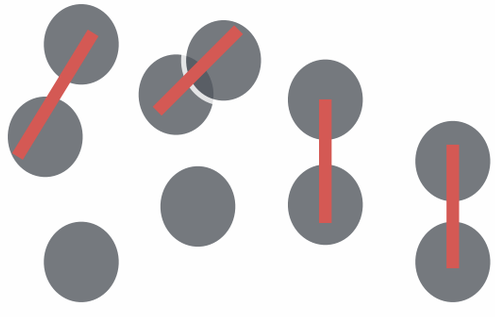


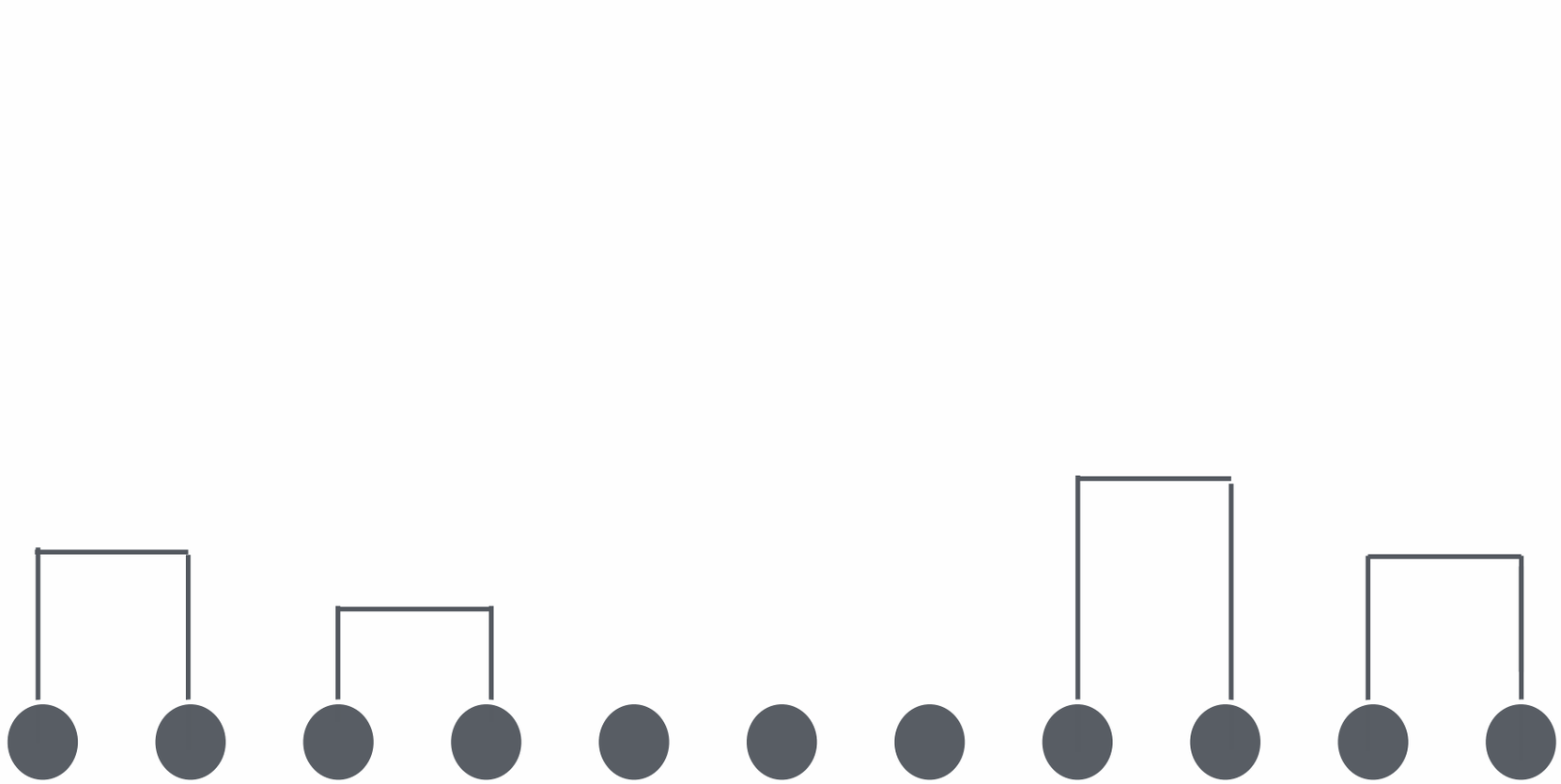
Agglomerative hierarchical clustering

bottom-up:

start with N clusters of 1

end with 1 cluster of N



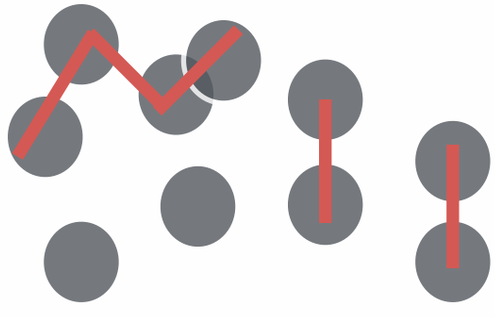


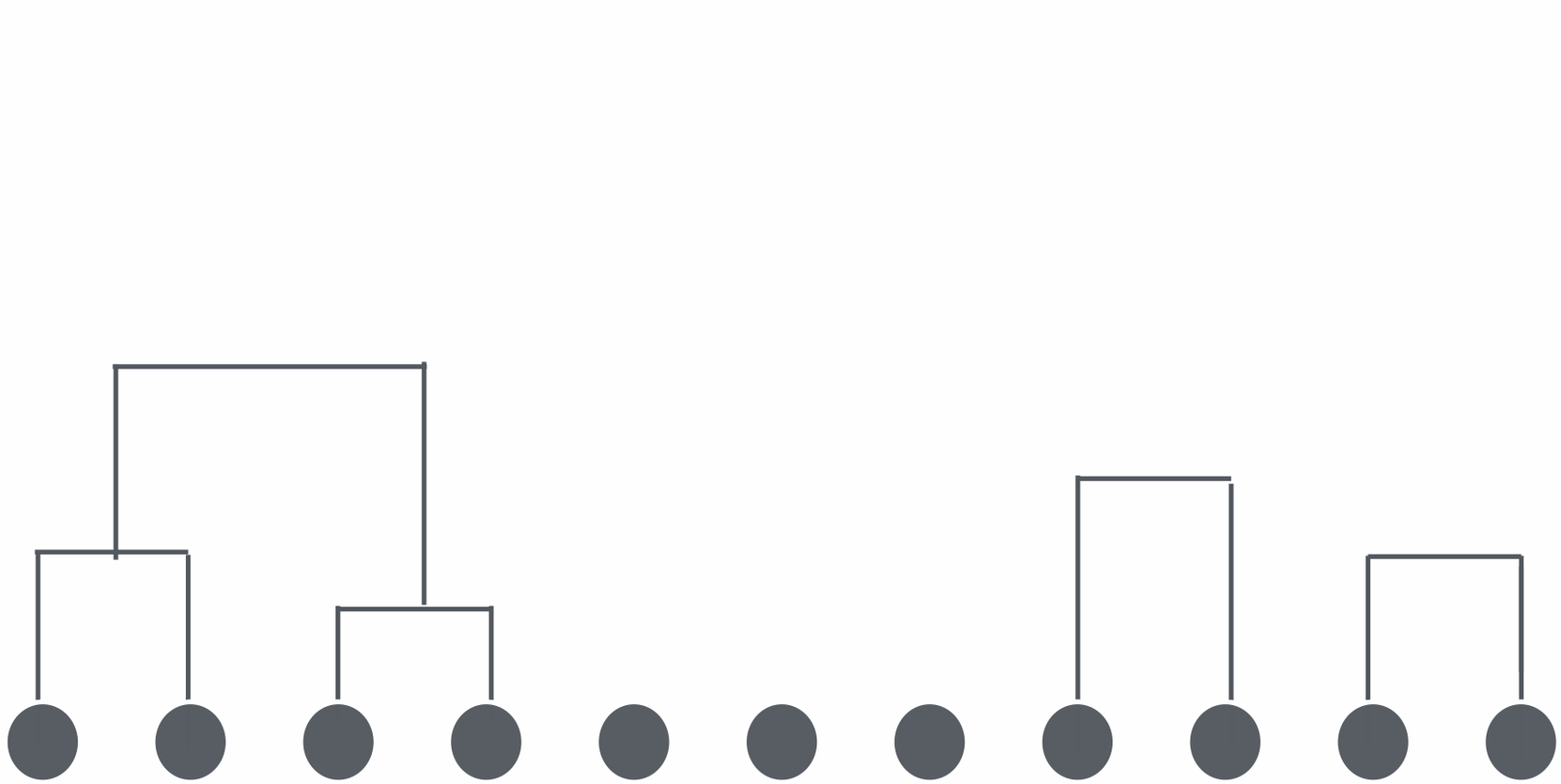
Agglomerative hierarchical clustering

bottom-up:

start with N clusters of 1

end with 1 cluster of N



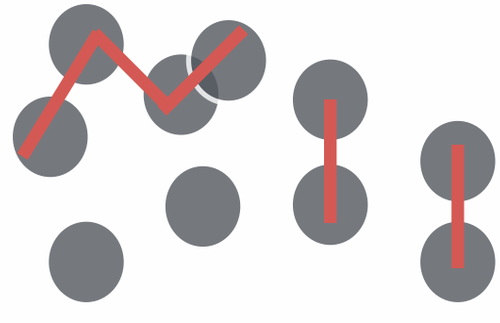


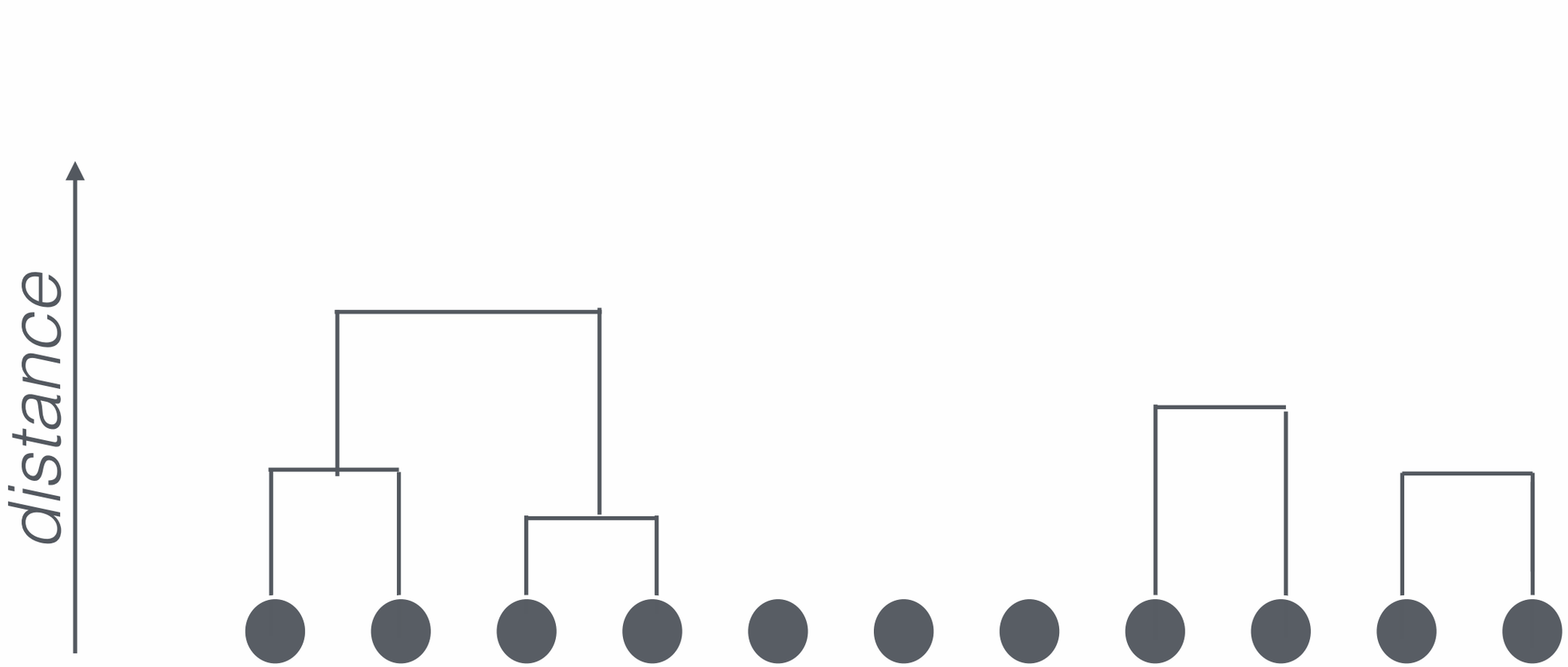
Agglomerative hierarchical clustering

bottom-up:

start with N clusters of 1

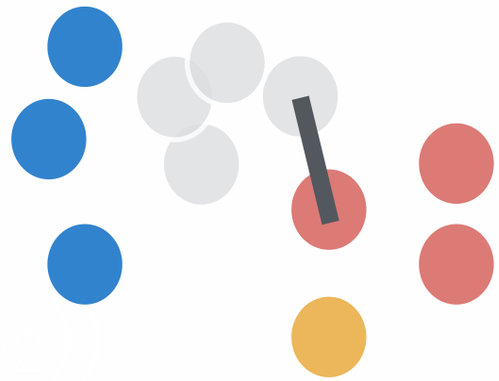
end with 1 cluster of N





Linkage

How to measure the distance between clusters

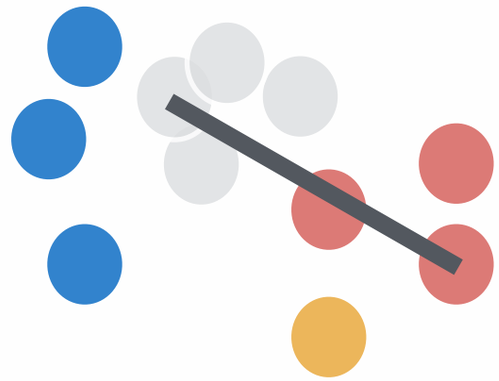


single linkage

$$min(D(ij)_{i \in I, j \in J})$$

Linkage

How to measure the distance between clusters

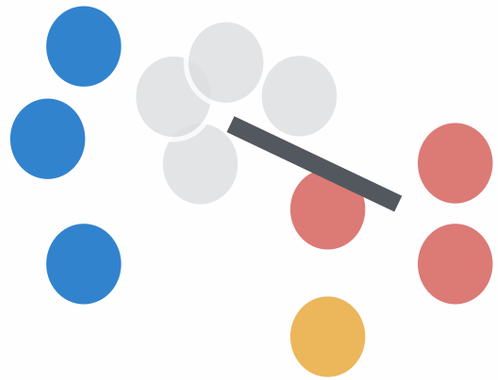


complete linkage

$$max(D(ij)_{i \in I, j \in J})$$

Linkage

How to measure the distance between clusters



average linkage

$$\sum_{j} \sum_{i, x_i \in J} (x_i - \mu_j)^2$$


jupyter

Agglomerative hierarchical clustering Algorithm

1. Start with clusters of 1
2. Compute the distance to each pair of clusters
3. Cluster closest pairs of clusters
4. Repeat 2-3 till all clusters have been merged

Agglomerative hierarchical clustering Pros-Cons

Scalability: #datapoints^2 #dimensions or #datapoints^3

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

Allows user to inspect the hierarchical structure

Deterministic: the dendrogram is always reproduced

The ideal clustering algorithm:

- Scalability (naive algorithms are Np 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
- Allows incorporation of constraints
- Interpretable

But an analysis can be fully reproducible and still be wrong

[...]

We have found that the most frequent failure in data analysis is mistaking the type of question being considered.

SHARE PerspectiveSTATISTICS

What is the question?

Jeffery T. Leek, Roger D. Peng
Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD 21205, US
E-mail: jleek{at}jhsph.edu, jtleek{at}gmail.com
Science 20 Mar 2015:
Vol. 347, Issue 6228, pp. 1314-1315
DOI: 10.1126/science.aaa6146

Science

Home News Journals T

Latest News ScienceInsider ScienceShots

PerspectiveSTATISTICS

What is the question?

Jeffery T. Leek, Roger D. Peng

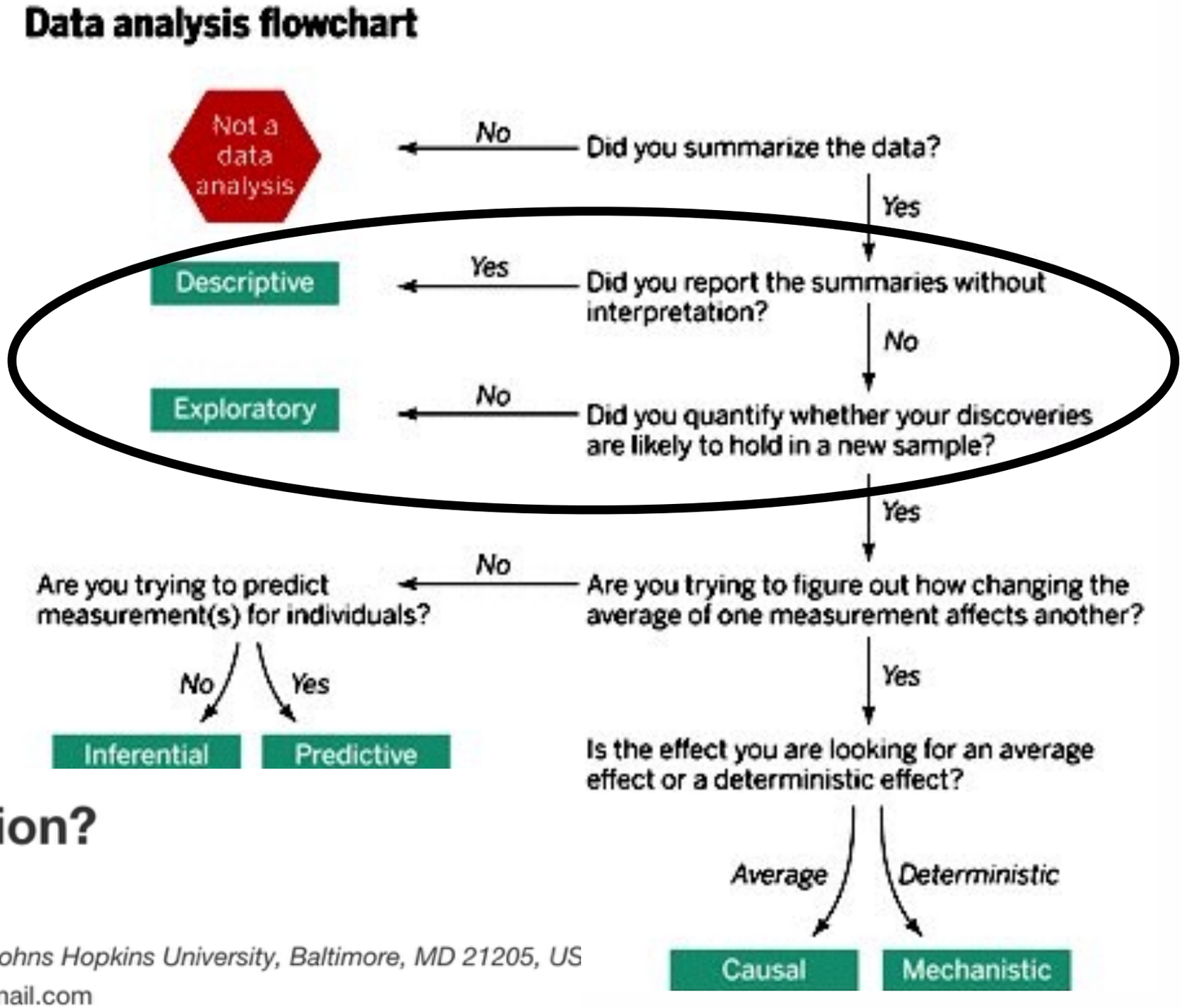
Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD 21205, US

E-mail: jleek{at}jhsph.edu, jtleek{at}gmail.com

Science 20 Mar 2015:

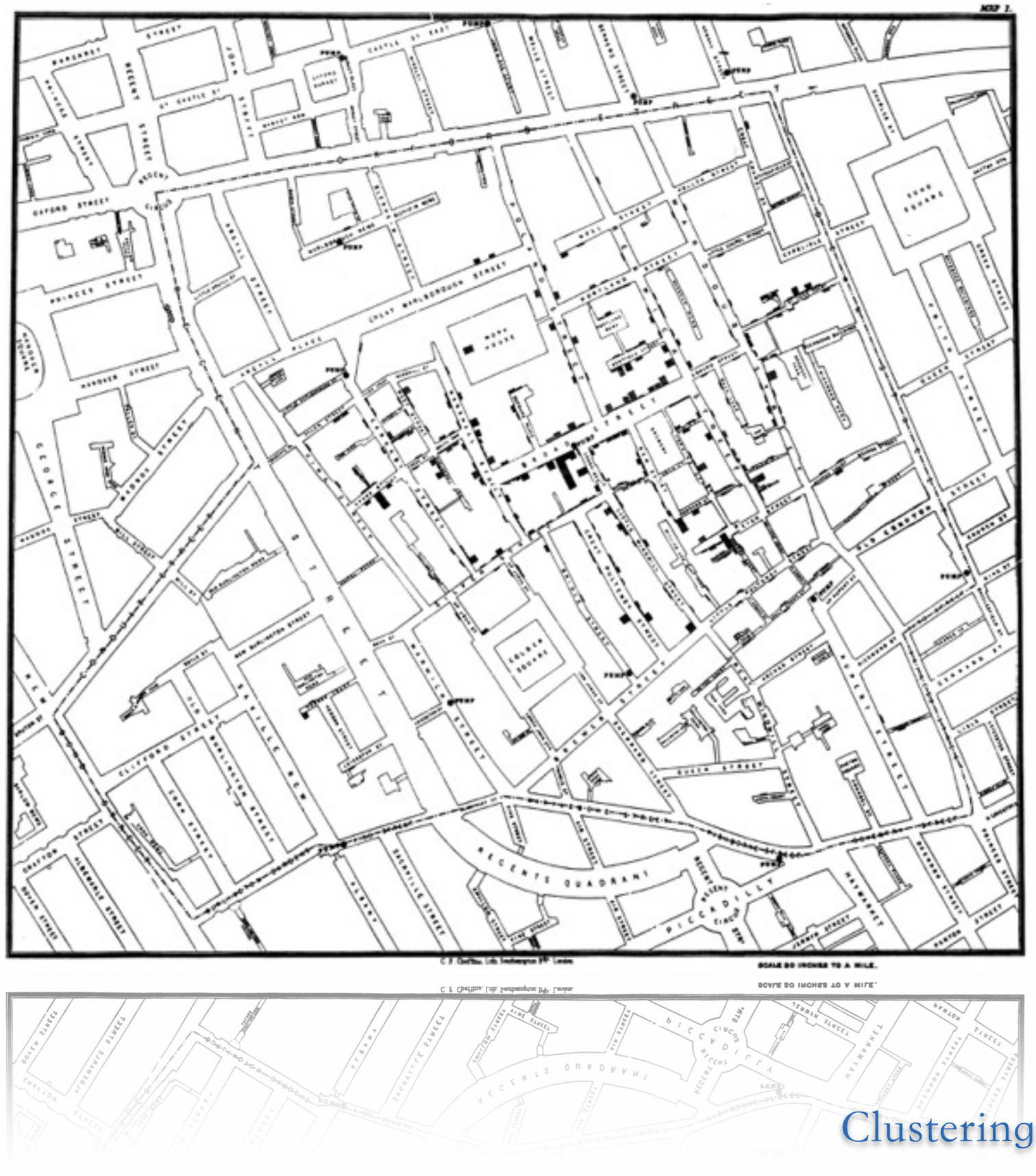
Vol. 347, Issue 6228, pp. 1314-1315

DOI: 10.1126/science.aaa6146



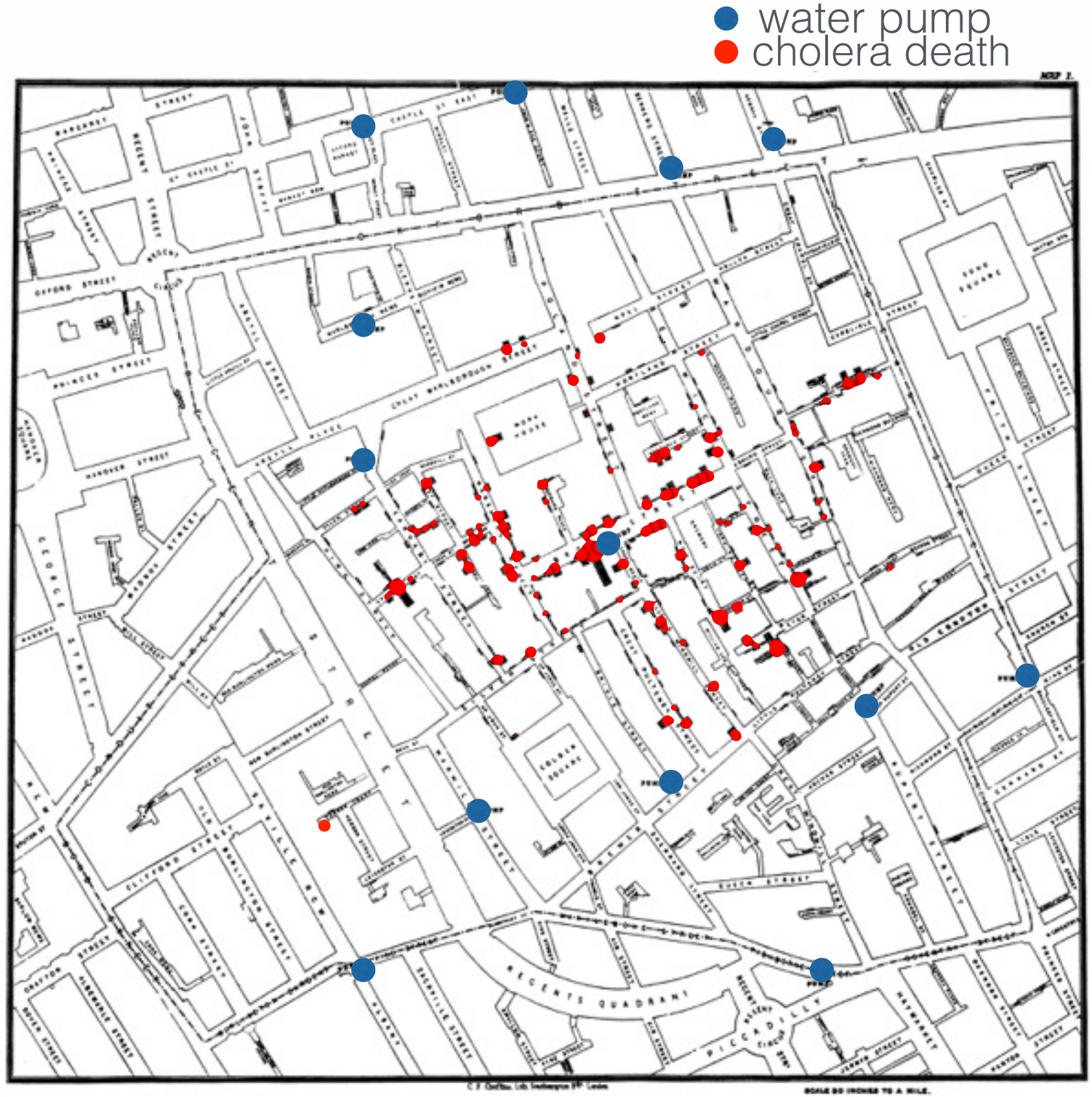
Dr. John Snow's map showing the clusters of cholera cases in the London epidemic of 1854.
A water pump is located at the center of the cluster.

Map drawn+lithographed by Charles Cheffins.



Dr. John Snow's map showing the clusters of cholera cases in the London epidemic of 1854.
A water pump is located at the center of the cluster.

Map drawn+lithographed by Charles Cheffins.



Summary and Key concepts

machine learning exploits robots to do human jobs

algorithms that can learn from and make predictions on data

Supervised Learning:

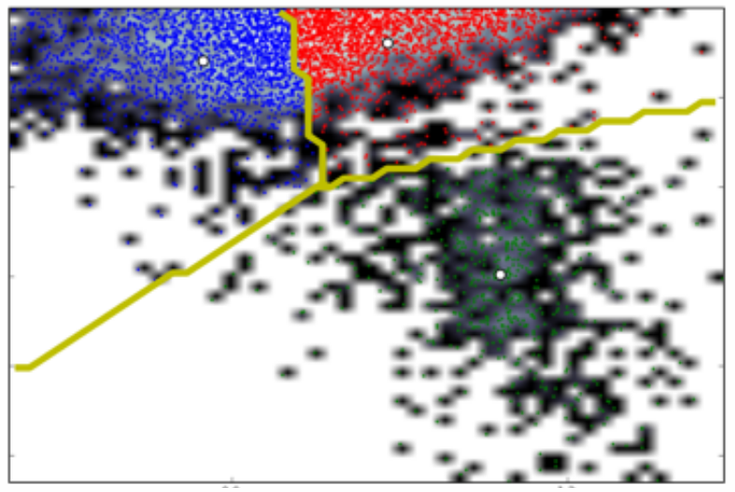
Data is analyzed on the basis of a subset of data that that was labelled by humans.
Labelling can be extended or enriched.

Unsupervised Learning:

Explores data structure, no labelled data.

Summary and Key concepts

clustering is easy,
but interpreting results is tricky



Human input in choice of features,
definition of distance metric + linkage

Distance metrics:
Euclidean and other Minkowski metrics
angular distances
correlations and metrics for non continuous data

Partitioning methods: inexpensive, typically non deterministic
Hard (or crisp) methods: *K-means, K-medoids*
Soft (or fuzzy) methods: (i.e. probabilistic approach)
Expectation Maximization Mixture models

Hierarchical methods:
divisive vs agglomerative,
dendrograms for exploration of the cluster hierarchy

RESOURCES:

AstroML Chapter 5 *Unsupervised Learning*

Three Types of Gamma-Ray Bursts
Mukherjee, Feigelson, Babu, Murtagh, Fraley, Raftery,
The Astrophysical Journal, Volume 508, Issue 1, pp. 314-327.
excellent application clear/detailed explenation of theory + practical issues

a comprehensive review of clustering methods
Data Clustering: A Review, Jain, Mutry, Flynn 1999
https://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf

a blog post on how to generate and interpret a python dendrogram with scipy by Jörn Hees
https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/

RESOURCES:

class notebook : https://github.com/fedhere/clustering

class slides : https://github.com/fedhere/clustering

HW1: fiddle with example code

1. improve the code efficiency: replace iterations with slicing, use local variables, remove all redundant calculations from inside loops… measure the coputational efficiency as CPUS and memory usage

HW2: cluster SDSS QSOs - color, spectra

Using the data available in the SDSS QSO and stars data (which you can get through AstroML) find the feature space and distance metric that:

1. Supports the highest number of clusters
2. Best separates the QSOs from the other objects (largest distance)
3. Produces the least contaminated QSOs catalog compared to the labelled sample *(semi-supervised learning)*

Use both *k-means* and another clustering method of your choice (e.g. sklearn offers *agglomerative* and *DBSCAN*) for at least 1 tast and compare.

DATA: photometry SDSS Galaxy colors, SDSS Imaging data
(optional spectra: SDSS QSO spectra, SDSS spectra)