OUTLINE

Data Integration

- **Different schemas** →Schema matching
- **Duplicates** → Entity resolution
- **Contradicting data** → data fusion

Data Cleaning

- **Missing values** → Value imputation
- **Wrong data**
- **Missing records** → Species estimation

WRONG DATA: RULE-BASED APPROACHES

Name	Address	#Employees	Revenue	Profit
Google	1600 Amphitheatre Parkway, Mountain View, CA, 94043, USA	60k	$89B	null
Apple	1 Infinite Loop; Cupertino, CA 95014, USA	66	$215B	$45B
IBM	1 New Orchard Rd; New York 10504, USA	380k	$80B	$12B
International Business Machine	10504; 1 New Orchard Rd	380k	$-999B	$12B
Microsoft	Albuquerque, Mexico	120k	$85B	$85B
Tableau	Seattle, Washington, United States	-	$0.9B	$1B
Tamr	[illegible], United States	20	null	[illegible]

Encoding Error
(nb in thousands)

Rule Violations

Outdated data / wrong data

Spelling mistakes / abbreviations

TWO COMPONENTS

1. **Detection**
2. **Repair**
   - Detection techniques can be used for repair
   - Missing value techniques

Ideas on how to detect wrong data?

COMMON TECHNIQUES

- **Consistency Violations**
- **Outliers**
- **Manual Validation / (Crowd-)Sourcing**

ERROR DETECTION

FD: [country] -> [capital]

CFD: [country = China] -> [capital = Beijing]

emp

	name	country	capital	city	salary	tax
r1	Nan	China	Beijing	Beijing	50000	1000
r2	Yin	China	Shanghai	Hongkong	40000	1200
r3	Si	Netherlands	Den Hagg	Utrecht	60000	1400
r4	Lei	Netherlands	Amsterdam	Amsterdam	35000	800

cap

	country	capital
s1	China	Beijing
s2	Canada	Ottawa
s3	…	…

CD:]t1, t2 (t1.salary > t2.salary and t1.tax < t2.tax)

MD: (emp[country] = cap[country]) -> (emp[capital] <=> cap[capital])

COMPUTING A CONSISTENT DATABASE

FD1: [nationality] -> [capital]

FD2: [areacode] -> [capital]

	name	nationality	capital	areacode	bornAt	salary	tax
r1	Nan	China	Beijing	10	Shenyang	50000	1000
r2	Yan	China	Shanghai Beijing	10	Hangzhou	40000	900
r3	Si	China	Beijing	10	Changsha	60000	1400
r4	Miura	China	Tokyo Beijing	3	Kyoto	35000	800

Equivalence class

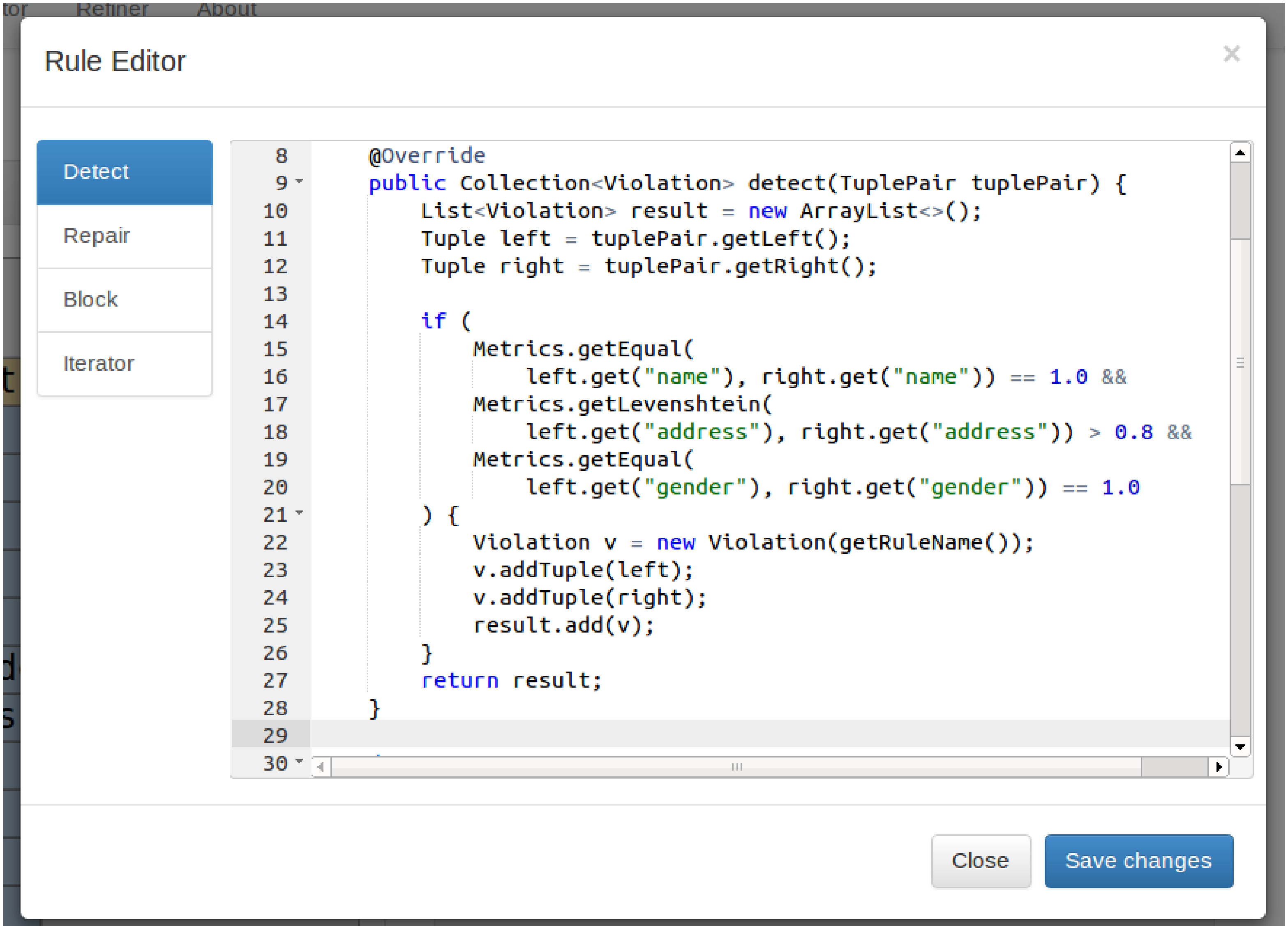
Vertex cover

SAT solver

... ...

NADEEF

https://github.com/daqcri/NADEEF
https://cs.uwaterloo.ca/~ilyas/papers/NADEEFSigmod2013.pdf



OUTLIER DETECTION

ANOMALY/OUTLIER DETECTION

What are anomalies/outliers?

ANOMALY/OUTLIER DETECTION

What are anomalies/outliers?

- The set of data points that are considerably different than the remainder of the data

Variants of Anomaly/Outlier Detection Problems

- Given a database D, find all the data points $\mathbf{x} \in D$ with anomaly scores greater than some threshold t
- Given a database D, find all the data points $\mathbf{x} \in D$ having the top-n largest anomaly scores $f(\mathbf{x})$
- Given a database D, containing mostly normal (but unlabeled) data points, and a test point **x**, compute the anomaly score of **x** with respect to D

Applications:

- Credit card fraud detection, telecommunication fraud detection, network intrusion detection, fault detection

ANOMALY DETECTION

Challenges

- How many outliers are there in the data?
- Method is unsupervised
  - Validation can be quite challenging (just like for clustering)
- Finding needle in a haystack

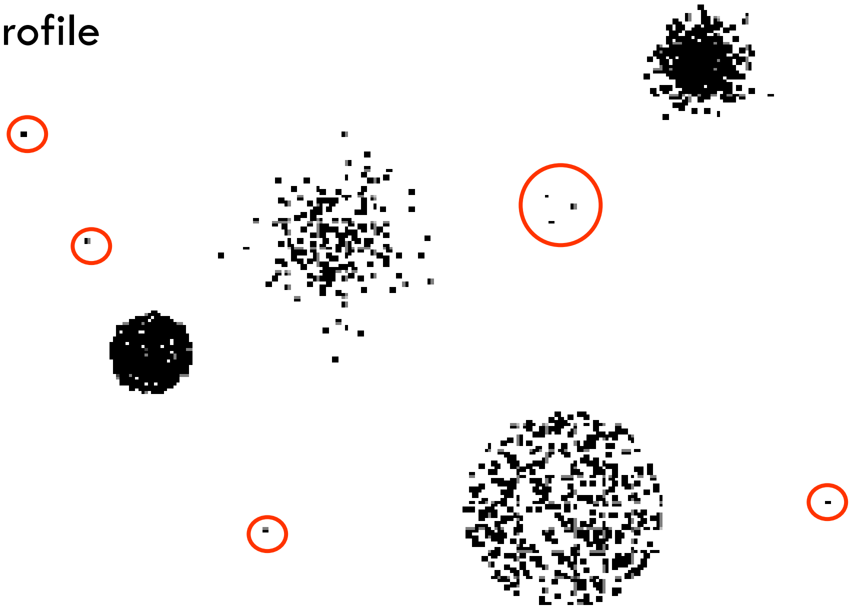
Working assumption:

- There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data

ANOMALY DETECTION SCHEMES

General Steps

- Build a profile of the "normal" behavior
  - Profile can be patterns or summary statistics for the overall population
- Use the "normal" profile to detect anomalies
  - Anomalies are observations whose characteristics differ significantly from the normal profile



Types of anomaly detection schemes

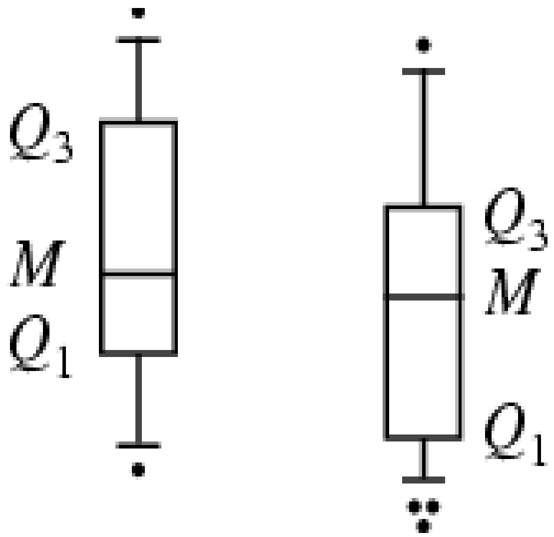
- Graphical
- Model-based
- Distance-based
- Clustering-based

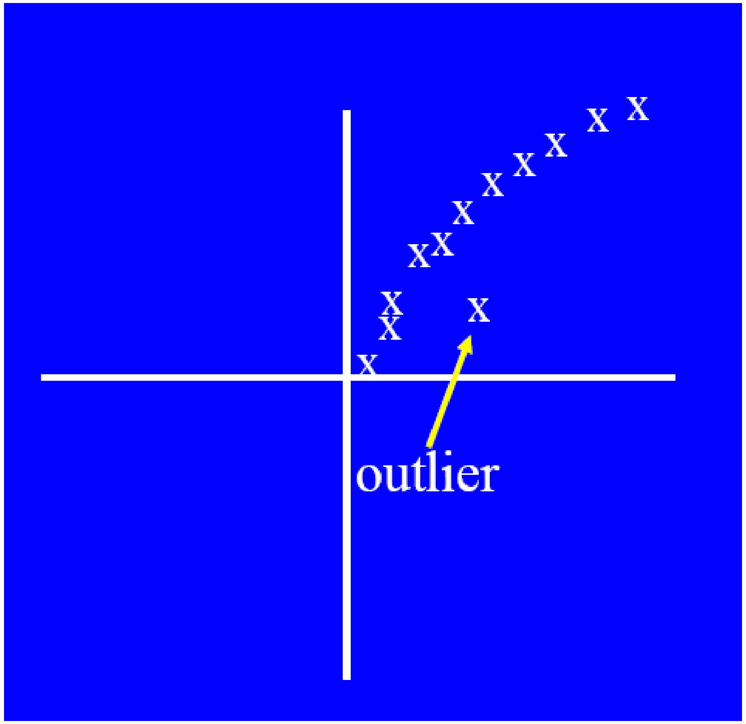
GRAPHICAL APPROACHES

Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)

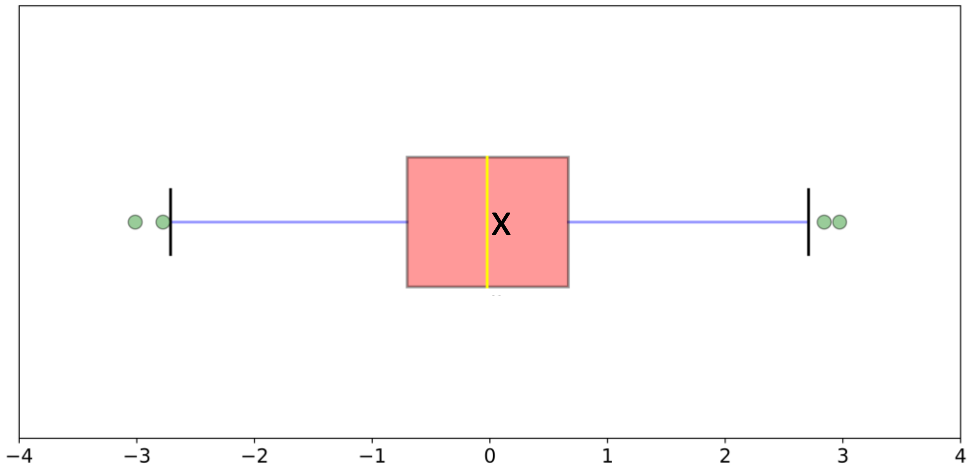
Limitations

- Time consuming
- Subjective

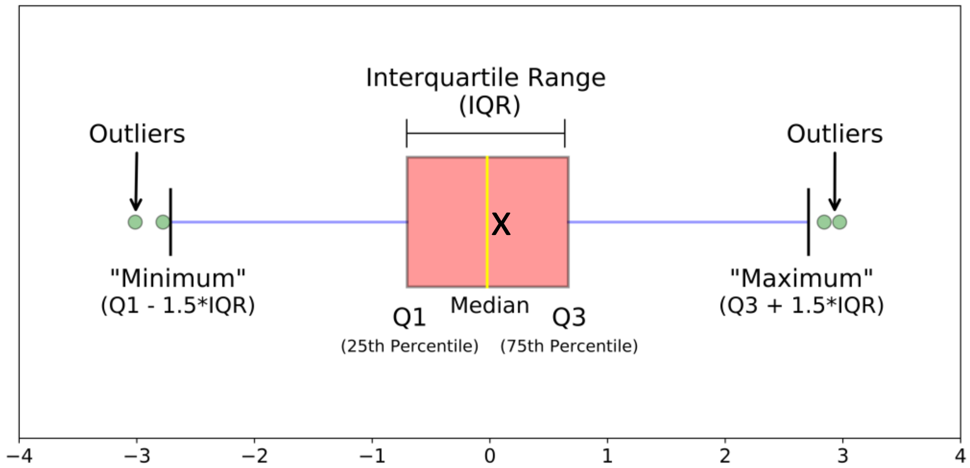




COMPONENTS OF A BOX PLOT



COMPONENTS OF A BOX PLOT



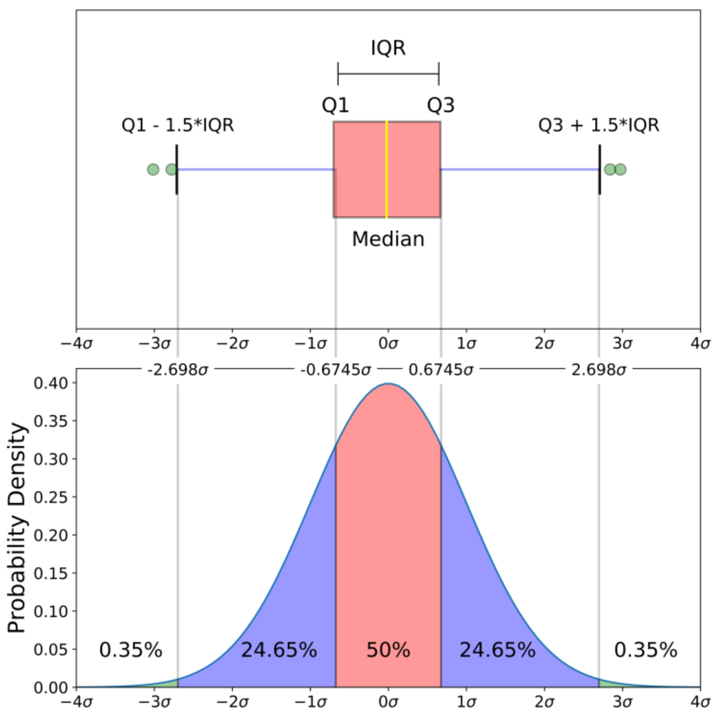
interquartile range (IQR): 25th to the 75th percentile.

outliers (shown as green circles)

Whiskers can stand for several other things.

- **"maximum"**: Q3 + 1.5* IQR and **"minimum"**: Q1 -1.5* IQR (example right)
- the lowest datum still within 1.5 IQR of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile (often called the **Tukey boxplot**)
- the minimum and maximum of all of the data
- one standard deviation above and below the mean of the data
- the 9th percentile and the 91st percentile
- the 2nd percentile and the 98th percentile.

Question: Why IQR and not standard deviation?

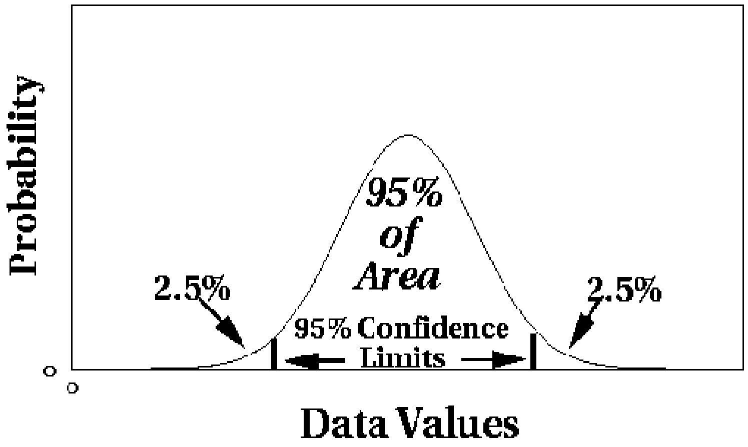


STATISTICAL APPROACHES---MODEL-BASED

Assume a parametric model describing the distribution of the data (e.g., normal distribution)

Apply a statistical test that depends on

- Data distribution
- Parameter of distribution (e.g., mean, variance)
- Number of expected outliers (confidence limit)



GRUBBS' TEST

Detect outliers in univariate data

Assume data comes from normal distribution

Detects one outlier at a time, remove the outlier, and repeat

- H_0: There is no outlier in data
- H_A: There is at least one outlier

Grubbs' test statistic:

Reject H_0 if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N, N-2)}}{N-2+t^2_{(\alpha/N, N-2)}}}$$

$$G = \frac{\max \left| X - \overline{X} \right|}{s}$$

STATISTICAL-BASED – LIKELIHOOD APPROACH

Assume the data set D contains samples from a mixture of two probability distributions:

- M (majority distribution)
- A (anomalous distribution)

General Approach:

- Initially, assume all the data points belong to M
- Let $L_t(D)$ be the log likelihood of D at time t
- For each point x_t that belongs to M, move it to A
  - Let L_{t+1} (D) be the new log likelihood.
  - Compute the difference, $\Delta = L_t(D) - L_{t+1}(D)$
  - If $\Delta > c$ (some threshold), then x_t is declared as an anomaly and moved permanently from M to A

LIMITATIONS OF STATISTICAL APPROACHES

Most of the tests are for a single attribute

In many cases, data distribution/model may not be known

For high dimensional data, it may be difficult to estimate the true distribution (possible approach: use EM-algorithms to fit Gaussian Mixture Model)

DISTANCE-BASED APPROACHES

Data is represented as a vector of features

Three major approaches

- Nearest-neighbor based
- Density based
- Clustering based

NEAREST-NEIGHBOR BASED APPROACH

Approach:

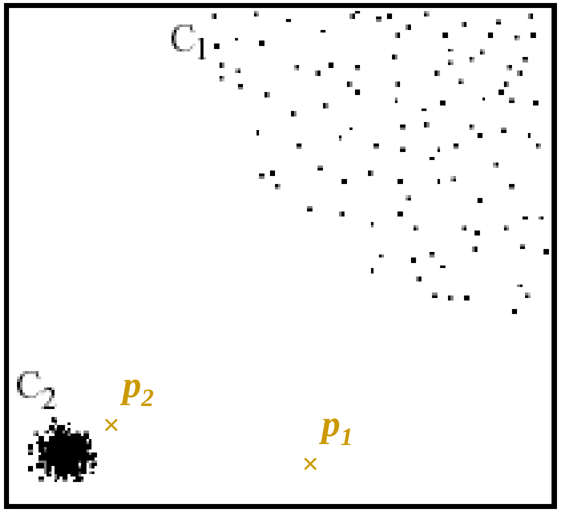
- Compute the distance between every pair of data points
- There are various ways to define outliers:
  - Data points for which there are fewer than *p* neighboring points within a distance *D*
  - The top n data points whose distance to the kth nearest neighbor is greatest
  - The top n data points whose average distance to the k nearest neighbors is greatest

DENSITY-BASED: LOF APPROACH

For each point, compute the density of its local neighborhood; e.g. use DBSCAN's approach

Compute local outlier factor (LOF) of a sample *p* as the average of the ratios of the density of sample *p* and the density of its nearest neighbors

Outliers are points with largest LOF value



In the NN approach, p_2 is not considered as outlier, while LOF approach find both p_1 and p_2 as outliers

Alternative approach: directly use density function; e.g. DENCLUE's density function

CLUSTERING-BASED

Idea: Use a clustering algorithm that has some notion of outliers!

Problem what parameters should I choose for the algorithm; e.g. DBSCAN?

Rule of Thumb: Less than x% of the data should be outliers (with x typically chosen between 0.1 and 10); x might be determined with other methods; e.g. statistical tests.

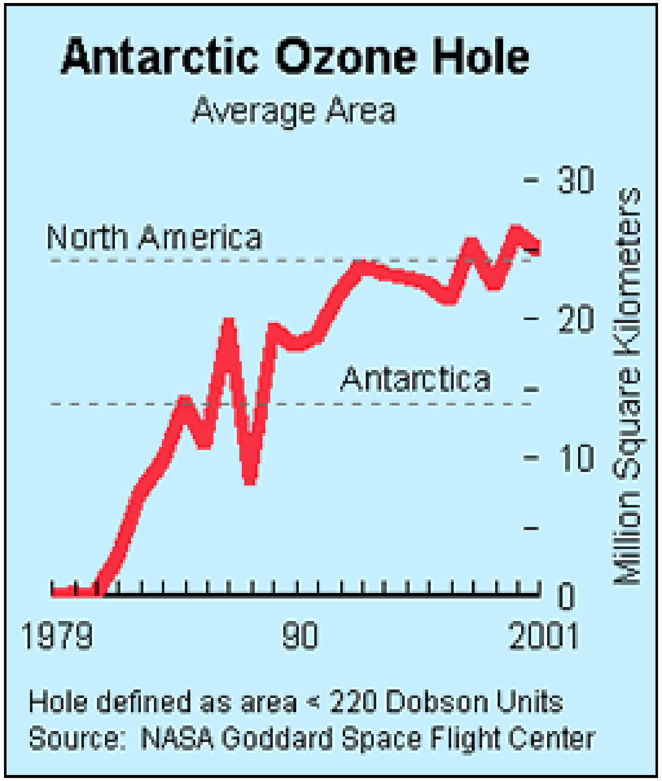
REMOVING OUTLIERS CAN BE DANGEROUS

Ozone Depletion History

In 1985 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal levels

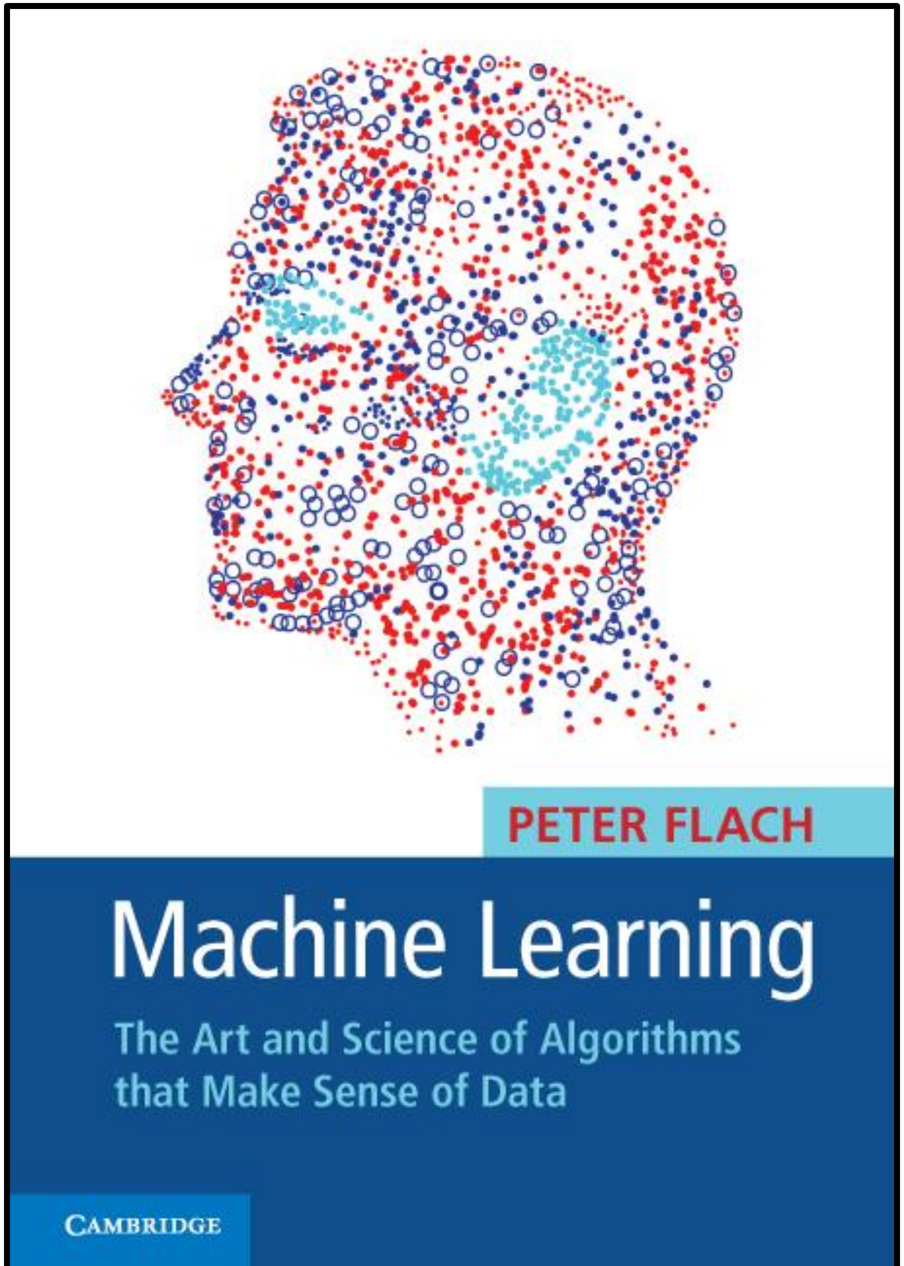
Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations?

The ozone concentrations recorded by the satellite were so low they were being treated as outliers by a computer program and discarded!



Sources:
http://exploringdata.cqu.edu.au/ozone.html
http://www.epa.gov/ozone/science/hole/size.html

A PRACTICAL GUIDE TO MACHINE LEARNING


PETER FLACH
Machine Learning
The Art and Science of Algorithms
that Make Sense of Data
CAMBRIDGE

MACHINE LEARNING PROBLEMS

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

MACHINE LEARNING PROBLEMS

(Boosted-) Decision Trees

K-Means
Agglomerative clustering
DBScan

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

(Boosted-) Decision Trees

PCA

MACHINE LEARNING PROBLEMS

(Boosted-) Decision Trees

K-Means
Agglomerative clustering
DBScan

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

(Boosted-) Decision Trees

PCA

CLUSTERING STRATEGIES

K-means

- Iteratively re-assign points to the nearest cluster center

Agglomerative clustering

- Start with each point as its own cluster and iteratively merge the closest clusters

Mean-shift clustering

- Estimate modes of PDF (i.e., the value x at which its probability mass function takes its maximum value)

Spectral clustering

- Split the nodes in a graph based on assigned links with similarity weights

DBSCAN (Density-based spatial clustering of applications with noise)

As we go down this chart, the clustering strategies have more tendency to transitively group points even if they are not nearby in feature space

CLUSTERING STRATEGIES

K-means

- Iteratively re-assign points to the nearest cluster center

Agglomerative clustering

- Start with each point as its own cluster and iteratively merge the closest clusters

Mean-shift clustering

- Estimate modes of PDF (i.e., the value x at which its probability mass function takes its maximum value)

Spectral clustering

- Split the nodes in a graph based on assigned links with similarity weights

DBSCAN (Density-based spatial clustering of applications with noise)

As we go down this chart, the clustering strategies have more tendency to transitively group points even if they are not nearby in feature space

K-MEANS ALGORITHM

Select K random data points $\{s_1, s_2, \ldots s_K\}$ as centroids c_j.

Until clustering converges or other stopping criterion {

  For each data point x_i:

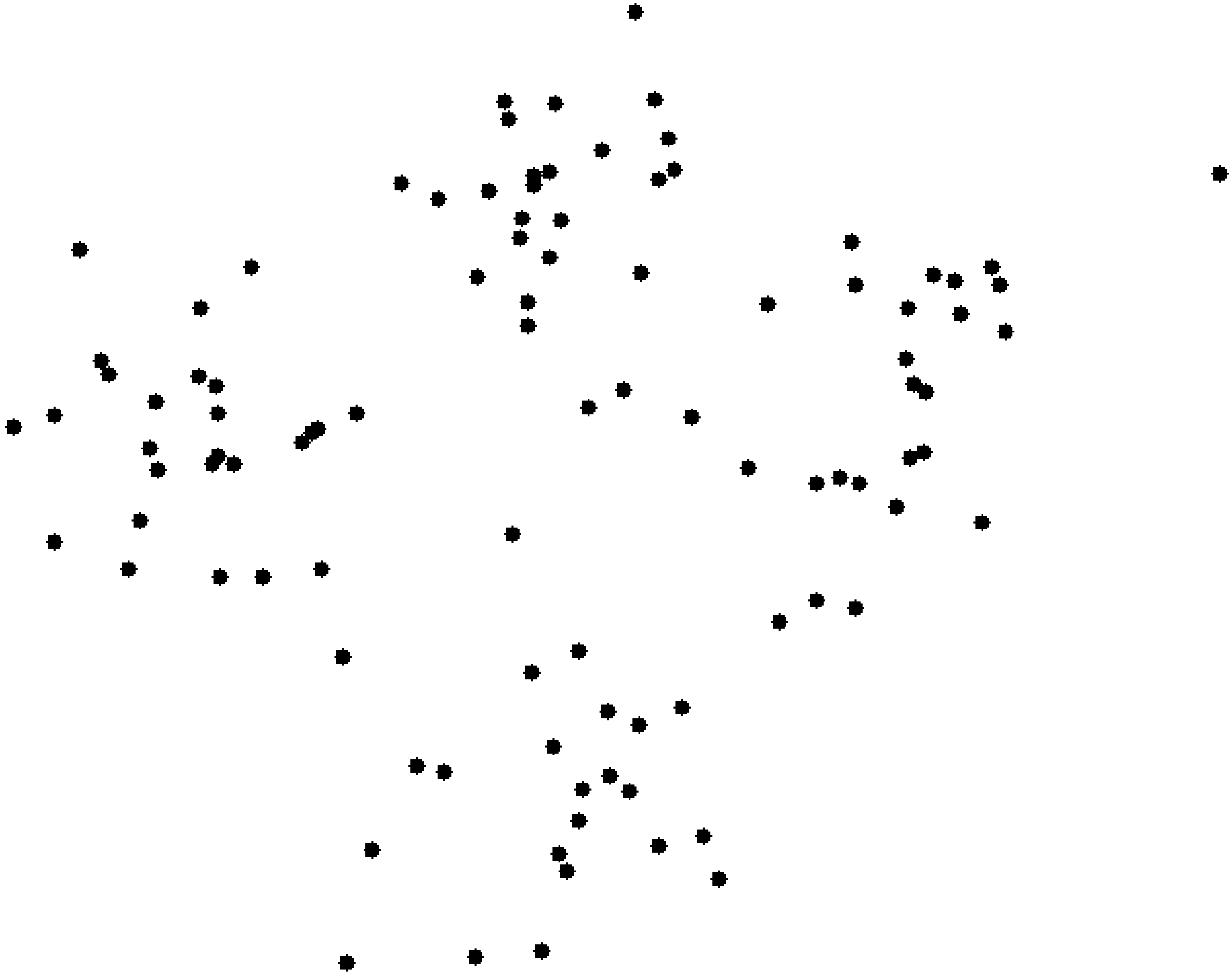
    Assign x_i to the closes centroid such that

      $dist(x_i, c_j)$ is minimal.

    For each cluster c_j, update the centroids

      $c_j = \mu(c_j)$

}

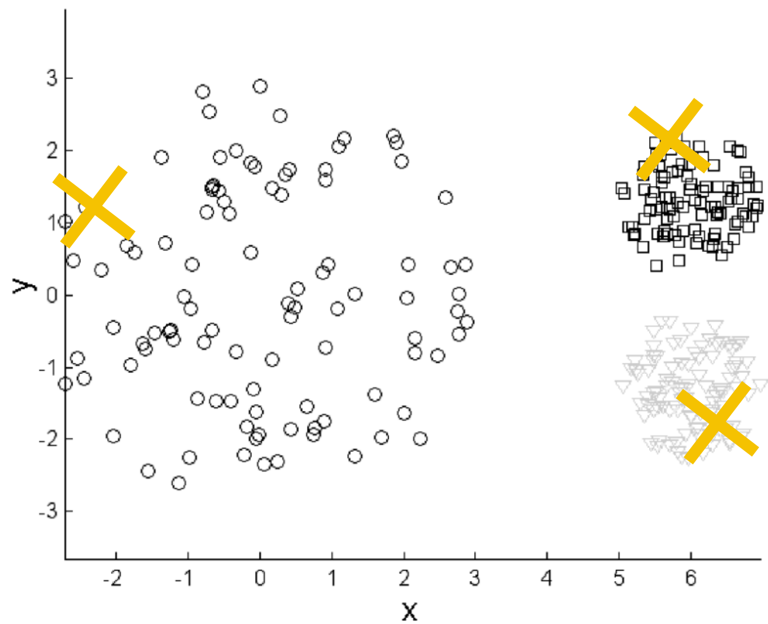


TERMINATION CONDITIONS

Several possibilities, e.g.,

- A fixed number of iterations.
- Partition unchanged.
- Centroid positions don't change.

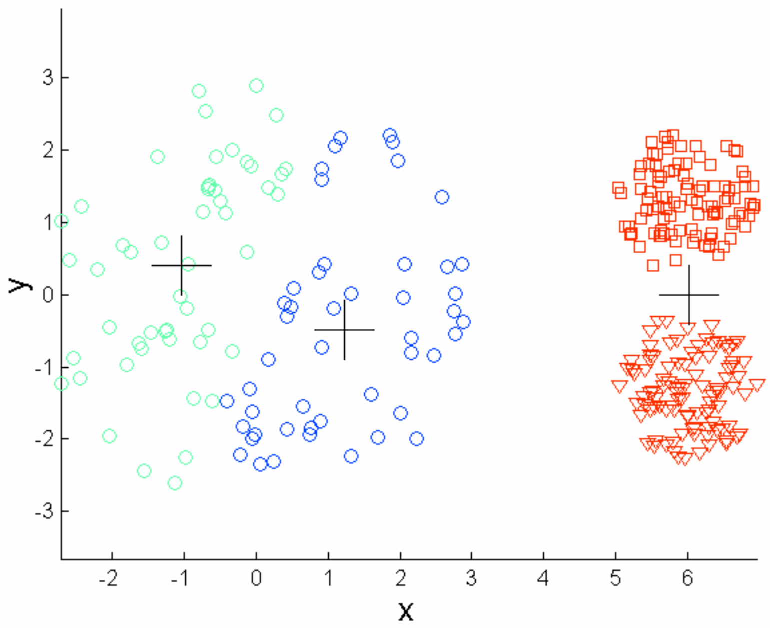
CLASS TASK



Original Points

What cluster will you get with the yellow centroids?

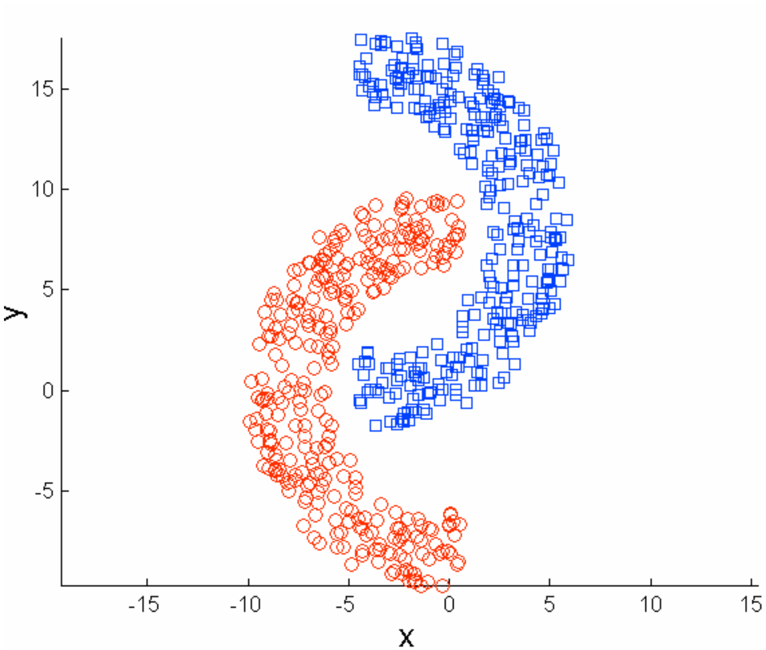
CLASS TASK



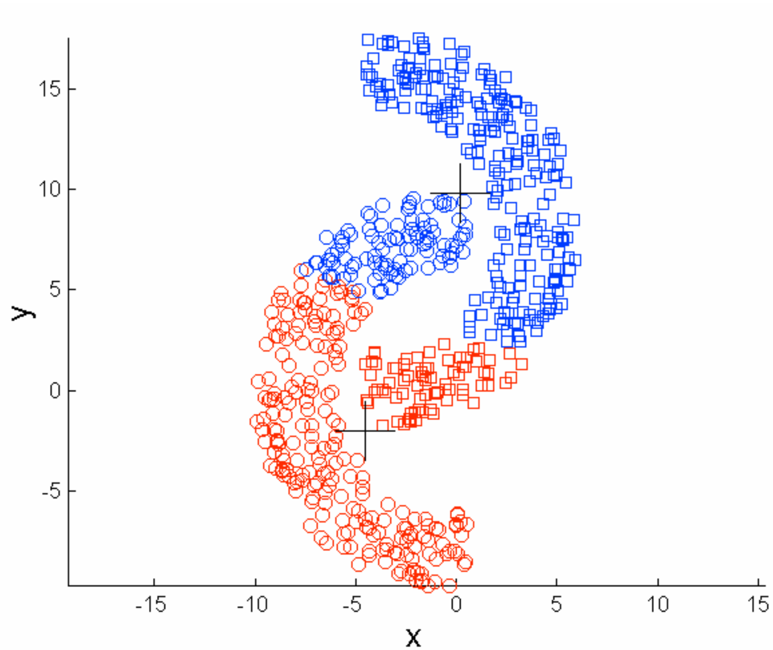
K-means (3 Clusters)

LIMITATIONS OF K-MEANS

Non-globular Shapes

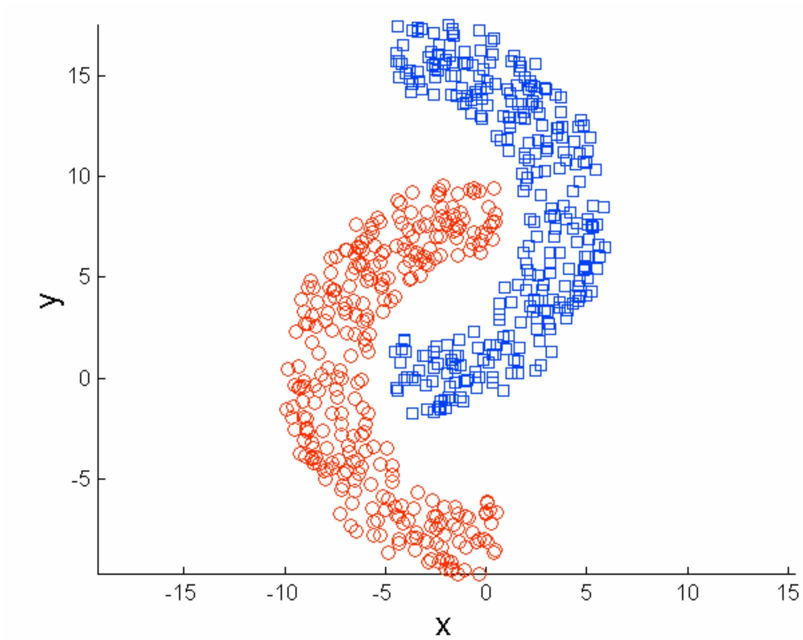


Original Points

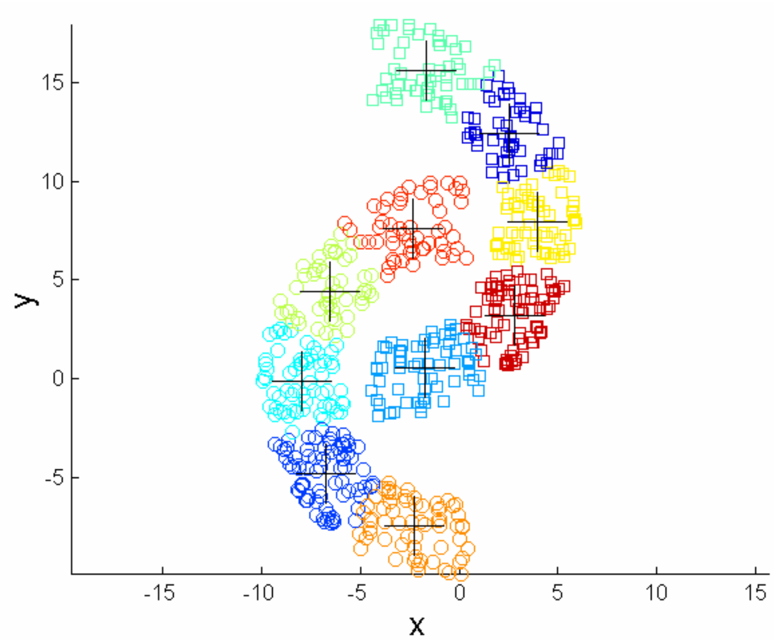


K-means (2 Clusters)

OVERCOMING K-MEANS LIMITATIONS



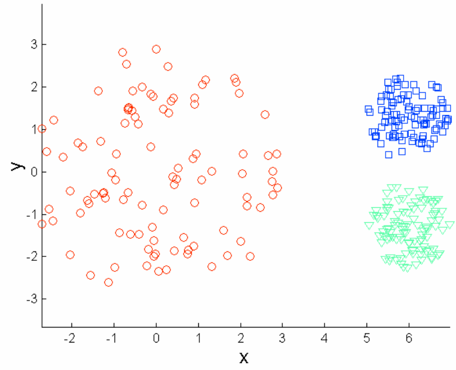
Original Points



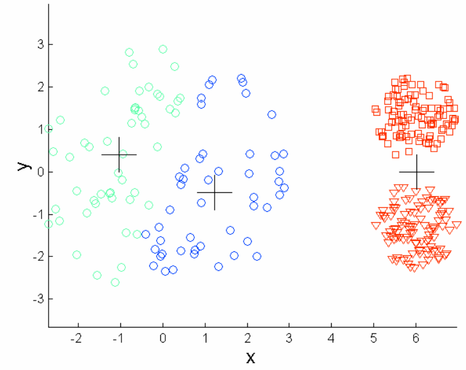
K-means Clusters

Can you think of other ways to overcome the limitations?

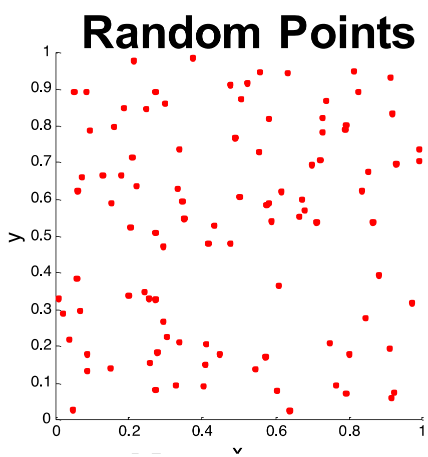
How do I know how good the clustering is?

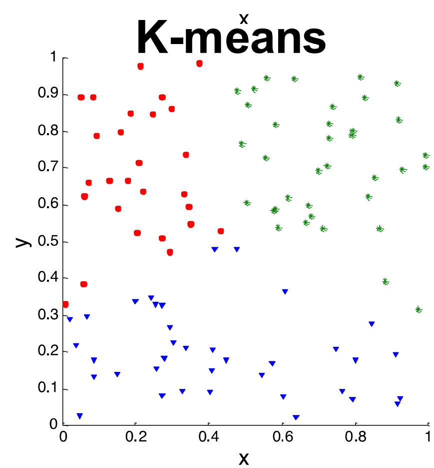


Original Points



K-means (3 Clusters)

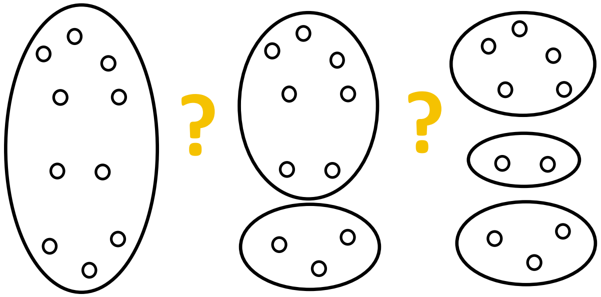




Measuring clustering validity

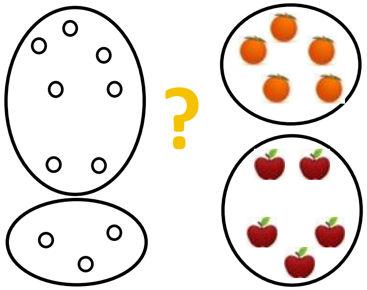
Internal Index:

- Validate *without* external info
- With different number of clusters



External Index

Validate against ground truth



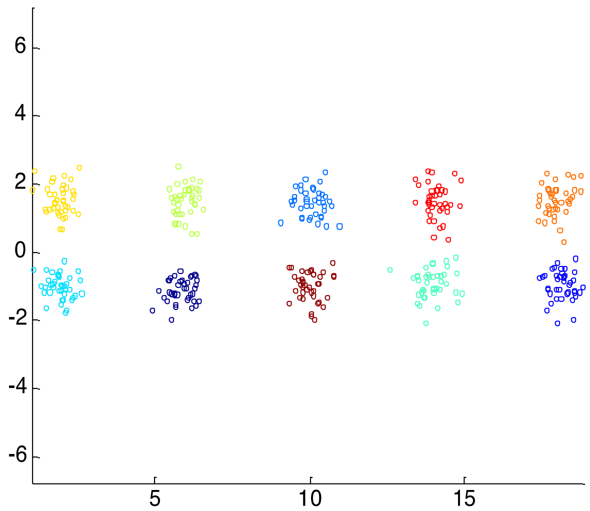
INTERNAL INDEXES

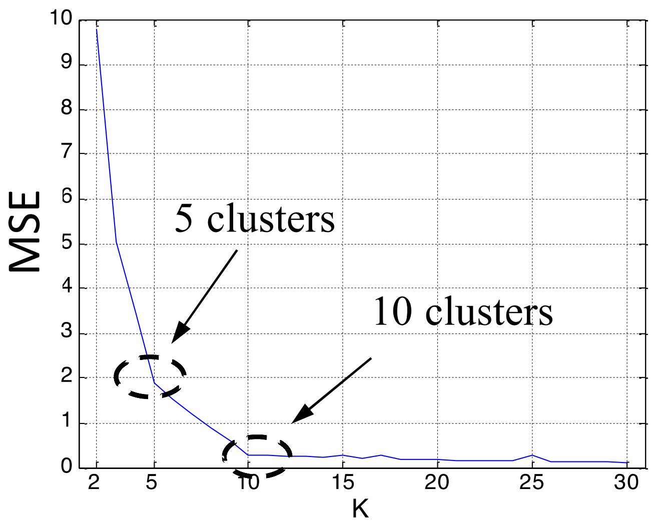
Ground truth is rarely available but unsupervised validation must be done.

Minimizes (or maximizes) internal index:

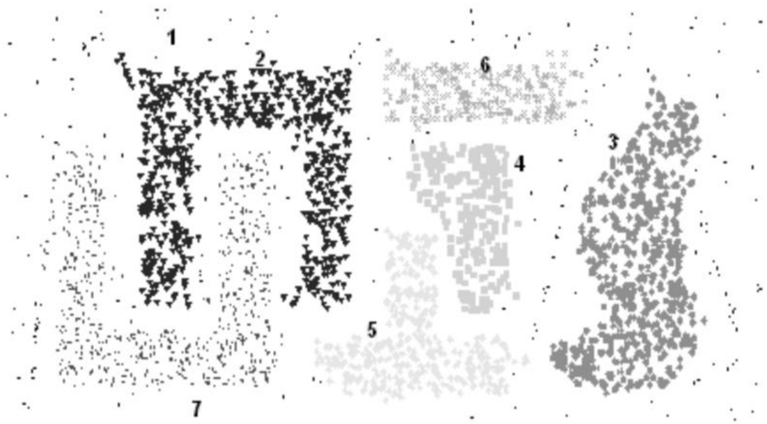
- Variances of within cluster and between clusters
- Rate-distortion method
- F-ratio
- Davies-Bouldin index (DBI)
- Bayesian Information Criterion (BIC)
- Silhouette Coefficient
- Minimum description principle (MDL)
- Stochastic complexity (SC)

MEAN SQUARE ERROR (MSE)

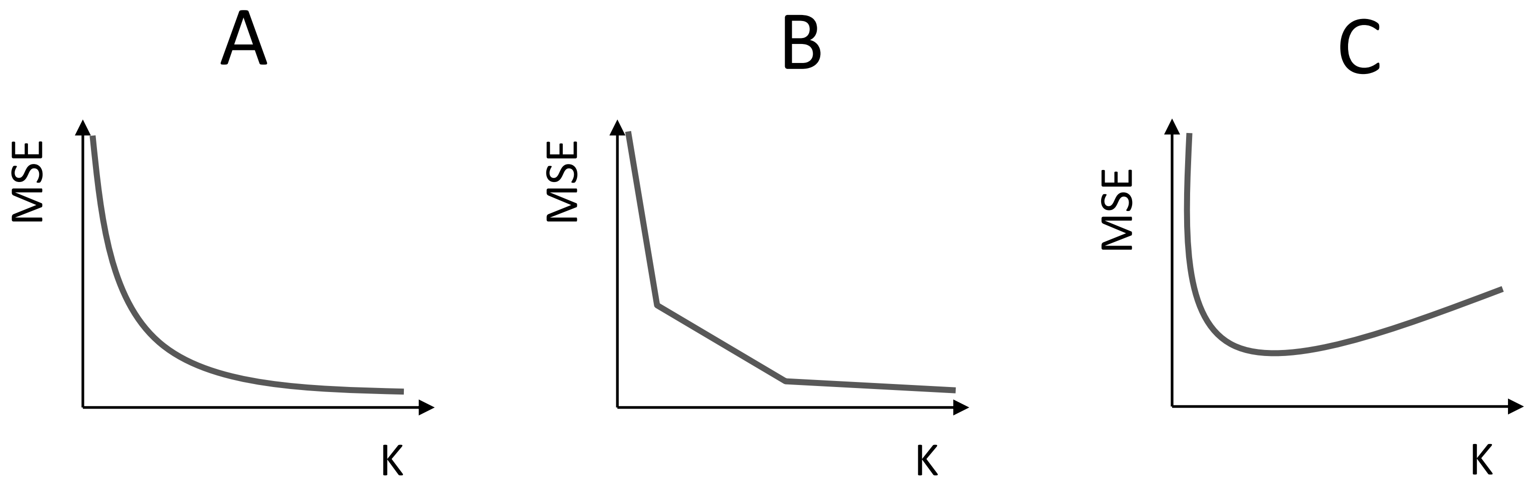




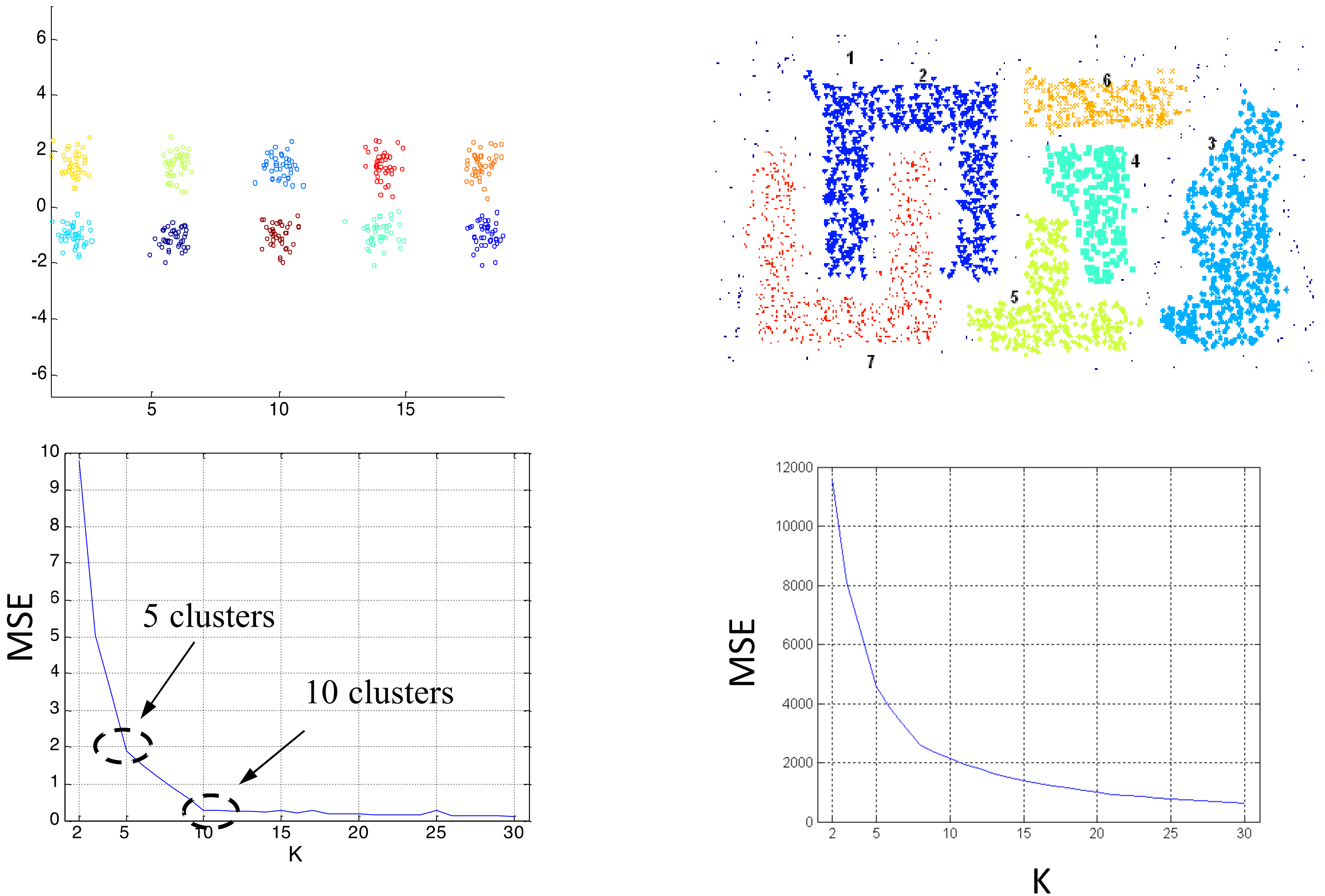
CLICKER - HTTPS://CLICKER.CSAIL.MIT.EDU/6.S079/



What MSE curve to you expect for this data?

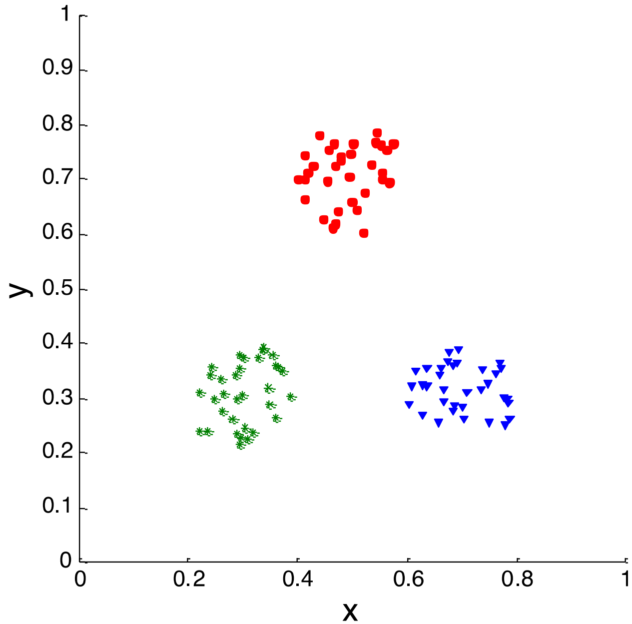


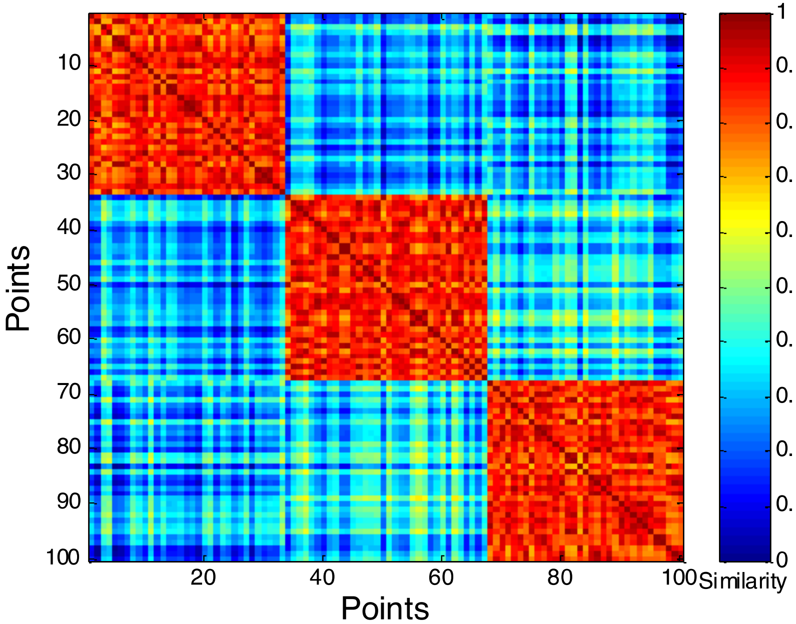
MEAN SQUARE ERROR (MSE)



USING SIMILARITY MATRIX FOR CLUSTER VALIDATION

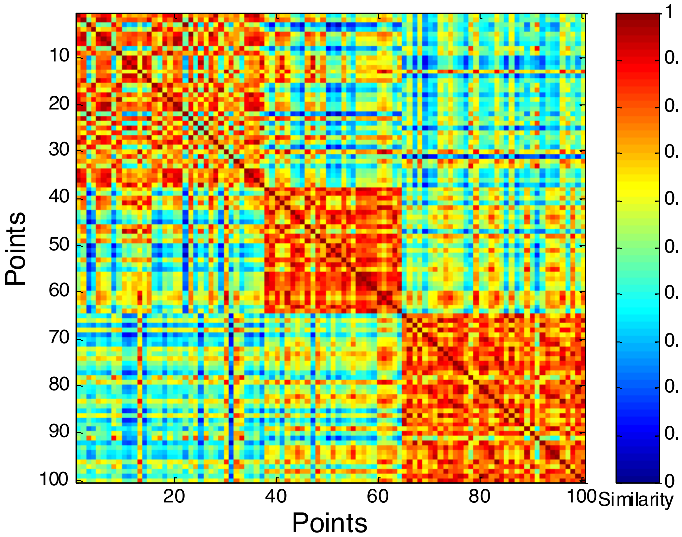
Order the similarity matrix with respect to cluster labels and inspect visually.

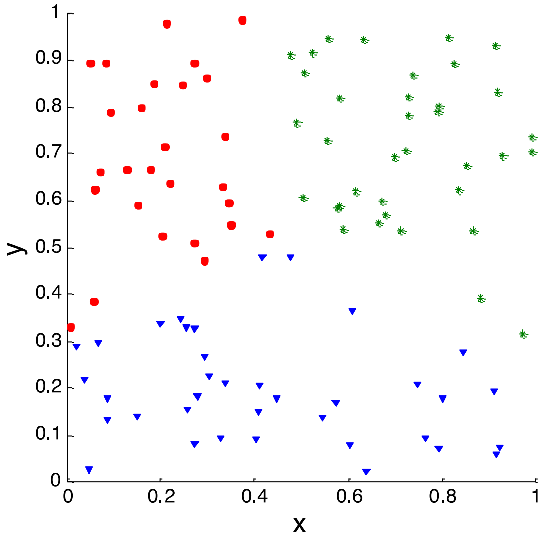




USING SIMILARITY MATRIX FOR CLUSTER VALIDATION

Clusters in random data are not so crisp





K-means

CLUSTERING STRATEGIES

K-means

- Iteratively re-assign points to the nearest cluster center

Agglomerative clustering

- Start with each point as its own cluster and iteratively merge the closest clusters

Mean-shift clustering

- Estimate modes of PDF (i.e., the value x at which its probability mass function takes its maximum value)

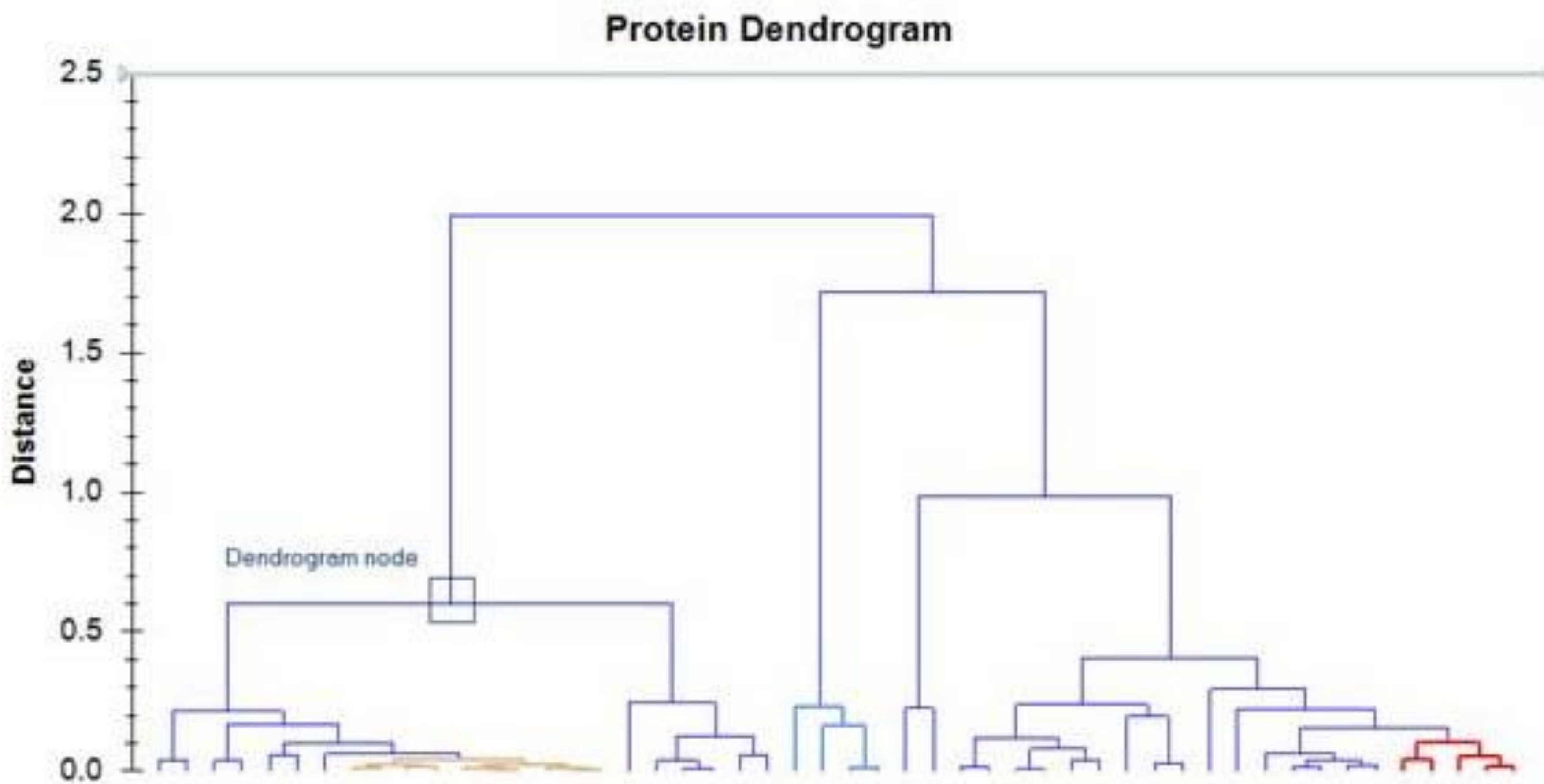
Spectral clustering

- Split the nodes in a graph based on assigned links with similarity weights

DBSCAN (Density-based spatial clustering of applications with noise)

As we go down this chart, the clustering strategies have more tendency to transitively group points even if they are not nearby in feature space

DENDROGRAM EXAMPLE



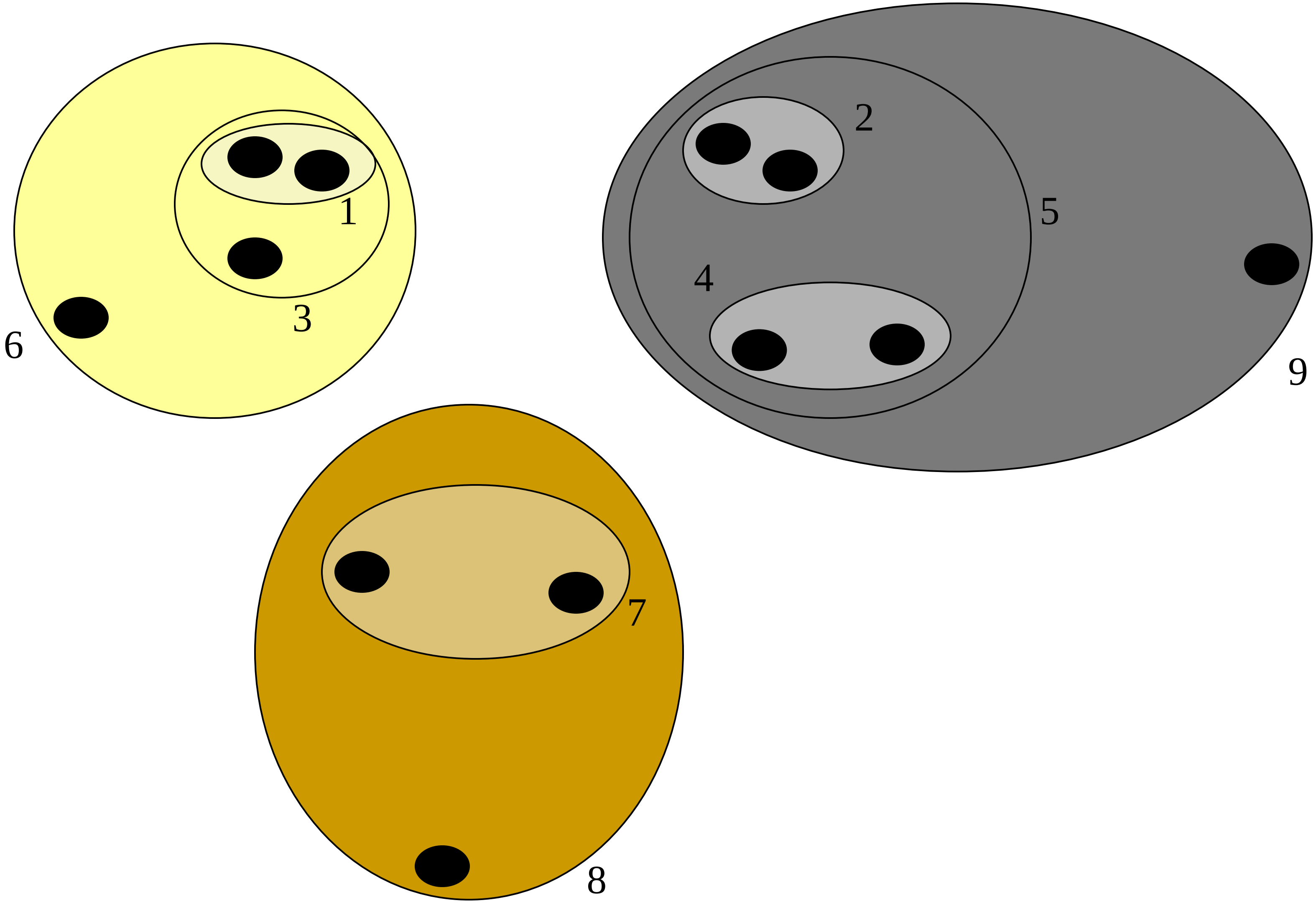
HIERARCHICAL AGGLOMERATIVE CLUSTERING METHODS

Generic Agglomerative Procedure (Salton '89):

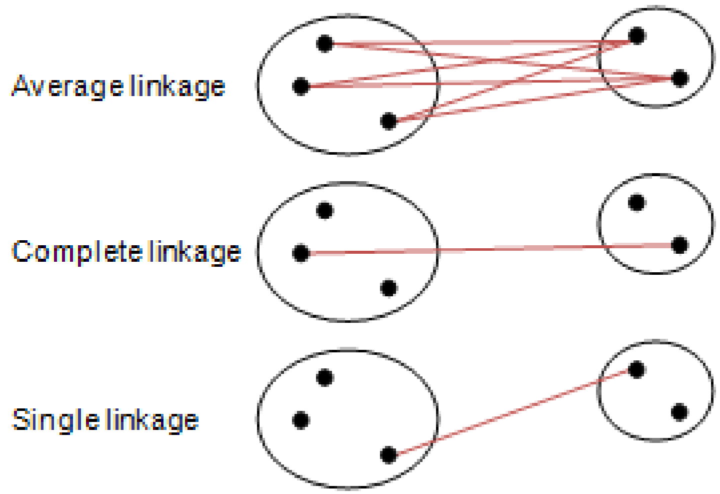
- result in nested clusters via iterations

1. Compute all pairwise document-document similarity coefficients
2. Place each of *n* documents into a class of its own
3. Merge the two most similar clusters into one;
   - replace the two clusters by the new cluster
   - recompute intercluster similarity scores w.r.t. the new cluster
4. Repeat the above step until there are only k clusters left (note k could = 1).

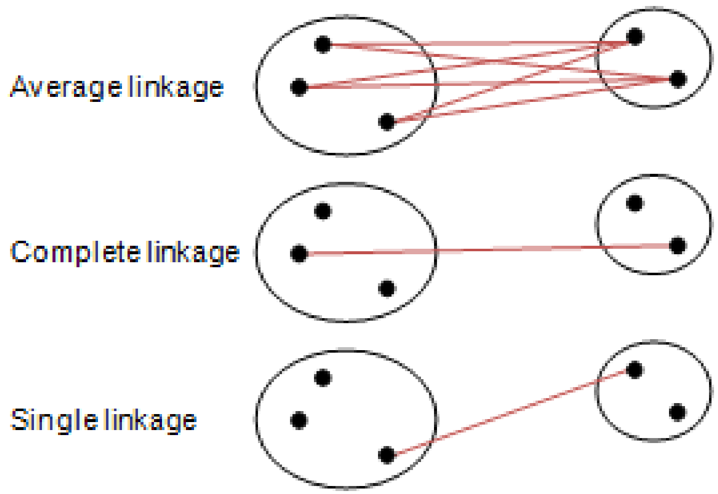
Group Agglomerative Clustering

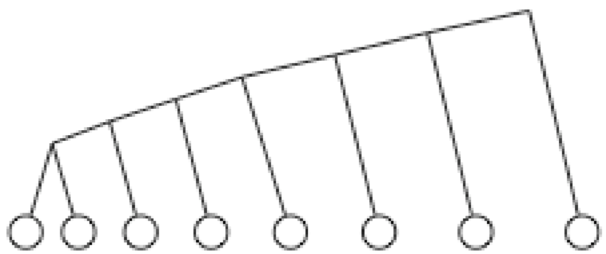


LINKAGE TYPES



CLICKER - HTTPS://CLICKER.CSAIL.MIT.EDU/6.S079/

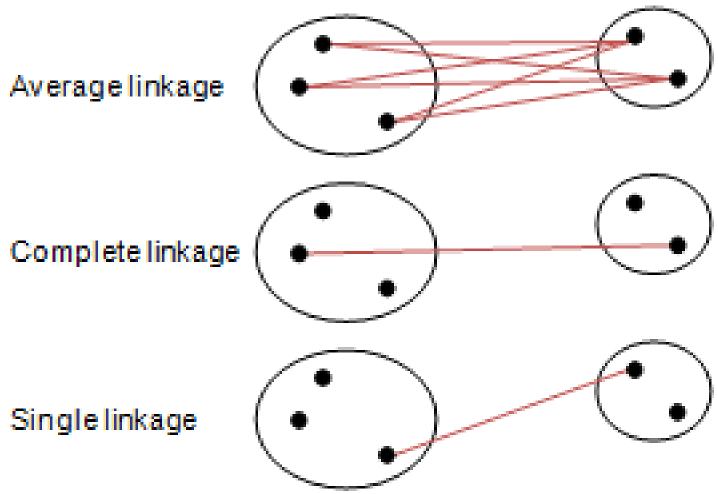


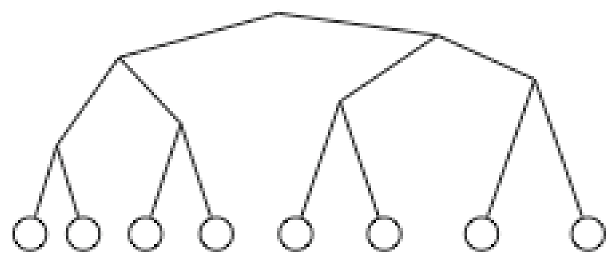


Which linkage type was used for this clustering?

a) Average
b) Complete
c) Single

CLICKER - HTTPS://CLICKER.CSAIL.MIT.EDU/6.S079/





Which linkage type was used for this clustering?

a) Average
b) Complete
c) Single

CLUSTERING STRATEGIES

K-means

- Iteratively re-assign points to the nearest cluster center

Agglomerative clustering

- Start with each point as its own cluster and iteratively merge the closest clusters

Mean-shift clustering

- Estimate modes of PDF (i.e., the value x at which its probability mass function takes its maximum value)

Spectral clustering

- Split the nodes in a graph based on assigned links with similarity weights

DBSCAN (Density-based spatial clustering of applications with noise)

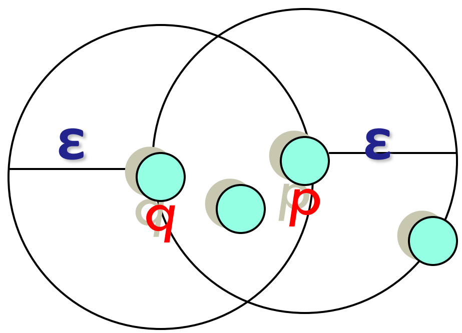
As we go down this chart, the clustering strategies have more tendency to transitively group points even if they are not nearby in feature space

ε-NEIGHBORHOOD

ε-Neighborhood – Objects within a radius of *ε* from an object.

$$N_{\varepsilon}(p):\{q \mid d(p,q) \le \varepsilon\}$$

"High density" - ε-Neighborhood of an object contains at least *MinPts* of objects.



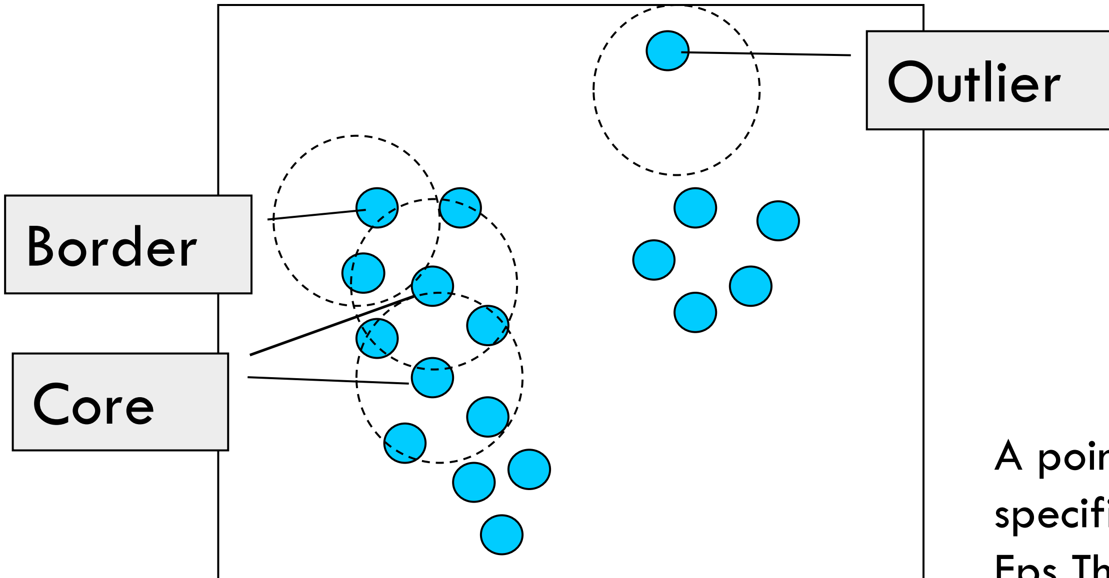
ε-Neighborhood of *p*

ε-Neighborhood of *q*

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 4)

CORE, BORDER & OUTLIER



ε = 1unit, MinPts = 5

Given ε and *MinPts*, categorize the objects into three exclusive groups.

A point is a core point if it has more than a specified number of points (MinPts) within Eps These are points that are at the interior of a cluster.

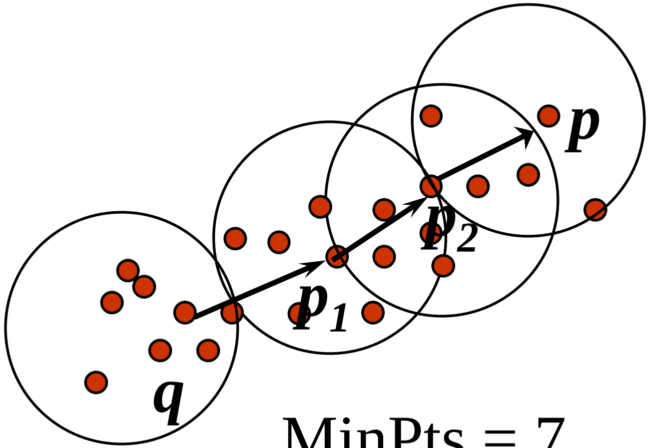
A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point..

A noise point (outlier) is any point that is not a core point nor a border point.

DENSITY-REACHABILITY

Density-Reachable (directly and indirectly):

- A point p is directly density-reachable from p2;
- p2 is directly density-reachable from p1;
- p1 is directly density-reachable from q;
- p←p2←p1←q form a chain.



MinPts = 7

p is (indirectly) density-reachable from q

q is not density- reachable from p?

DBSCAN ALGORITHM

Input: The data set D

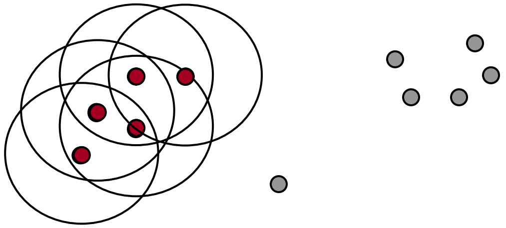
Parameter: ε, MinPts

```
For each object p in D
    if p is a core object and not processed then
        C = retrieve all objects density-reachable from p
        mark all objects in C as processed
        report C as a cluster
    else mark p as outlier
    end if
End For
```

DBSCAN ALGORITHM: EXAMPLE

Parameter

- $\varepsilon = 2$ cm
- $MinPts = 3$

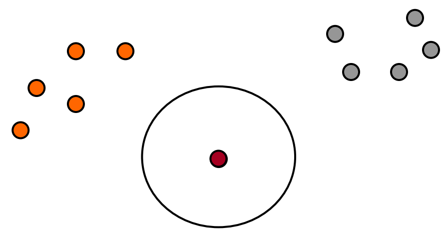


for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

DBSCAN ALGORITHM: EXAMPLE

Parameter

- $\varepsilon = 2$ cm
- $MinPts = 3$

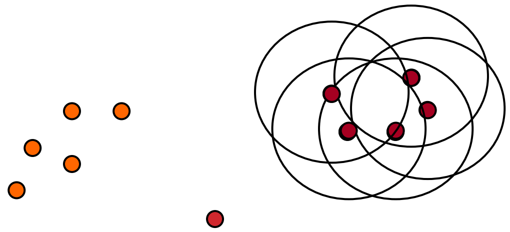


for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

DBSCAN ALGORITHM: EXAMPLE

Parameter

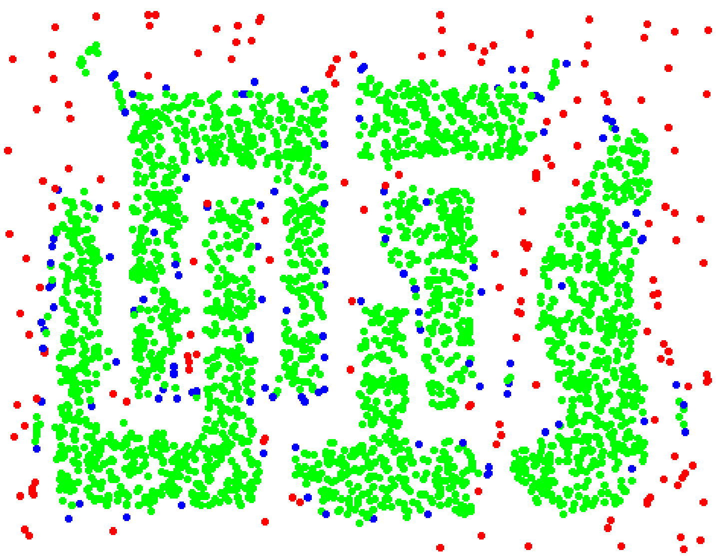
- $\varepsilon = 2$ cm
- $MinPts = 3$



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

EXAMPLE

Original Points

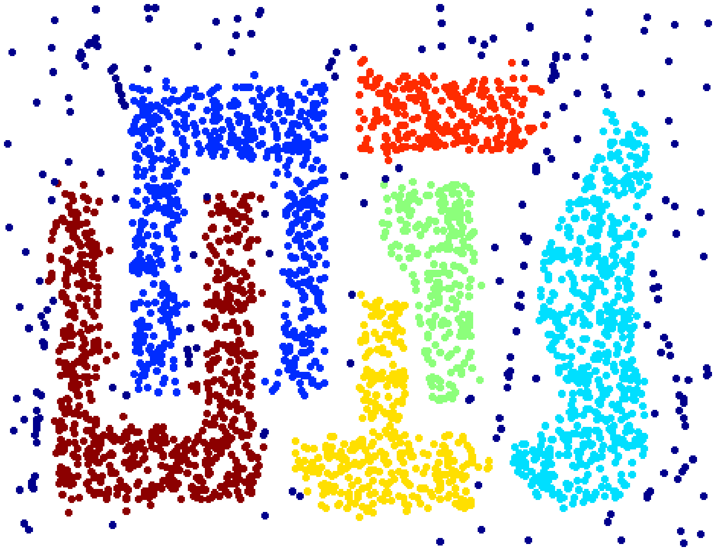


Point types: core, border and outliers

ε = 10, MinPts = 4

WHEN DBSCAN WORKS WELL

Original Points

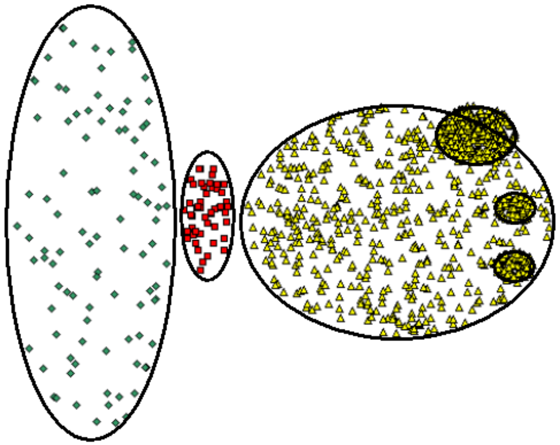


Clusters

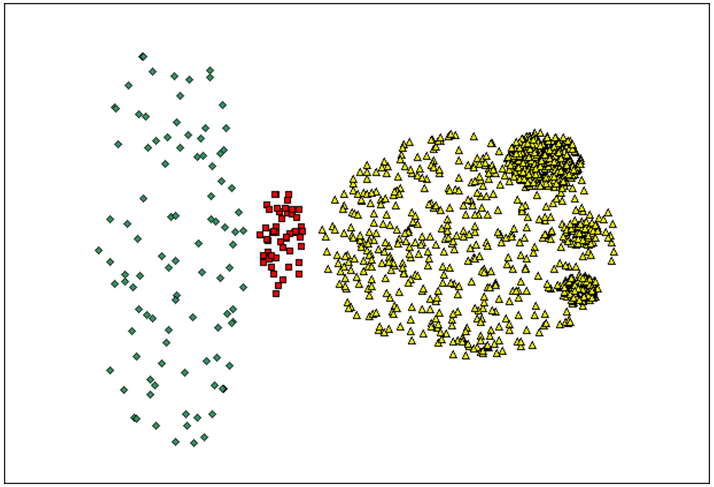
- Resistant to Noise
- Can handle clusters of different shapes and sizes

CAN YOU CREATE AN EXAMPLE FOR WHICH DBSCAN WILL NOT WORK WELL

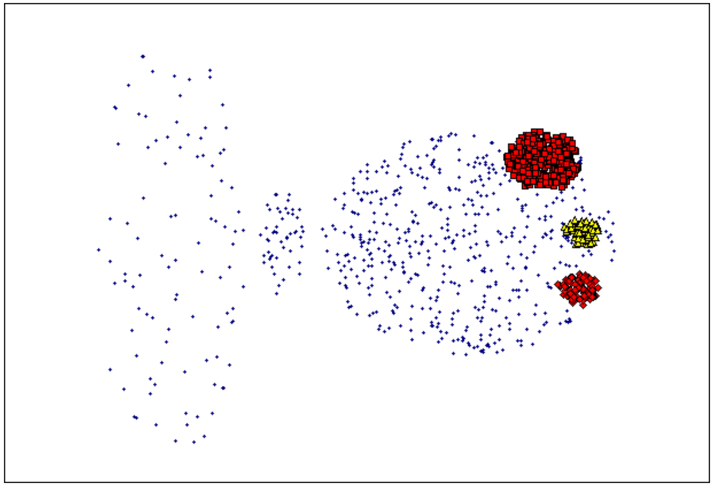
WHEN DBSCAN DOES NOT WORK WELL



Original Points



(MinPts=4, Eps=9.92).

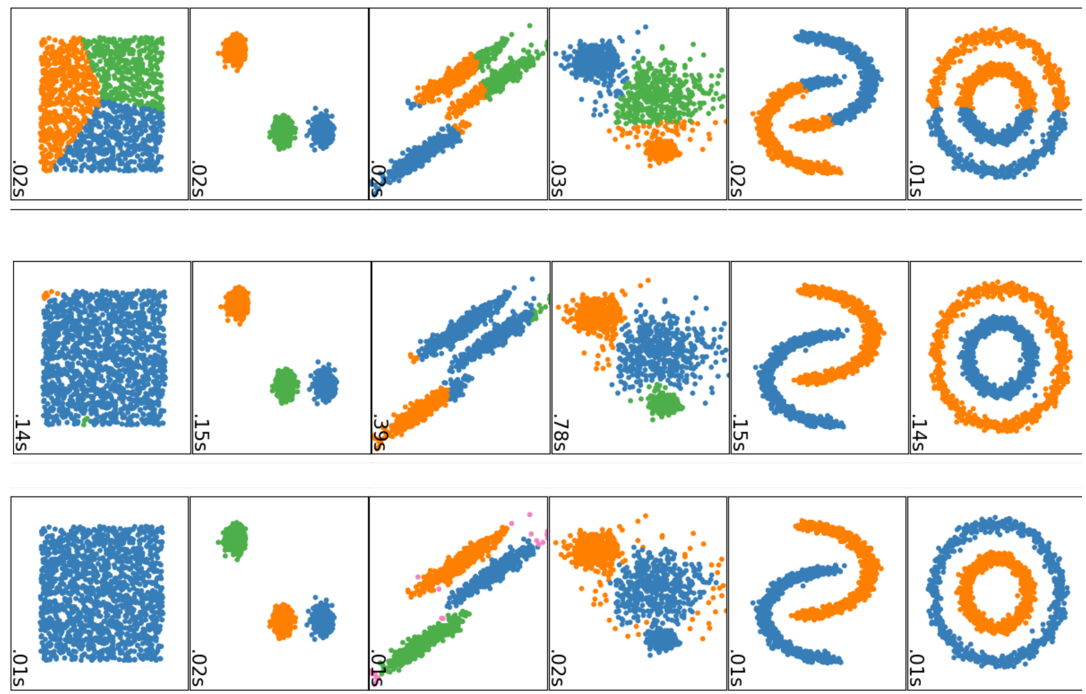


(MinPts=4, Eps=9.75)

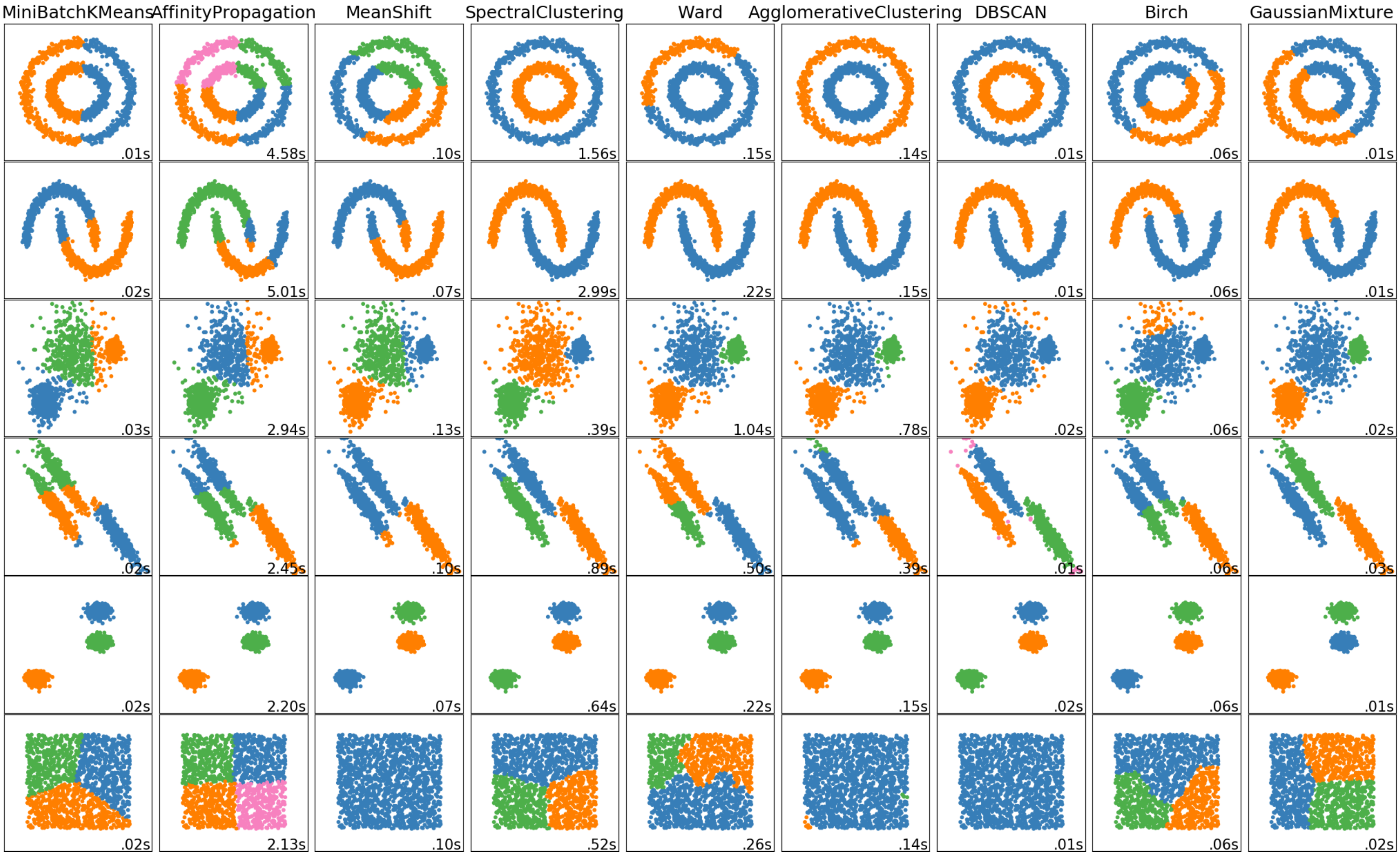
- Cannot handle Varying densities
- Sensitive to parameters

CLICKER - HTTPS://CLICKER.CSAIL.MIT.EDU/6.S079/

A	B	C
KMeans	Aggl. clustering	KMeans
DBScan	KMeans	Aggl. clustering
Aggl. clustering	DBScan	DBScan



CLUSTERING

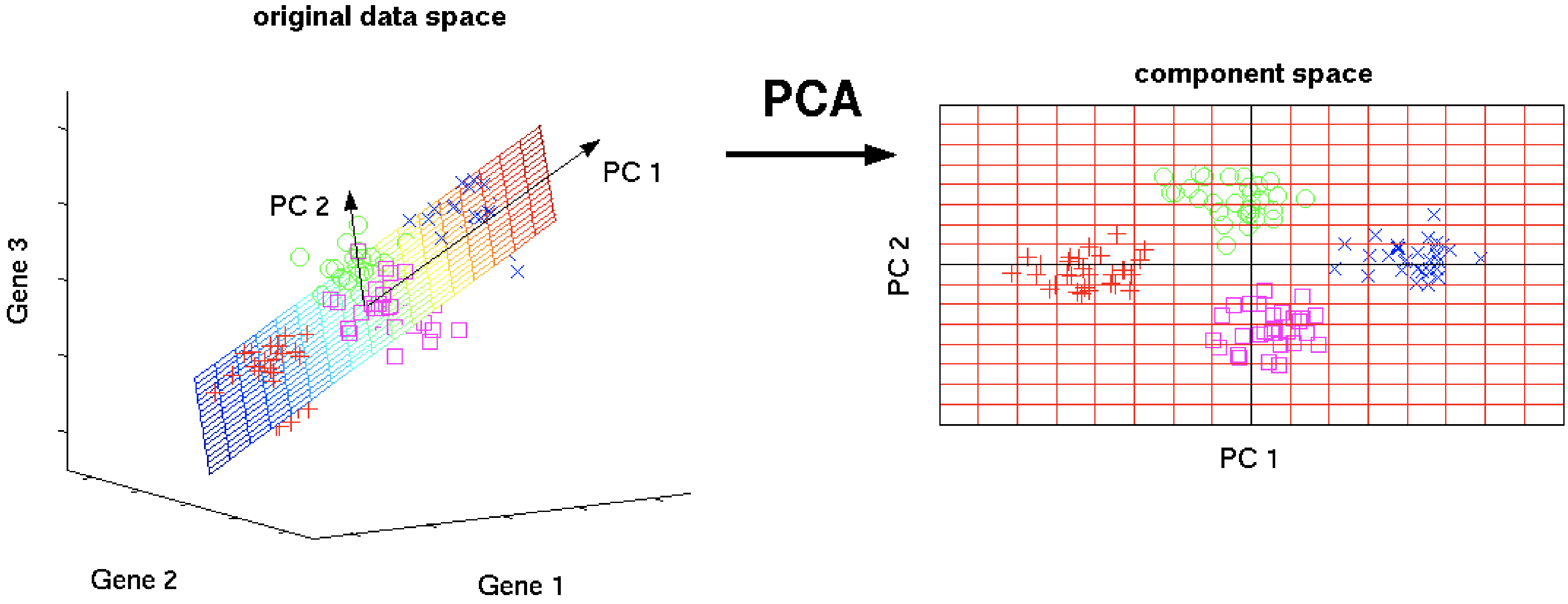


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.01s
4.58s
.10s
1.56s
.15s
.14s
.01s
.06s
.01s
.02s
5.01s
.07s
2.99s
.22s
.15s
.01s
.06s
.01s
.03s
2.94s
.13s
.39s
1.04s
.78s
.02s
.06s
.02s
.02s
2.45s
.10s
.89s
.50s
.39s
.01s
.06s
.03s
.02s
2.20s
.07s
.64s
.22s
.15s
.02s
.06s
.01s
.02s
2.13s
.10s
.52s
.26s
.14s
.01s
.06s
.02s

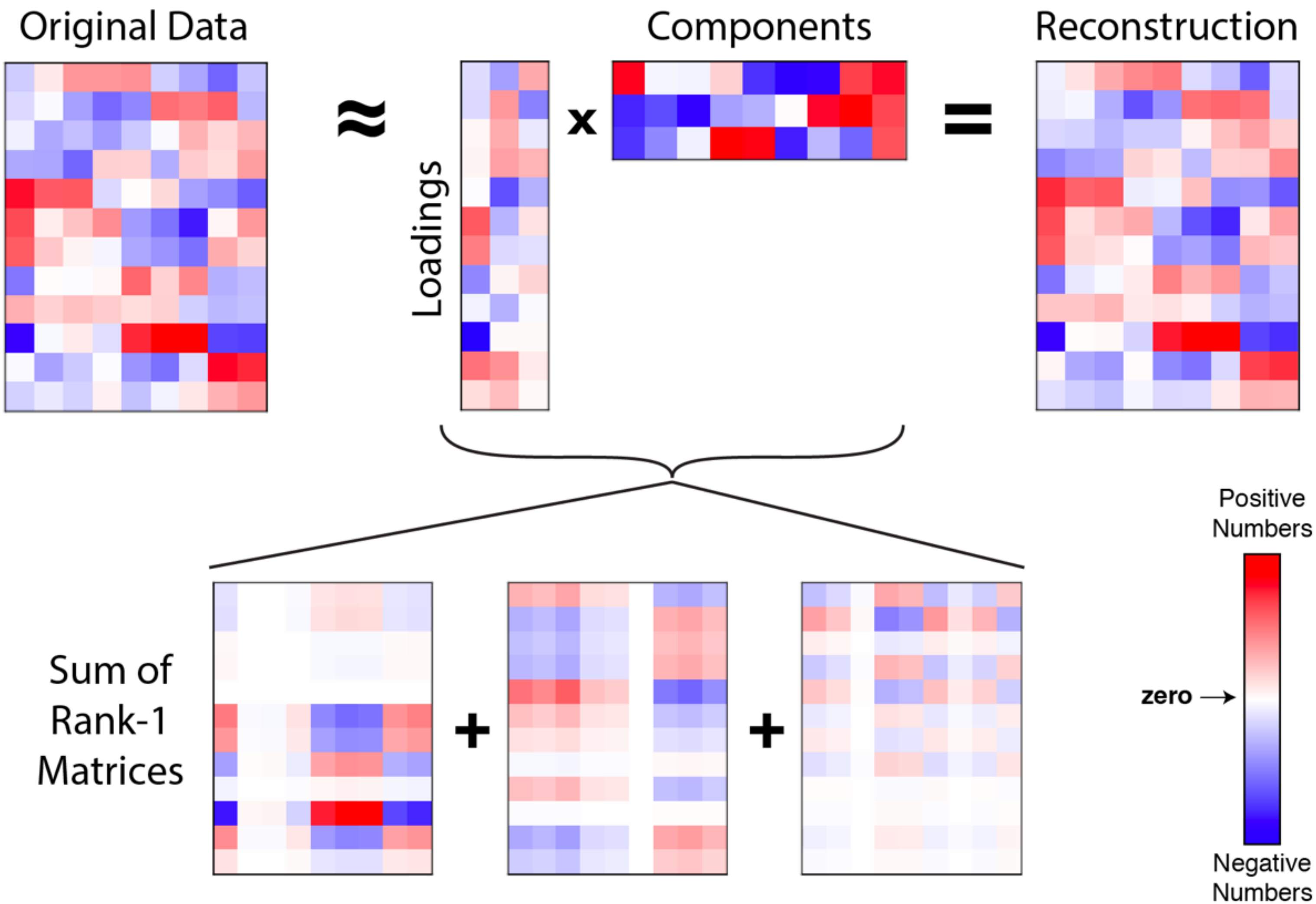
MACHINE LEARNING PROBLEMS

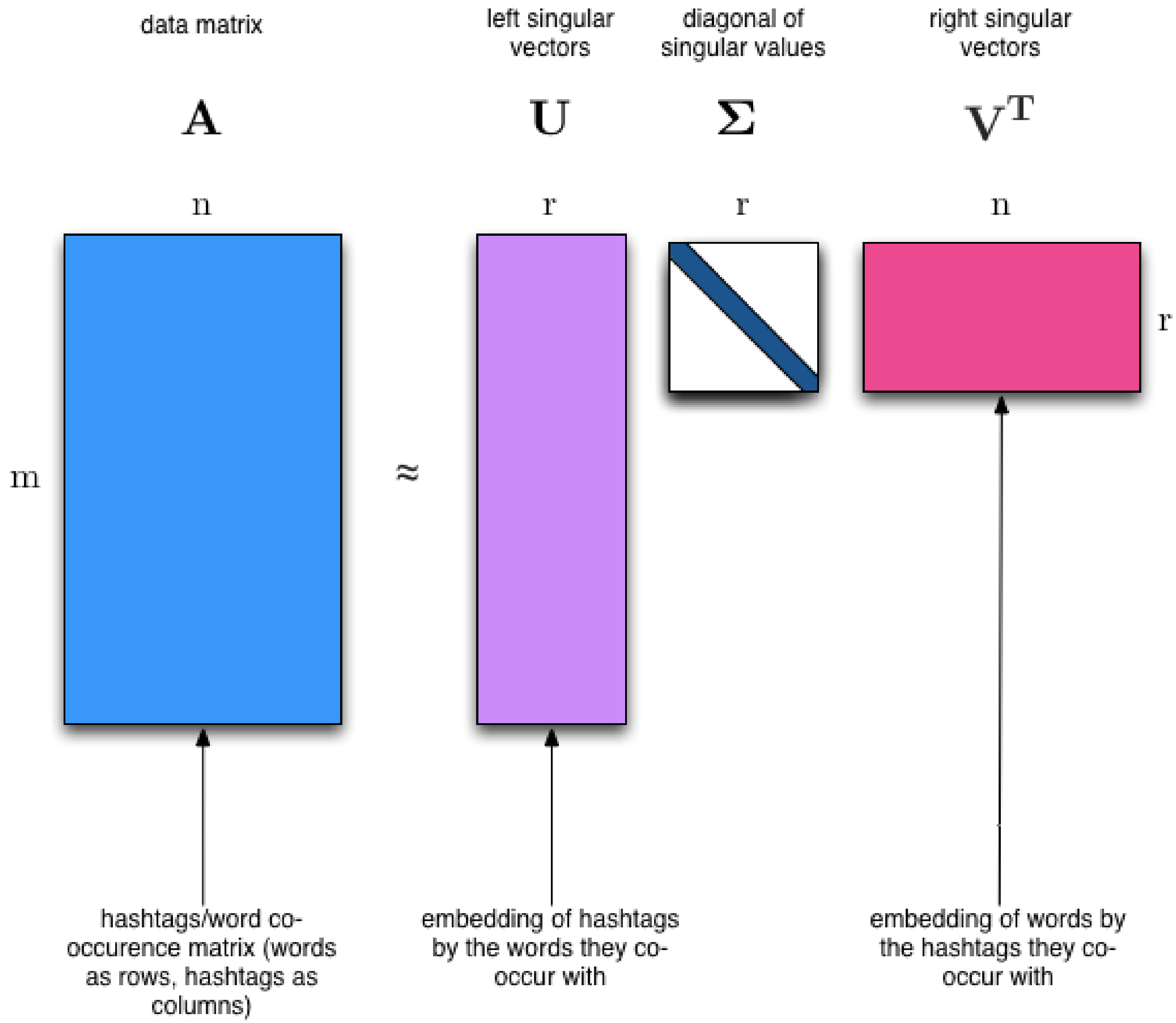
	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	$\boxed{\text{dimensionality reduction}}$

PCA

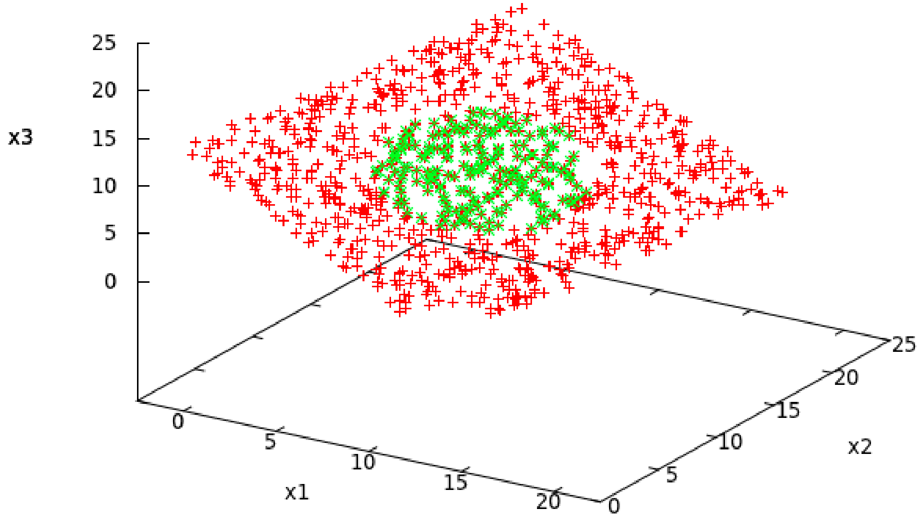

original data space
PC 2
PC 1
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

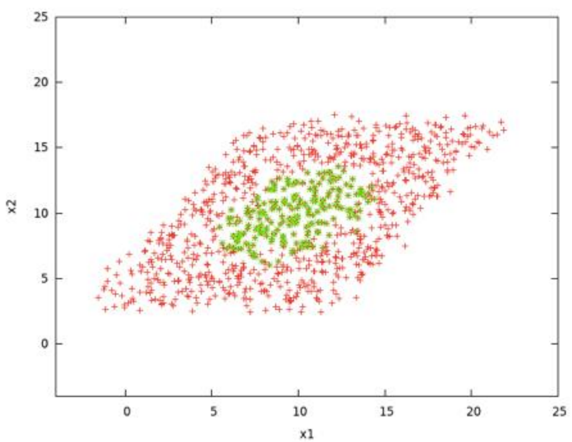
PCA INTUITION

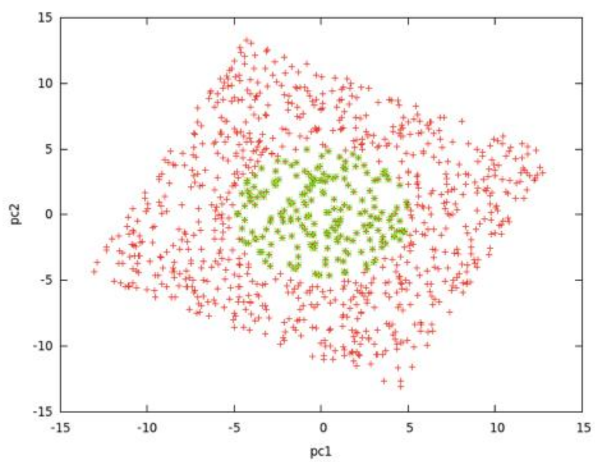



data matrix
left singular vectors
diagonal of singular values
right singular vectors
A
U
Σ
V^T
n
r
r
n
m
≈
r
hashtags/word co-occurence matrix (words as rows, hashtags as columns)
embedding of hashtags by the words they co-occur with
embedding of words by the hashtags they co-occur with

PRINCIPAL COMPONENT ANALYSIS (PCA)



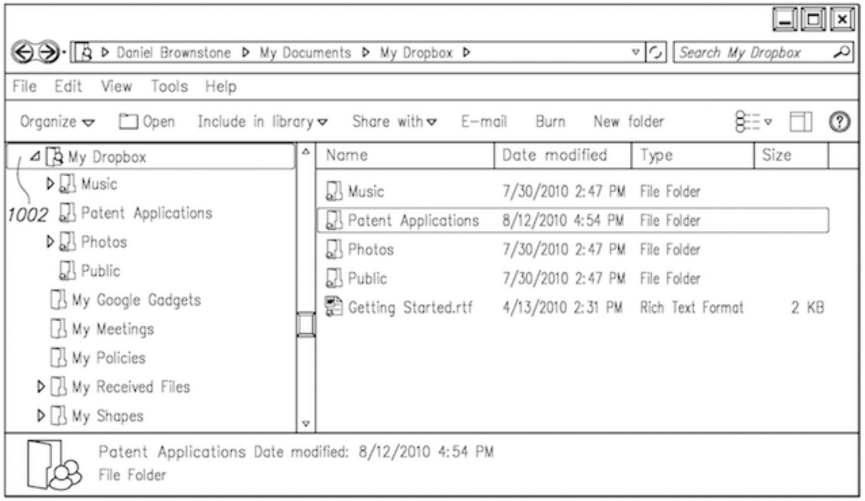




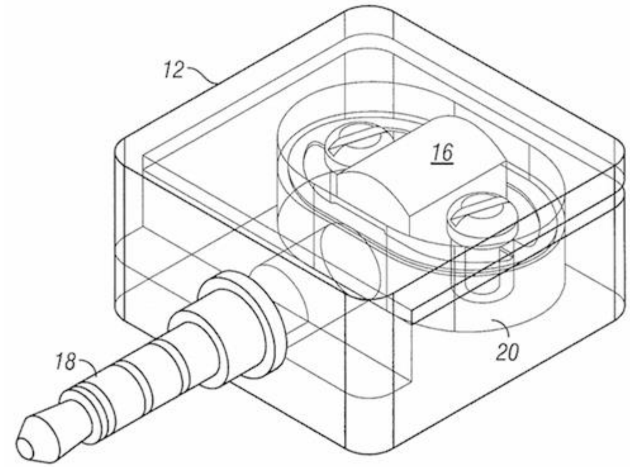
The EM Algorithm

Motivational Example

Around **300.000**
US **Patent** Applications
Granted **per Year**



Network folder synchronization (DropBox)



Systems and methods for decoding card s wipe signals (Square)

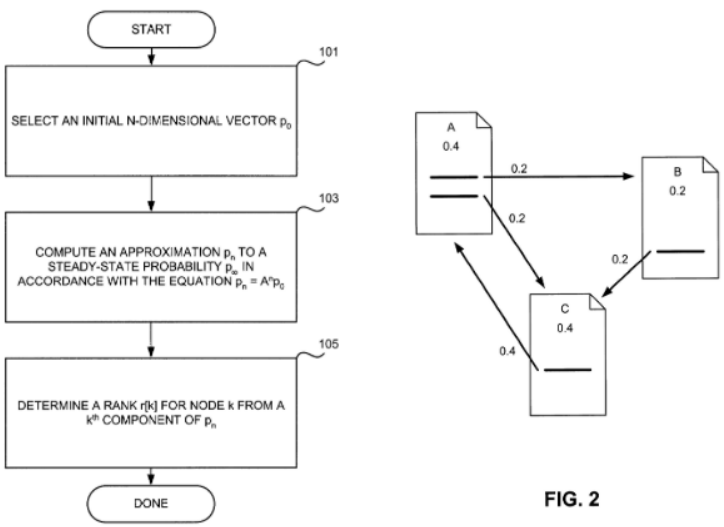


FIG. 2

Method for node ranking in a linked database (Google)

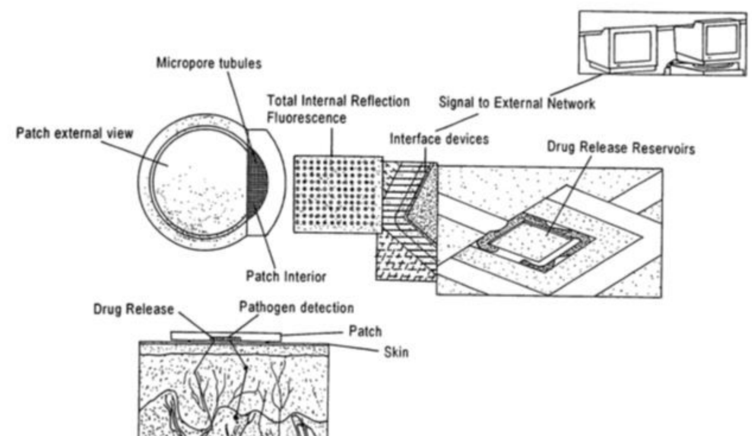
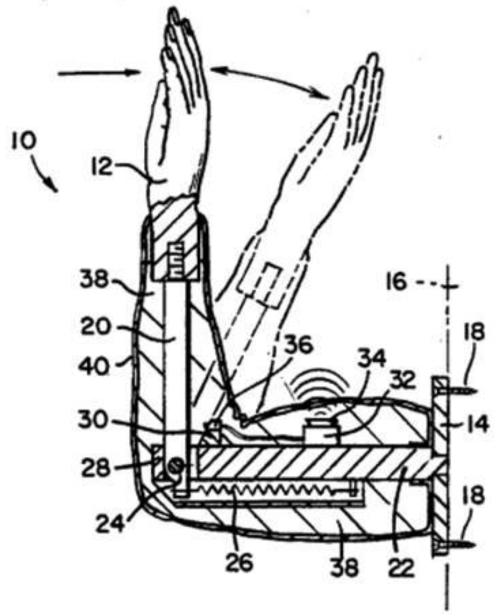
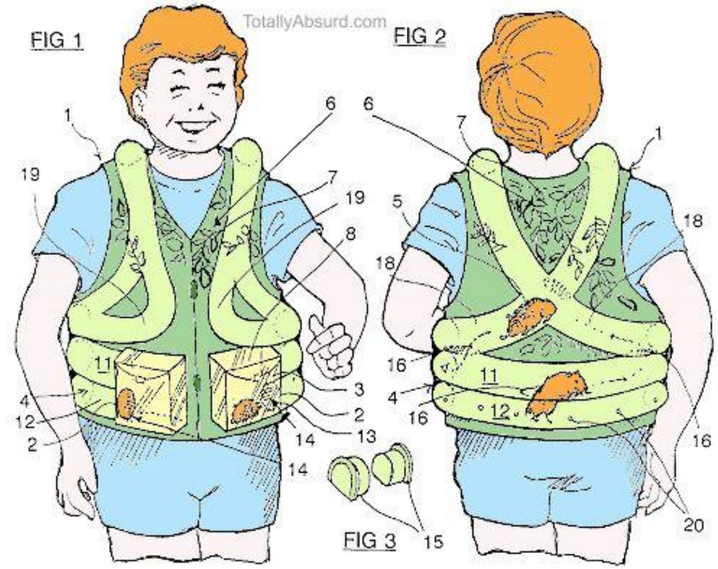


Fig. 2

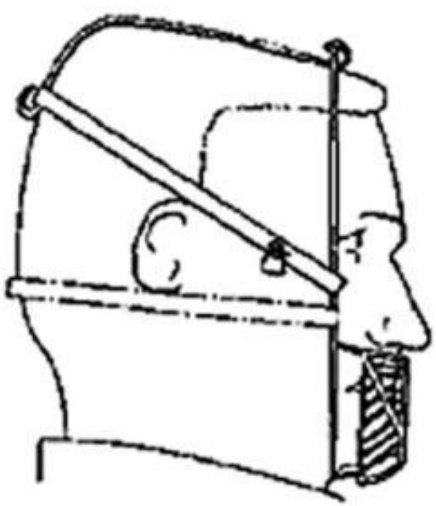
Medical device for analyte monitoring and drug delivery (Theranos)



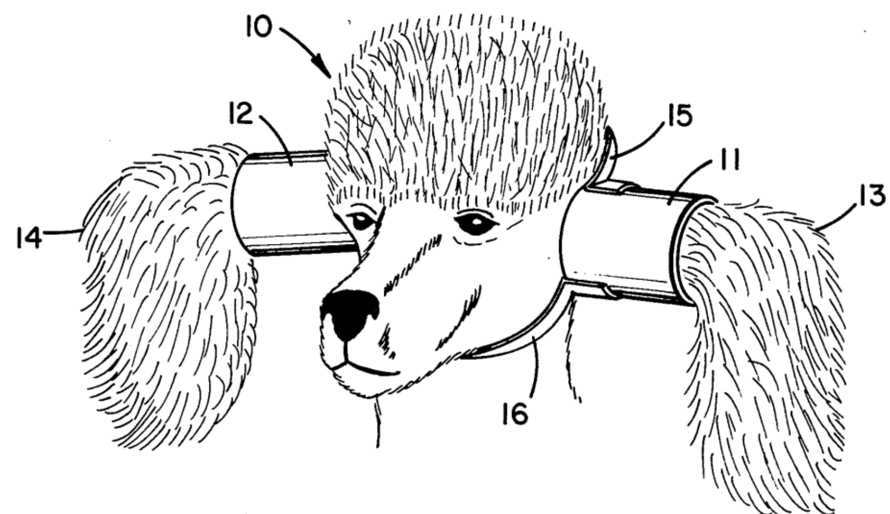
High-Five Machine



Gerbil Shirt



Anti Eating Device



Dog Ear Protection

Note, that I could not actually verify them as real patents, but they could easily be some

Goal:

We want to build a **model** to automatically **classify patents** into useful or bogus?

What do we need?

1. The patent data (easy thanks to Google Patents)
2. A training data set:
   some pre-labeled patents
3. A model

What do we need?

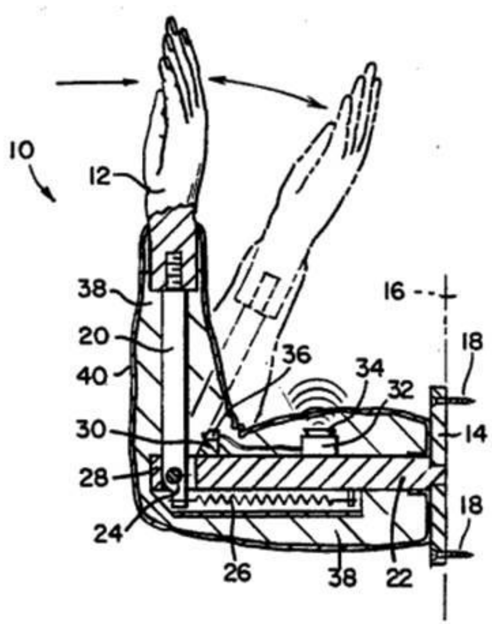
1. The patent data (easy thanks to Google Patents)
2. A training data set:
   some pre-labeled patents
3. A model

How do we get a labeled data set?

How do we get a labeled data set?

A Crowd Task

Is this Patent Bogus?

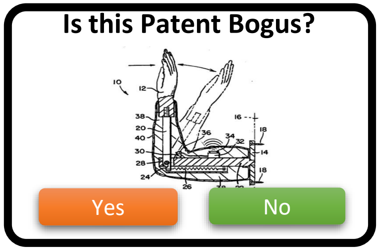


Yes

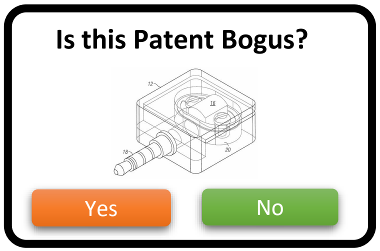
No

$$l = 1$$

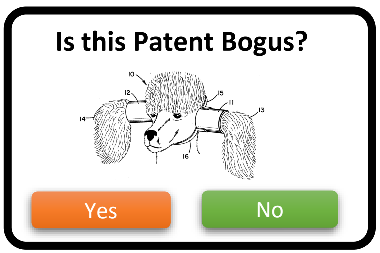
A Crowd Task



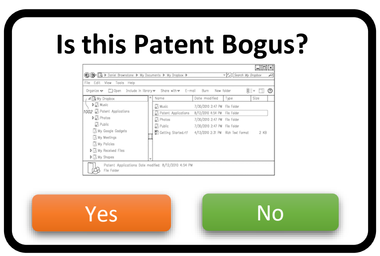
o_1



o_2

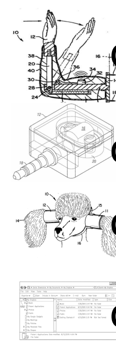


o_3



o_4

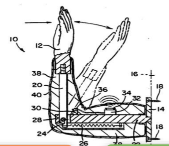
$\vdots$



$$l[n] = \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ \\ \end{array} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$$

A Crowd Task

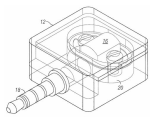
Is this Patent Bogus?



Yes No

o_1

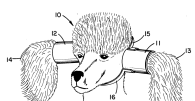
Is this Patent Bogus?



Yes No

o_2

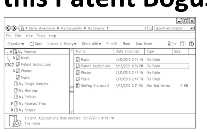
Is this Patent Bogus?



Yes No

o_3

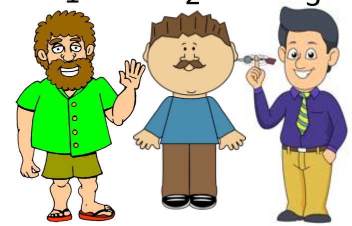
Is this Patent Bogus?

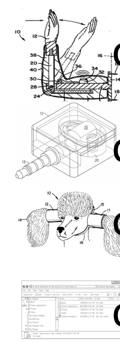


Yes No

o_4

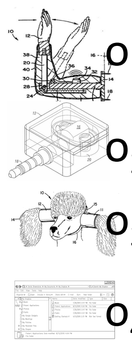
⋮



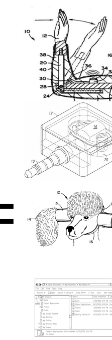


$$l[k][n] = \begin{array}{cc} & \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ \\ \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$

What should the final labels be?



$$l[k][n] = \begin{array}{cc} & \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ \\ \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$



$$T(n) = \begin{array}{cc} \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ \\ \end{array} & \begin{pmatrix} ? \\ ? \\ ? \\ ? \\ \vdots \end{pmatrix} \end{array}$$

Maximum Likelihood Estimate

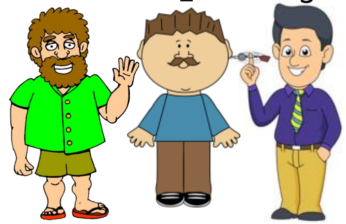
- Given some data X= $(x_1, \dots, x_n)$

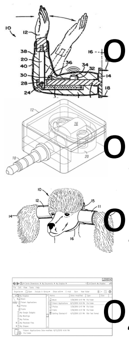
- Model $\mathcal{L}(\theta, X) = p_\theta(X) = \prod_i^n p_\theta(x_i)$

- **Maximum Likelihood Estimator (MLE)**

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathcal{L}(\theta, X)$$

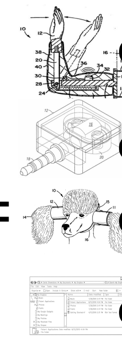
A Maximum Likelihood Estimate (MLE)





$$l[k][n] = \begin{array}{c c} & \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ \ \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$

What should the final labels be?

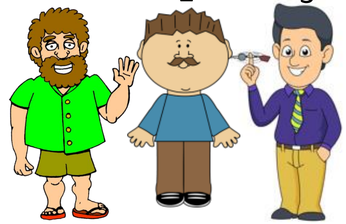


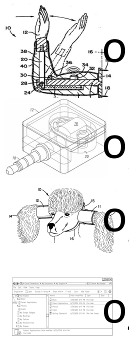
$$T(n) = \begin{array}{c c} \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ \ \end{array} & \begin{pmatrix} ? \\ ? \\ ? \\ ? \\ \vdots \end{pmatrix} \end{array}$$

$$\begin{array}{cc} \text{Bogus} & \text{Not Bogus} \end{array}$$

$$\begin{pmatrix} 3/3 & 0/3 \\ 0/3 & 3/3 \\ 2/3 & 1/3 \\ 1/3 & 2/3 \\ \vdots & \vdots \end{pmatrix}$$

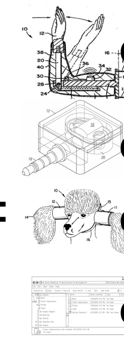
A Maximum Likelihood Estimate (MLE)





$$l[k][n] = \begin{array}{c|ccc} & k_1 & k_2 & k_3 \\ \hline o_1 & 1 & 1 & 1 \\ o_2 & 0 & 0 & 0 \\ o_3 & 0 & 1 & 1 \\ o_4 & 0 & 0 & 1 \\ & \vdots & \vdots & \vdots \end{array}$$

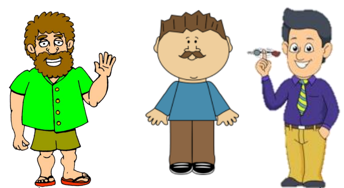
What should the final labels be?

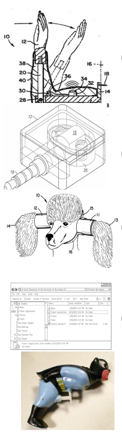


$$T(n) = \begin{array}{c|c} o_1 & 1 \\ o_2 & 0 \\ o_3 & 1 \\ o_4 & 0 \\ & \vdots \end{array}$$

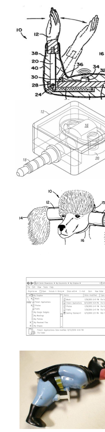
$$\begin{array}{cc} \text{Bogus} & \text{Not Bogus} \\ \left(\begin{matrix} 3/3 \\ 0/3 \\ 2/3 \\ 1/3 \\ \vdots \end{matrix}\right. & \left.\begin{matrix} 0/3 \\ 3/3 \\ 1/3 \\ 2/3 \\ \vdots \end{matrix}\right) \end{array}$$

So Everything is Good



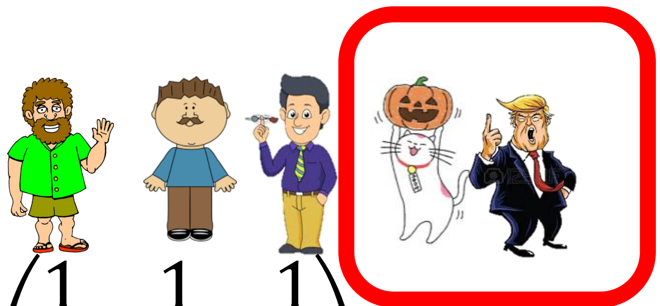


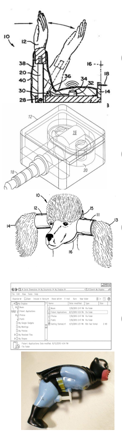
$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$



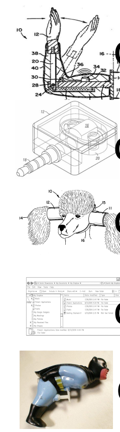
$$T(n) = \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

But what happens if we add
Crazy Cat with Pumpkin and the
Trumpworker



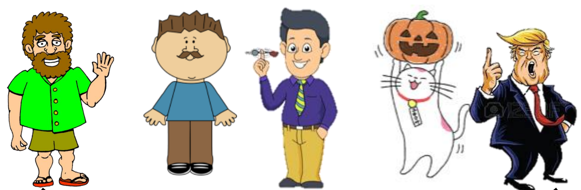


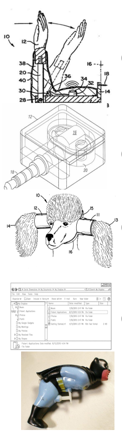
$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$



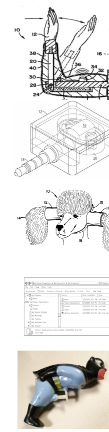
$$T(n) = \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

What if the Workers do not have the same Quality?





$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

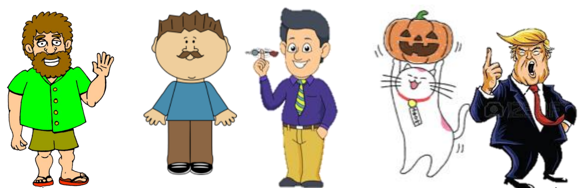


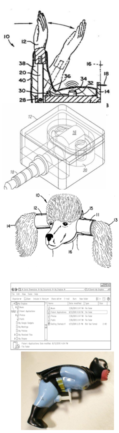
$$T(n) = \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 \\ 0 \\ \mathbf{0} \\ 0 \\ 1 \end{pmatrix}$$

What if the Workers do not have the same Quality?

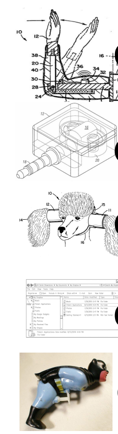
Latent (hidden) Variables

$Z_1 \quad Z_2 \quad Z_3 \quad Z_4 \quad Z_5$





$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$



$$T(n) = \begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{pmatrix}$$

Maximum Likelihood Estimate

- Given some data X= $(x_1, \ldots, x_n)$
- Model $\mathcal{L}(\theta, X) = p_\theta(X) = \prod_i^n p_\theta(x_i)$
- **Maximum Likelihood Estimator (MLE)**

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathcal{L}(\theta, X)$$

Maximum Likelihood Estimate

- Given some data X= $(x_1, \ldots, x_n)$
- Model $\mathcal{L}(\theta, X, Z) = p_\theta(X, Z) = \prod_i^n p_\theta(x_i, z)$
- **Maximum Likelihood Estimator (MLE)**

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta, X) = \sum_z p_\theta(X, Z)$$

  - Z has been marginalized
  - Hard to compute

Expectation Maximization Algorithm

Initialize $\theta \in \Theta$

For t = 0,1,2,…

E-Step: Calculate the expected value of the log likelihood function, with respect to the conditional distribution of Z given X under the current estimate of the parameters θ_t:

$$Q(Q|\theta_t) = E_{Z|X,\theta_t}[\log \mathcal{L}(\theta, X, Z)]$$

M-Step: Find the parameter that maximizes this quantity

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmax}}\, Q(Q|\theta_t)$$

EM – In our Example

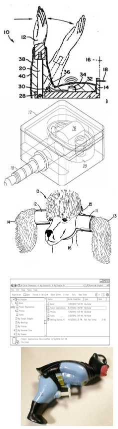
Initialize θ_0

For t = 0,1,2,…

E-Step: Calculate the expected labels (e.g., bogus or not-bogus) given θ_t

M-Step: Given the estimated label, optimize θ and set it to θ_{t+1}

EM Algorithm - Example



$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Guess

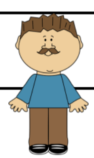
True		Bogus	!Bogus
	Bogus	1	0
	!Bogus	0	1

Guess

True		Bogus	!Bogus
	Bogus	1	0
	!Bogus	0	1

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

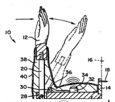
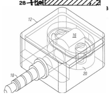
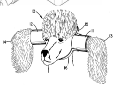
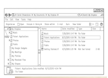
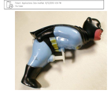
Guess

True		Bogus	!Bogus
	Bogus	1	0
	!Bogus	0	1

Guess

True		Bogus	!Bogus
	Bogus	1	0
	!Bogus	0	1

EM Algorithm - Example

$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
\qquad
\begin{array}{cc}
\text{Bogus} & \text{Not Bogus} \\
\end{array}
\begin{pmatrix}
0.6 & 0.4 \\
0.4 & 0.6 \\
0.4 & 0.6 \\
0.2 & 0.8 \\
0.6 & 0.4
\end{pmatrix}
$$

Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

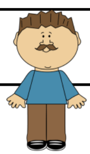
Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

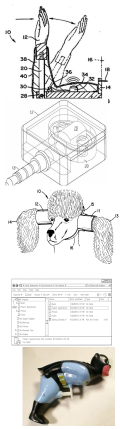
Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

EM Algorithm - Example



$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array}\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{array}{cc} \text{Bogus} & \text{Not Bogus} \end{array}$$

$$\begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \\ 0.6 & 0.4 \end{pmatrix}$$

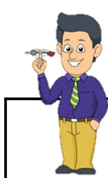
"Correct" Labels

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Guess

True

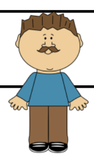
	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1



Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1



Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

Guess

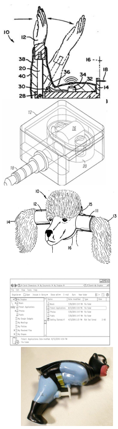
True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

EM Algorithm - Example

"Correct" Labels



$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
\qquad
\begin{pmatrix}
1 \\ 0 \\ 0 \\ 0 \\ 1
\end{pmatrix}
$$

Guess

True

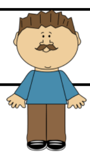
	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

Guess

True

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?



Guess

True

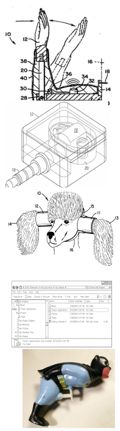
	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

Guess

True

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

EM Algorithm - Example



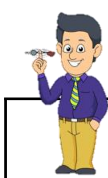
$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

"Correct" Labels

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

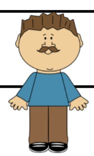
Guess

True	Bogus	!Bogus
Bogus	1	0.25
!Bogus	0	0.75



Guess

True	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?



Guess

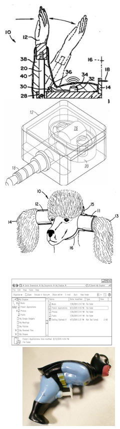
True	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

Guess

True	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

EM Algorithm - Example



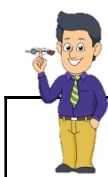
$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

"Correct" Labels

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

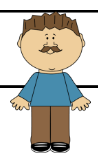
Guess

True	Bogus	!Bogus
Bogus	1	0.25
!Bogus	0	0.75



Guess

True	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?



Guess

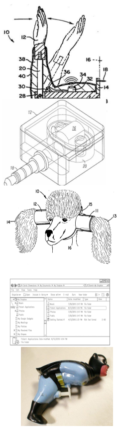
True	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

Guess

True	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

EM Algorithm - Example



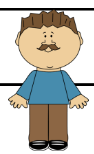
$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
\qquad
\begin{array}{c}
\text{“Correct” Labels} \\
\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
\end{array}
$$

Guess

	Bogus	!Bogus
Bogus	1	0.25
!Bogus	0	0.75

True

Guess

	Bogus	!Bogus
Bogus	0.66	0
!Bogus	0.33	1

True

Guess

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

True

Guess

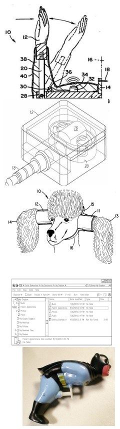
	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

True

	Bogus	!Bogus
Bogus	?	?
!Bogus	?	?

EM Algorithm - Example

"Correct" Labels



$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
\qquad
\begin{pmatrix}
1 \\ 0 \\ 0 \\ 0 \\ 1
\end{pmatrix}
$$

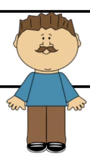
Guess

True		Bogus	!Bogus
	Bogus	1	0.25
	!Bogus	0	.75

Guess

True		Bogus	!Bogus
	Bogus	.66	0
	!Bogus	.33	1

Guess

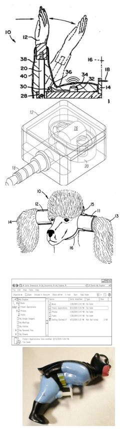
True		Bogus	!Bogus
	Bogus	0.66	0
	!Bogus	0.33	1

Guess

True		Bogus	!Bogus
	Bogus	.5	0.33
	!Bogus	.5	0.66

	Bogus	!Bogus
Bogus	0	0.66
!Bogus	1	0.33

EM Algorithm - Example



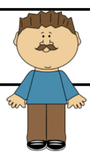
$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
\qquad
\begin{array}{c}
\text{“Correct”} \\ \text{Labels} \\
\begin{pmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{pmatrix}
\end{array}
$$

Guess

True	Bogus	!Bogus
Bogus	1	0.25
!Bogus	0	.75

Guess

True	Bogus	!Bogus
Bogus	.66	0
!Bogus	.33	1



Guess

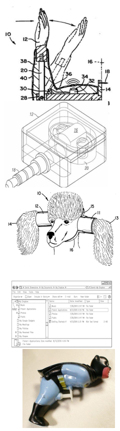
True	Bogus	!Bogus
Bogus	0.66	0
!Bogus	0.33	1

Guess

True	Bogus	!Bogus
Bogus	.5	0.33
!Bogus	.5	0.66

	Bogus	!Bogus
Bogus	0	0.66
!Bogus	1	0.33

EM Algorithm - Example



$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
$$

$$
\begin{array}{cc}
\text{Bogus} & \text{Not Bogus} \\
\end{array}
$$

$$
\begin{pmatrix}
1 + .66 + .66 + .33 + .66 & 0 + .33 + .33 + .66 + .33 \\
0.25 + 0 + 0 + .5 + 0 & .75 + 1 + 1 + 0.5 + 1 \\
0.25 + .66 + .66 + 0.33 + .66 & .75 + .33 + .33 + 0.66 + .33 \\
0.25 + 0 + 0 + .33 + 0 & .75 + 1 + 1 + .66 + 1 \\
.25 + .66 + .66 + .5 + .66 & .75 + .33 + .33 + .5 + .33
\end{pmatrix}
$$

Guess

True

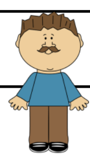
	Bogus	!Bogus
Bogus	1	0.25
!Bogus	0	.75

Guess

True

	Bogus	!Bogus
Bogus	.66	0
!Bogus	.33	1

	Bogus	!Bogus
Bogus	0	0.66
!Bogus	1	0.33



Guess

True

	Bogus	!Bogus
Bogus	0.66	0
!Bogus	0.33	1

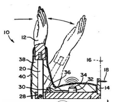
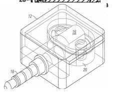
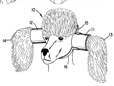
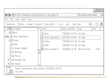
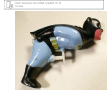
Guess

True

	Bogus	!Bogus
Bogus	.5	0.33
!Bogus	.5	0.66

EM Algorithm - Example

$$
\begin{array}{cc}
o_1 & \\
o_2 & \\
o_3 & \\
o_4 & \\
o_5 &
\end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{pmatrix}
$$

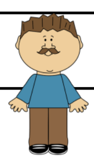






$$
\begin{array}{cc}
\text{Bogus} & \text{Not Bogus}
\end{array}
$$

$$
\begin{pmatrix}
1 + .66 + .66 + .33 + .66 & 0 + .33 + .33 + .66 + .33 \\
0.25 + 0 + 0 + .5 + 0 & .75 + 1 + 1 + 0.5 + 1 \\
0.25 + .66 + .66 + 0.33 + .66 & .75 + .33 + .33 + 0.66 + .33 \\
0.25 + 0 + 0 + .33 + 0 & .75 + 1 + 1 + .66 + 1 \\
.25 + .66 + .66 + .5 + .66 & .75 + .33 + .33 + .5 + .33
\end{pmatrix}
$$

Guess

		Bogus	!Bogus
True	Bogus	1	0.25
	!Bogus	0	.75



Guess

		Bogus	!Bogus
True	Bogus	0.66	0
	!Bogus	0.33	1

Guess

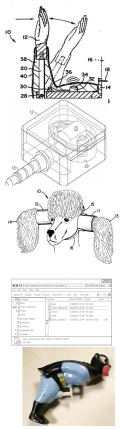
		Bogus	!Bogus
True	Bogus	.66	0
	!Bogus	.33	1

Guess

		Bogus	!Bogus
True	Bogus	.5	0.33
	!Bogus	.5	0.66

	Bogus	!Bogus
Bogus	0	0.66
!Bogus	1	0.33

EM Algorithm - Example



$$
\begin{array}{c}
o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5
\end{array}
\begin{pmatrix}
\boxed{1 \quad 1 \quad 1 \quad 0 \quad 0} \\
0 \quad 0 \quad 0 \quad 1 \quad 1 \\
0 \quad 1 \quad 1 \quad 0 \quad 0 \\
0 \quad 0 \quad 0 \quad 0 \quad 1 \\
0 \quad 1 \quad 1 \quad 1 \quad 0
\end{pmatrix}
$$

$$
\begin{array}{cc}
\text{Bogus} & \text{Not Bogus} \\
\end{array}
$$

$$
\begin{pmatrix}
1 + .66 + .66 + .33 + .66 & 0 + .33 + .33 + .66 + .33 \\
0.25 + 0 + 0 + .5 + 0 & .75 + 1 + 1 + 0.5 + 1 \\
0.25 + .66 + .66 + 0.33 + .66 & .75 + .33 + .33 + 0.66 + .33 \\
0.25 + 0 + 0 + .33 + 0 & .75 + 1 + 1 + .66 + 1 \\
.25 + .66 + .66 + .5 + .66 & .75 + .33 + .33 + .5 + .33
\end{pmatrix}
$$

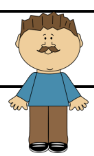
Guess

True		Bogus	!Bogus
	Bogus	1	0.25
	!Bogus	0	.75

Guess

True		Bogus	!Bogus
	Bogus	.66	0
	!Bogus	.33	1

	Bogus	!Bogus
Bogus	0	0.66
!Bogus	1	0.33



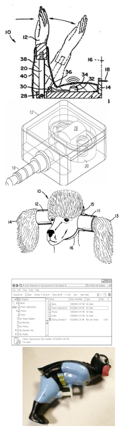
Guess

True		Bogus	!Bogus
	Bogus	0.66	0
	!Bogus	0.33	1

Guess

True		Bogus	!Bogus
	Bogus	.5	0.33
	!Bogus	.5	0.66

EM Algorithm - Example



$$\begin{array}{c} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{array}{cc} \text{Bogus} & \text{Not Bogus} \end{array} \qquad \text{“Correct” Labels}$$

$$\begin{pmatrix} 0.66 & .33 \\ 0.15 & .85 \\ 0.52 & 0.48 \\ .12 & 0.88 \\ 0.55 & 0.45 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Guess

True

	Bogus	!Bogus
Bogus	1	0.25
!Bogus	0	.75

Guess

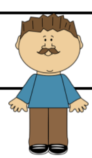
True

	Bogus	!Bogus
Bogus	.66	0
!Bogus	.33	1

	Bogus	!Bogus
Bogus	0	0.66
!Bogus	1	0.33

Guess

True

	Bogus	!Bogus
Bogus	0.66	0
!Bogus	0.33	1

Guess

True

	Bogus	!Bogus
Bogus	.5	0.33
!Bogus	.5	0.66

Dawid and Skene EM Algorithm [1]

Input: Labels $l[k][n]$ from worker (k) to object o_n,

Output: Confusion matrix $\pi_{ij}^{(k)}$ for each worker (k), Correct labels $T(o_n)$ for each object o_n, Class priors $Pr\{C\}$ for each class C

1 Initialize error rates $\pi_{ij}^{(k)}$ for each worker (k) (e.g., assume each worker is perfect);

2 Initialize correct label for each object $T(o_n)$ (e.g., using majority vote);

3 **while** *not converged* **do**

4   Estimate the correct label $T(o_n)$ for each object, using the labels $l[\cdot][n]$ assigned to o_n by workers, weighting the votes using the error rates $\pi_{ij}^{(k)}$;

5   Estimate the error rates $\pi_{ij}^{(k)}$, for each worker (k), using the correct labels $T(o_n)$ and the assigned labels $l[k][n]$;

6   Estimate the class priors $Pr\{C\}$, for each class C;

7 **end**

8 **return** *Estimated error rates* $\pi_{ij}^{(k)}$, *Estimated correct labels* $T(o_n)$, *Estimated class priors* $Pr\{C\}$

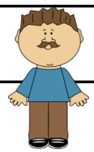
[1] Panos Ipeirotis, Foster Provost, Jing Wang: **Quality management on Amazon Mechanical Turk.** Proceedings of the ACM SIGKDD Workshop on Human Computation, 2010

[2] Dawid, A. P., and Skene, A. M. **Maximum likelihood estimation of observer error-rates using the EM algorithm**. Applied Statistics 28, 1 (Sept. 1979), 20–28.

Confusion Matrices in the 2nd iteration

True

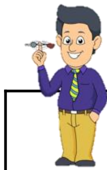
	Bogus	!Bogus
Bogus	1	0.5
!Bogus	0	.5



Guess

True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1



True

	Bogus	!Bogus
Bogus	1	0
!Bogus	0	1

Guess

True

	Bogus	!Bogus
Bogus	.5	0.66
!Bogus	.5	0.33

	Bogus	!Bogus
Bogus	0	1
!Bogus	1	0

Which worker is the worst?

EM-Algorithm: Many other applications

```
Initialize θ ∈ Θ
For t = 0,1,2,…
      E-Step:    Calculate the expected value
                 of the log likelihood
                 function, with respect to the
                 conditional distribution of Z
                 given X under the current
                 estimate of the parameters θ_t:
```

$$Q(Q|\theta_t) = E_{Z|X,\theta_t}[\log \mathcal{L}(\theta, X, Z)]$$

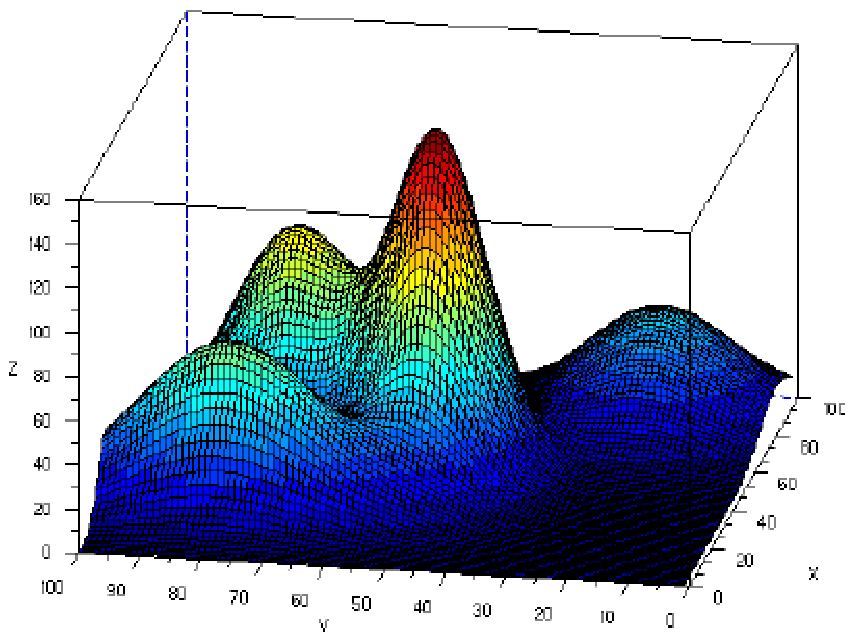
```
      M-Step:    Find the parameter that
                 maximizes this quantity
```

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmax}}\, Q(Q|\theta_t)$$

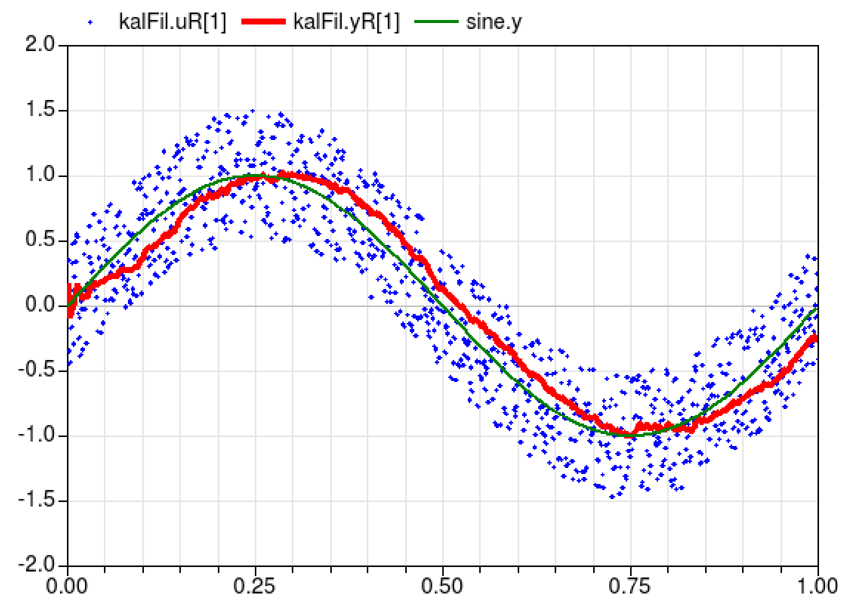
EM-Algorithm: Many other applications

Initialize $\theta \in \Theta$

Gaussian Mixture Models (GMM)



Kalman filter



M-Step: Find the parameter that maximizes this quantity

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmax}}\, Q\left(Q \middle| \theta_t\right)$$