DATA MINING 2
Clustering – Advanced Topics

Riccardo Guidotti

a.a. 2019/2020

UNIVERSITÀ DI PISA

Mixture Models and the EM Algorithm

Model-based Clustering (probabilistic)

- In order to understand our data, we will assume that there is a generative process (a model) that creates/describes the data, and we will try to find the model that **best fits** the data.
  - Models of different complexity can be defined, but we will assume that our model is a distribution from which data points are sampled
  - **Example**: the data is the height of all people in **Greece**
- In most cases, a single distribution is not good enough to describe all data points**: different parts of the data follow a different distribution**
  - **Example**: the data is the height of all people in Greece and China
  - We need a mixture model
  - Different distributions correspond to different clusters in the data.

EM Algorithm

Algorithm 9.2 EM algorithm.

1: Select an initial set of model parameters.
(As with K-means, this can be done randomly or in a variety of ways.)
2: **repeat**
3: **Expectation Step** For each object, calculate the probability that each object belongs to each distribution, i.e., calculate $prob(distribution\ j|\mathbf{x}_i, \Theta)$.
4: **Maximization Step** Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
5: **until** The parameters do not change.
(Alternatively, stop if the change in the parameters is below a specified threshold.)

EM (Expectation Maximization) Algorithm

- Initialize the values of the parameters in Θ to some random values
- Repeat until convergence
  - **E-Step**: Given the parameters Θ **estimate** the membership probabilities $\mathrm{P}\left(G_j \middle| x_i\right)$
  - **M-Step:** Given the probabilities $\mathrm{P}\left(G_j \middle| x_i\right)$, calculate the parameter values Θ that (in expectation) **maximize** the data likelihood

- **Examples**
  - **E-Step**: Assignment of points to clusters
    - K-means: **hard** assignment, EM: soft assignment
  - **M-Step**: Parameters estimation
    - K-means: Computation of centroids, EM: Computation of the new model parameters

Gaussian Distribution

- Example: the data is the height of all people in Greece
- Experience has shown that this data follows a Gaussian (Normal) distribution

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- μ = mean, σ = standard deviation

Mixture Gaussian Model

- What is a model?
  - A Gaussian distribution is defined by the mean μ and the standard deviation σ
  - We define our model as the pair of parameters $\theta = (\mu, \sigma)$
- More generally, a model is defined as a vector of parameters θ

- We want to find the normal distribution $N(\mu, \sigma)$ that best fits our data
  - Find the best values for μ and σ
  - But what does "best fit" mean?

Maximum Likelihood Estimation (MLE)

- Suppose that we have a vector $X = \{x_1, \dots, x_n\}$ of values
- We want to fit a Gaussian model $N(\mu, \sigma)$ to the data
- Probability of observing a point x_i

$$P(x_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- Probability of observing all points (we assume independence)

$$P(X) = \prod_{i=1}^{n} P(x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- We want to find the parameters $\theta = (\mu, \sigma)$ that maximizes the probability $P(X|\theta)$

Maximum Likelihood Estimation (MLE)

- The probability $P(X|\theta)$ as a function of θ is the **Likelihood** function

$$L(\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- It is usually easier to work with the **Log-Likelihood** function

$$LL(\theta) = -\sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2} - \frac{1}{2} n \log 2\pi - n \log \sigma$$

- Thus, the Maximum Likelihood Estimation for the Gaussian Model consists in finding the parameters μ, σ that maximize $LL(\theta)$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \mu_X$$

Sample Mean

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i-\mu)^2 = \sigma_X^2$$

Sample Variance

Maximum Likelihood Estimation (MLE)

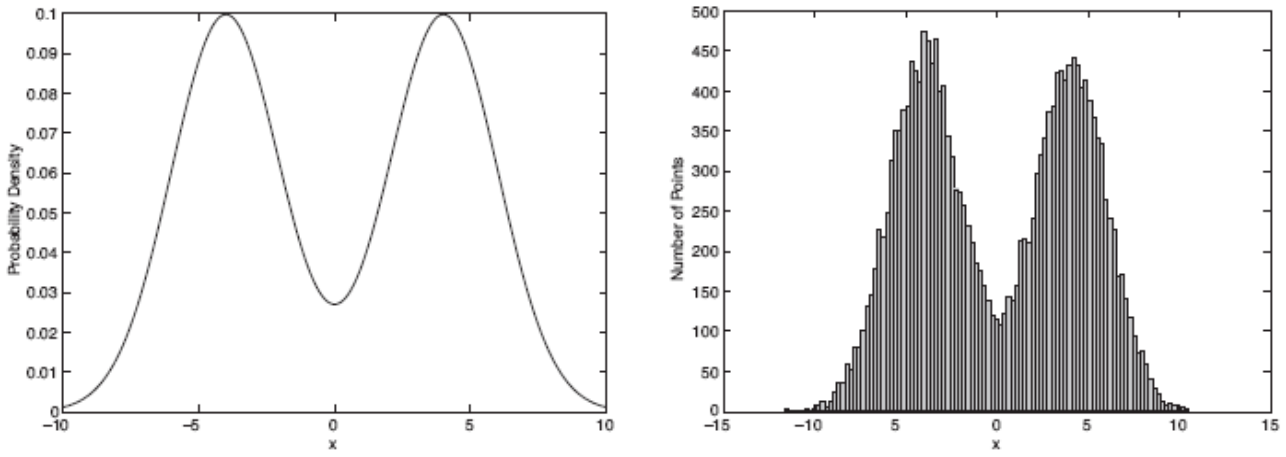
- Note: these are also the most likely parameters given the data.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

- If we have no prior information about θ, or X, then maximizing $P(\theta|X)$ is the same as maximizing $P(X|\theta)$.

Mixture of Gaussians

- Suppose that you have the heights of people from Greece and China and the distribution looks like the figure below (dramatization)



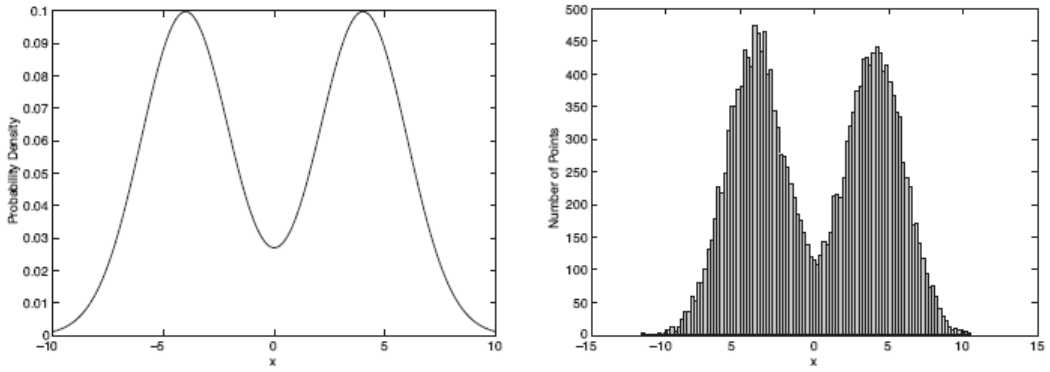
(a) Probability density function for the mixture model.

(b) 20,000 points generated from the mixture model.

Figure 9.2. Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

Mixture of Gaussians

- In this case the data is the result of the **mixture** of two Gaussians
  - One for Greek people, and one for Chinese people
  - Identifying for each value which Gaussian is most likely to have generated it will give us a clustering.



(a) Probability density function for the mixture model.

(b) 20,000 points generated from the mixture model.

Figure 9.2. Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

Mixture Model

- A value x_i is generated according to the following process:
  - First select the nationality
    - With probability π_G select Greek, with probability π_C select China ($\pi_G + \pi_C = 1$)
  - Given the nationality, generate the point from the corresponding Gaussian
    - $P(x_i|\theta_G) \sim N(\mu_G, \sigma_G)$ if Greece
    - $P(x_i|\theta_C) \sim N(\mu_C, \sigma_C)$ if China

Mixture Model

- Our model has the following parameters

$$\Theta = (\pi_G, \pi_C, \mu_G, \mu_C, \sigma_G, \sigma_C)$$

Mixture probabilities | Distribution Parameters

- For value x_i, we have:

$$P(x_i|\Theta) = \pi_G P(x_i|\theta_G) + \pi_C P(x_i|\theta_C)$$

- For all values $X = \{x_1, \dots, x_n\}$

$$P(X|\Theta) = \prod_{i=1}^{n} P(x_i|\Theta)$$

- We want to estimate the parameters that **maximize** the Liekelihood

Mixture Model

- Once we have the parameters $\theta = (\pi_G, \pi_C, \mu_G, \sigma_G, \mu_C, \sigma_C)$, we can **estimate** the **membership probabilities** $P(G|x_i)$ and $P(C|x_i)$ for each point x_i:

- This is the probability that point x_i Belongs to the Greek or the Chinese population (cluster)

$$P(G|x_i) = \frac{P(x_i|G)P(G)}{P(x_i|G)P(G) + P(x_i|C)P(C)}$$
$$= \frac{P(x_i|G)\pi_G}{P(x_i|G)\pi_G + P(x_i|C)\pi_C}$$

EM (Expectation Maximization) Algorithm

- Initialize the values of the parameters in θ to some random values
- Repeat until convergence
  - **E-Step**: Given the parameters Θ **estimate** the membership probabilities $\mathrm{P}(G|x_i)$ and $\mathrm{P}(C|x_i)$
  - **M-Step:** Calculate the parameter values Θ that (in expectation) **maximize** the data likelihood

$$\pi_G = \frac{1}{n}\sum_{i=1}^{n} P(G|x_i) \qquad \pi_C = \frac{1}{n}\sum_{i=1}^{n} P(C|x_i)$$

Fraction of population in G,C

$$\mu_C = \sum_{i=1}^{n} \frac{P(C|x_i)}{n * \pi_C} x_i \qquad \mu_G = \sum_{i=1}^{n} \frac{P(G|x_i)}{n * \pi_G} x_i$$

MLE Estimates if π's were fixed

$$\sigma_C^2 = \sum_{i=1}^{n} \frac{P(C|x_i)}{n * \pi_C} (x_i - \mu_C)^2 \qquad \sigma_G^2 = \sum_{i=1}^{n} \frac{P(G|x_i)}{n * \pi_G} (x_i - \mu_G)^2$$

Bisecting K-Means

Bisecting K-means

- Variant of K-Means that can produce a hierarchical clustering

```
Initialize the list of clusters to contain the cluster containing all points.
repeat
    Select a cluster from the list of clusters
    for i = 1 to number_of_iterations do
        Bisect the selected cluster using basic 2-Means
    end for
    Add the two clusters from the bisection with the lowest SSE to the list of clusters.
until Until the list of clusters contains K clusters
```

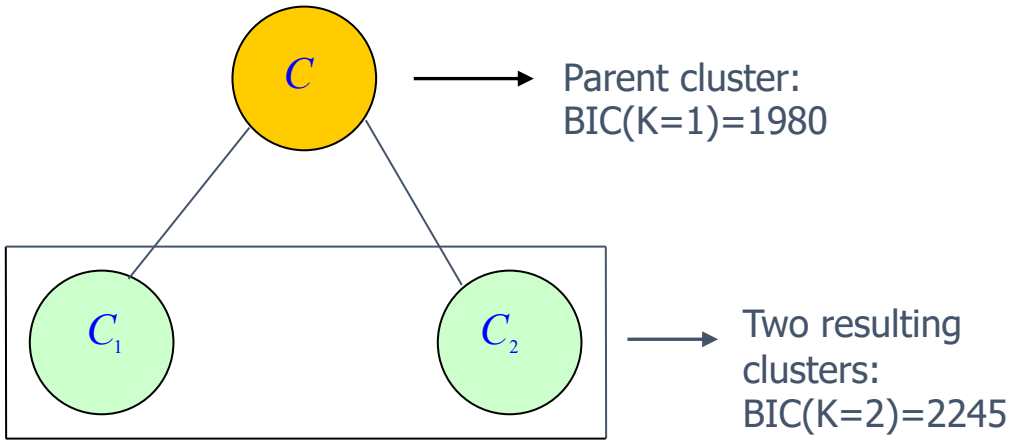
Bisecting K-means Limitations

- The algorithm is exhaustive terminating at singleton clusters (unless K is known or specified)
- Terminating at singleton clusters
  - Is time consuming
  - Singleton clusters are meaningless (i.e., over-splitting)
  - Intermediate clusters are more likely to correspond to real classes

- Bisecting K-Means do not uses any criterion for stopping bisections before singleton clusters are reached.

Bayesian Information Criterion (BIC)

- A strategy to stop the Bisecting algorithm when meaningful clusters are reached to avoid over-splitting.
- The **BIC** can be adopted as **splitting criterion** of a cluster in order to decide whether a cluster should split or no.
- BIC **measures the improvement** of the cluster structure between a cluster and its two children clusters.
- If the BIC of the parent is less than BIC of the children than we accept the bisection.



X-Means

X-Means

Search for the appropriate value of k in a given range $[r_1, r_{max}]$:

1. Improve Params
2. Improve Structure
3. If $k > r_{max}$ stop and return the best-scoring model

Improve Params

- Run K-Means with with the current k

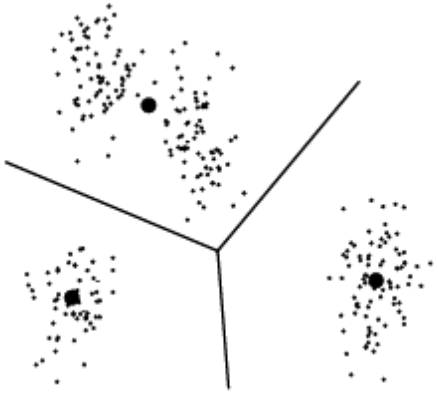
Improve Structure

- Recursively split each cluster in two and use local BIC to decide to keep the split

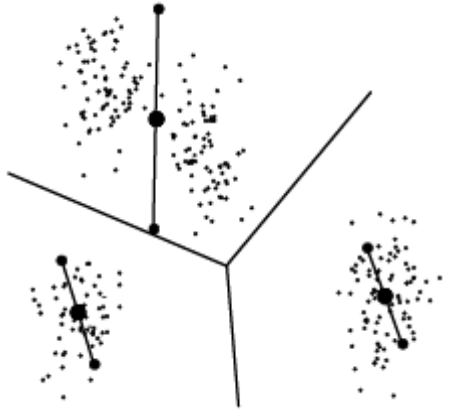
Finally, use global BIC score to decide which K to output at the end

X-Means

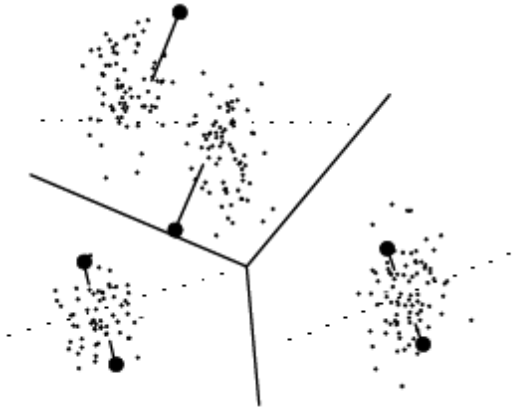
1. K-means with k=3



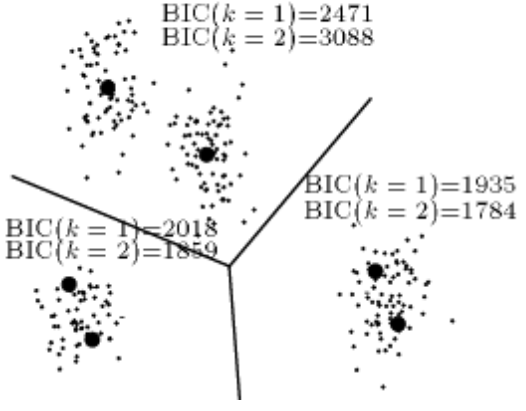
2. Split each centroid in 2 children moved a distance proportional to the region size in opposite direction (random)



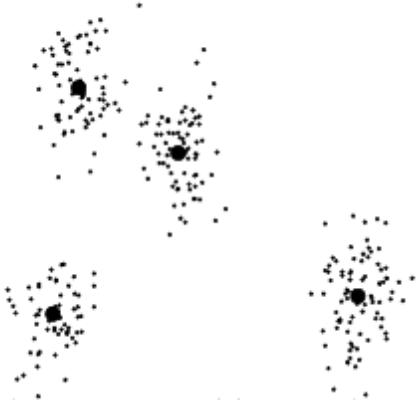
3. Run 2-means in each region locally



4. Compare BIC of parent and children



4. Only centroids with higher BIC survives



BIC Formula in X-Means

- The BIC score of a data collection is defined as (Kass and Wasserman, 1995):

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}\log R$$

- $\hat{l}_j(D)$ is the log-likelihood of the dataset D
- p_j is a function of the number of independent parameters: centroids coordinates, variance estimation.
- *R* is the number of points of a cluster, M is the number of dimensions
- Approximate the probability that the clustering in M_j is describing the real clusters in the data

BIC Formula in X-Means

- Adjusted Log-likelihood of the model.
- **The likelihood that the data is "explained by" the clusters** according to the spherical-Gaussian assumption of K-Means

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}\log R$$

- Focusing on the set D_n of points which belong to centroid n

$$\begin{aligned}\hat{l}(D_n) \quad = \quad & -\frac{R_n}{2}\log(2\pi) - \frac{R_n \cdot M}{2}\log(\hat{\sigma}^2) - \frac{R_n - K}{2} \\ & + R_n \log R_n - R_n \log R\end{aligned}$$

- It estimates how closely to the centroid are the points of the cluster.

Transactional Clustering

Clustering

- **Clustering**: Grouping of objects into different sets, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure

- Common distance functions:
  - Euclidean distance, Manhattan distance, …

- This kind of distance functions are suitable for **numerical data**

Not Only Numerical Data

Acceleration 0-100 (s)	Lenght (m)	Width (m)	Height (m)	Price (€)
12	4	1.6	1.7	20’ 000
14	3.7	1.5	1.65	16’ 000
15	3.5	1.5	1.6	12’ 000
9.4	4.2	1.8	1.7	24’ 000

Numerical Data

Categorical Data

Hairs	Eyes
brown	black
blond	blue
black	green
red	brown

Boolean and Categorical Attributes

- A **boolean** attribute corresponding to a single item in a transaction, if that item appears, the boolean attribute is set to ‘1’ or ‘0’ otherwise.

- A **categorical** attribute may have several values, each value can be treated as an item and represented by a boolean attribute.

Market Basket Data

- A transaction represents one customer, and each transaction contains set of items purchased by the customer.
- Clustering customers reveals customers with similar buying patterns putting them into the same cluster.
- It is useful for
  - Characterizing different customer groups
  - Targeted Marketing
  - Predict buying patterns of new customers based on profile
- A market basket database: A scenario where attributes of data points are non-numeric, transaction viewed as records with boolean attributes corresponding to a single item (TRUE if transaction contain item, FALSE otherwise).
- **Boolean** attributes are special case of **Categorical** attributes.

Shortcomings of Traditional Clustering

- For categorical data we:
  - Define new criterion for *neighbors* and/or *similarity*
  - Define the ordering criterion

- Consider the following 4 market basket transactions

T1= {1, 2, 3, 4}
T2= {1, 2, 4}
T3= {3}
T4= {4}

→

P1= (1, 1, 1, 1)
P2= (1, 1, 0, 1)
P3= (0, 0, 1, 0)
P4= (0, 0, 0, 1)

- using Euclidean distance to measure the closeness between all pairs of points, we find that d(P1,P2) is the smallest distance: **it is equal to 1**

Shortcomings of Traditional Clustering

- If we use a hierarchical algorithm then we merge P1 and P2 and get a new cluster (P12) with (1, 1, 0.5, 1) as a centroid
- Then, using Euclidean distance again, we find:
  - d(p12,p3)= $\sqrt{3.25}$
  - d(p12,p4)= $\sqrt{2.25}$
  - d(p3,p4)= $\sqrt{2}$
- So, **we should merge P3 and P4** since the distance between them is the shortest.
- **However, T3 and T4 don't have even a single common item.**
- So, using distance metrics as similarity measure for **categorical** data is not appropriate.

P1= (1, 1, 1, 1)
P2= (1, 1, 0, 1)
P3= (0, 0, 1, 0)
P4= (0, 0, 0, 1)

Clustering Algorithms for Categorical/Transactional Data

- K-Modes
- ROCK
- CLOPE
- TX-Means

K-Modes

$$\text{Minimise} \quad P(W, \boldsymbol{Q}) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l}\, d(X_i, Q_l)$$

$$\text{subject to} \quad \sum_{l=1}^{k} w_{i,l} = 1, \quad 1 \le i \le n$$

$$w_{i,l} \in \{0, 1\}, \quad 1 \le i \le n,\ 1 \le l \le k$$

- $X = \{ X_1, \ldots, X_n \}$ is the dataset of objects.
- $X_i = [x_1, \ldots, x_m]$ is an object i.e., a vector of m categorical attributes
- W is a matrix $n \times k$, with $w_{i,l}$ equal to 1 if X_i belongs to Cluster l, 0 otherwise.
- $Q = \{ Q_1, \ldots, Q_k \}$ is the set of representative objects (mode) for the k clusters.
- $d(X_i , Q_l)$ is a distance function for objects in the data

K-Modes: Distance

- K-Means as distance uses Euclidean distance

$$d(X,Y) = \sum_{i=1}^{m} (x_i - y_i)^2$$

- K-Modes as distance uses the number of mismatches between the attributes of two objects.

$$d_1(X, Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$

K-Modes: Mode

- K-Modes uses the mode as representative object of a cluster
- Given the set of objects in the cluster C_l the mode is get computing the max frequency for each attribute

$$f_r(A_j = c_{l,j} \mid X_l) = \frac{n_{c_{l,k}}}{n}$$

K-Modes: Algorithm

1. Select the initial objects as modes
2. Scan of the data to assign each object to the closer cluster identified by the mode
3. Re-compute the mode of each cluster
4. Repeat the steps 2 and 3 until no object changes the assigned cluster

ROCK: RObust Clustering using linK

- ROCK is a **hierarchical** algorithm for clustering transactional data (market basket databases)
- ROCK uses **links to cluster** instead of the classical distance notion
- ROCK uses the notion of **neighborhood** between pair of objects to identify **the number of links** between two objects

ROCK: The Neighbors Concept

- It captures a notion of **similarity**
  - A and B are neighbors if **sim(A, B) ≥ θ**

- ROCK uses the **Jaccard coefficient**
  - sim(A, B)= |A ∩ B| / | A U B |

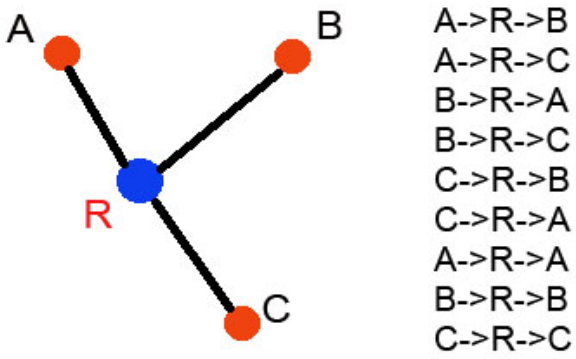
A = { 1 , 3 , 4 , 7 }

B = { 1 , 2 , 4 , 7 , 8 }

$$sim(A,B) = \frac{3}{6} = \frac{1}{2} = 0.5$$

ROCK: Links

- A **link** defines the number of common neighbors between two objects:
- **link(A, B) = |neighbor(A) ∩ neighbor(B) |**
- Higher values of *link(A, B)* means higher probability that *A* and *B* belong to the same cluster
- **Similarity** is **local** while **link** is capturing **global** information
- A point is considered a neighbor of itself
- There is a link from each neighbor of the "root" point back to itself through the root
- Therefore, if a point has *n* neighbors, then n^2 links are due to it.



ROCK: Example

- Data consisting of 6 Attributes: {Book, Water, Sun, Sand, Swimming, Reading}
  - {Book}
  - {Water, Sun, Sand, Swimming}
  - {Water, Sun, Sand, Reading}
  - {Reading, Sand}

- Resulting Jaccard Coefficient Matrix

	A	B	C	D
A	1	0	0	0
B	0	1	0.6	0.2
C	0	0.6	1	0.5
D	0	0.2	0.5	1

- Set Threshold = 0.2. Neighbors:
  - N(A)={A}; N(B)={B,C,D}
  - N(C)={B,C,D}, N(D) = {B,C,D}

- Number of Links Table
  - Link (B, C) = |{B,C,D}| = 3

	A	B	C	D
A	1	0	0	0
B	0	3	3	3
C	0	3	3	3
D	0	3	3	3

- Resulting Clusters after applying ROCK: {A}, {B,C,D}

ROCK – Criterion Function

Maximize $$E_l = \sum_{i=1}^{k} n_i * \sum_{p_q, p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}$$

$$f(\theta) = \frac{1-\theta}{1+\theta}$$

Dividing by the number **of expected links between pairs of objects in the cluster C_i** we avoid that objects with a low number of links are assigned all to the same cluster

Where ***C_i*** *denotes cluster i*
n_i *is the number of points in C_i*
k is the number of clusters
θ is the similarity threshold

This goodness measure helps to identify the best pair of clusters to be merged during each step of ROCK.

$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}$$

Number of expected cross-links between two clusters

ROCK: Clustering Algorithm

Input:

A set S of data points
Number of k clusters to be found
The similarity threshold

Output:

Groups of clustered data

The ROCK algorithm is divided into three major parts:

1. Draw a random sample from the data set
2. Perform a hierarchical agglomerative clustering algorithm
3. Label data

ROCK: Clustering Algorithm

Draw a random sample from the data set:

- Sampling is used to ensure scalability to very large data sets
- The initial sample is used to form clusters, then the remaining data on dataset is assigned to these clusters

ROCK: Clustering Algorithm

Perform a hierarchical agglomerative clustering algorithm:

- ROCK performs the following steps which are common to all hierarchical agglomerative clustering algorithms, but with different definition to the similarity measures:
  1. Places each single data point into a separate cluster
  2. Compute the similarity measure for all pairs of clusters
  3. Merge the two clusters with the highest similarity (goodness measure)
  4. Verify a stop condition. If it is not met then go to step 2.

ROCK: Clustering Algorithm

Label data

- Finally, the remaining data points are assigned to the clusters.
- This is done by selecting a random sample L_i from each cluster C_i, then we assign each point p to the cluster for which it has the strongest linkage with L_i.

CLOPE (Clustering with LOPE)

- Transactional clustering efficient for high dimensional data
- Uses a **global criterion function** that tries to increase the intra-cluster overlapping of transaction items **by increasing the height-to-width ratio of the cluster histogram**.

Example: 5 transactions {a,b} {a,b,c} {a,c,d} {d,e} {d,e,f}

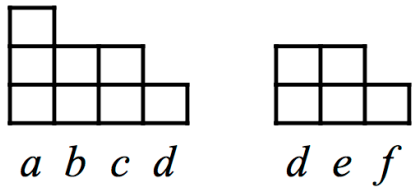
$D(C) = set$ of items in C

$$S(C) = \sum_{t_i \in C} |t_i|$$

$$W(C) = |D(C)|$$

$$H(C) = S(C) / W(C)$$

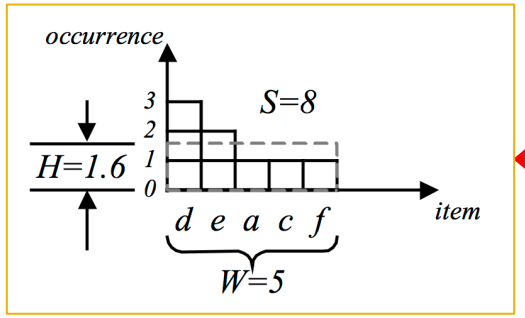
Clustering 1



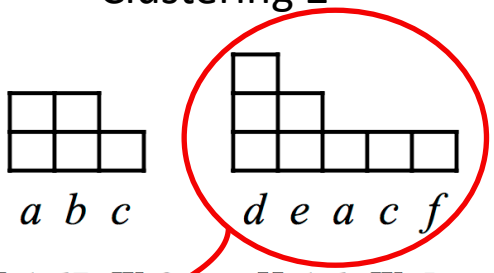
H=2.0, *W*=4 *H*=1.67, *W*=3

{*ab, abc, acd*} {*de, def*}

H/W=0.5 *H/W=0.55*



Clustering 2



H=1.67, *W*=3 *H*=1.6, *W*=5

{*ab, abc*} {*acd, de, def*}

H/W=0.55 *H/W=0.32*

Higher H/W means higher item overlapping

CLOPE: Criterion Function

- For evaluating the goodness of a clustering the **gradient of a cluster** is
- $G(C)=H(C)/W(C)=S(C)/W(C)^2$

Repulsion.
When *r* is large, transactions within the same cluster must share a large portion of common items.

$$Profit_r(\mathbf{C}) = \frac{\sum_{i=1}^{k} \frac{S(C_i)}{W(C_i)^r} \times |C_i|}{\sum_{i=1}^{k} |C_i|}$$

CLOPE: Algorithm

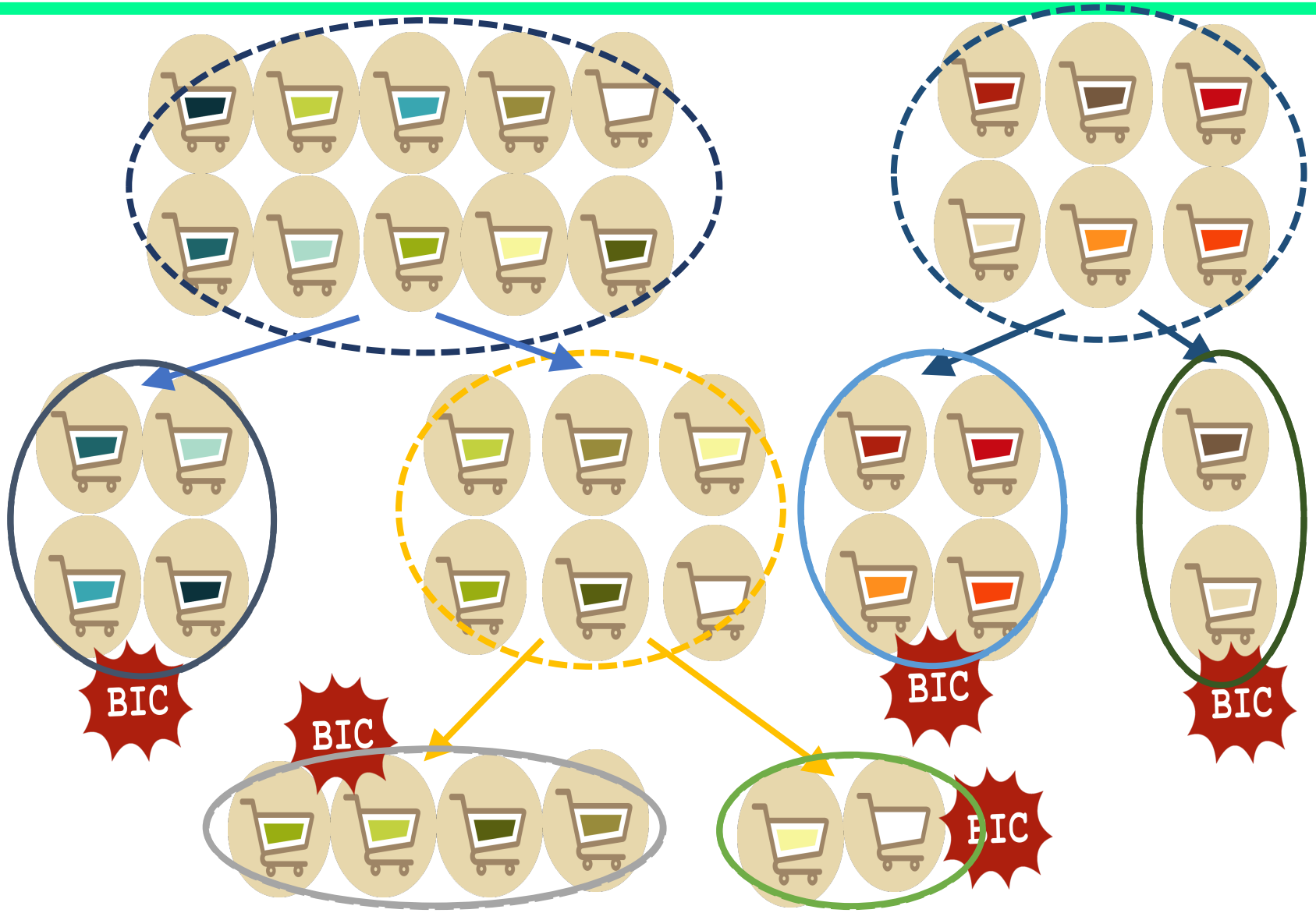
```
/* Phrase 1 - Initialization */
while not end of the database file
    read the next transaction ⟨t, unknown⟩;
    put t in an existing cluster or a new cluster C_i
      that maximize profit;
    write ⟨t, i⟩ back to database;

/* Phrase 2 - Iteration */
repeat
    rewind the database file;
    moved = false;
    while not end of the database file
        read ⟨t, i⟩;
        move t to an existing cluster or new cluster C_j
          that maximize profit;
        if C_i ≠ C_j then
            write ⟨t, j⟩;
            moved = true;
until not moved;
```

TX-MEANS

- A parameter-free clustering algorithm able to efficiently partitioning transactional data automatically
- Suitable for the case where clustering must be applied on a massive number of different datasets
  - E.g.: when a large set of users need to be analyzed individually and each of them has generated a long history of transactions
- TX-Means automatically estimates **the number of clusters**
- TX-Means provides the **representative transaction** of each cluster, which summarizes the pattern captured by that cluster.

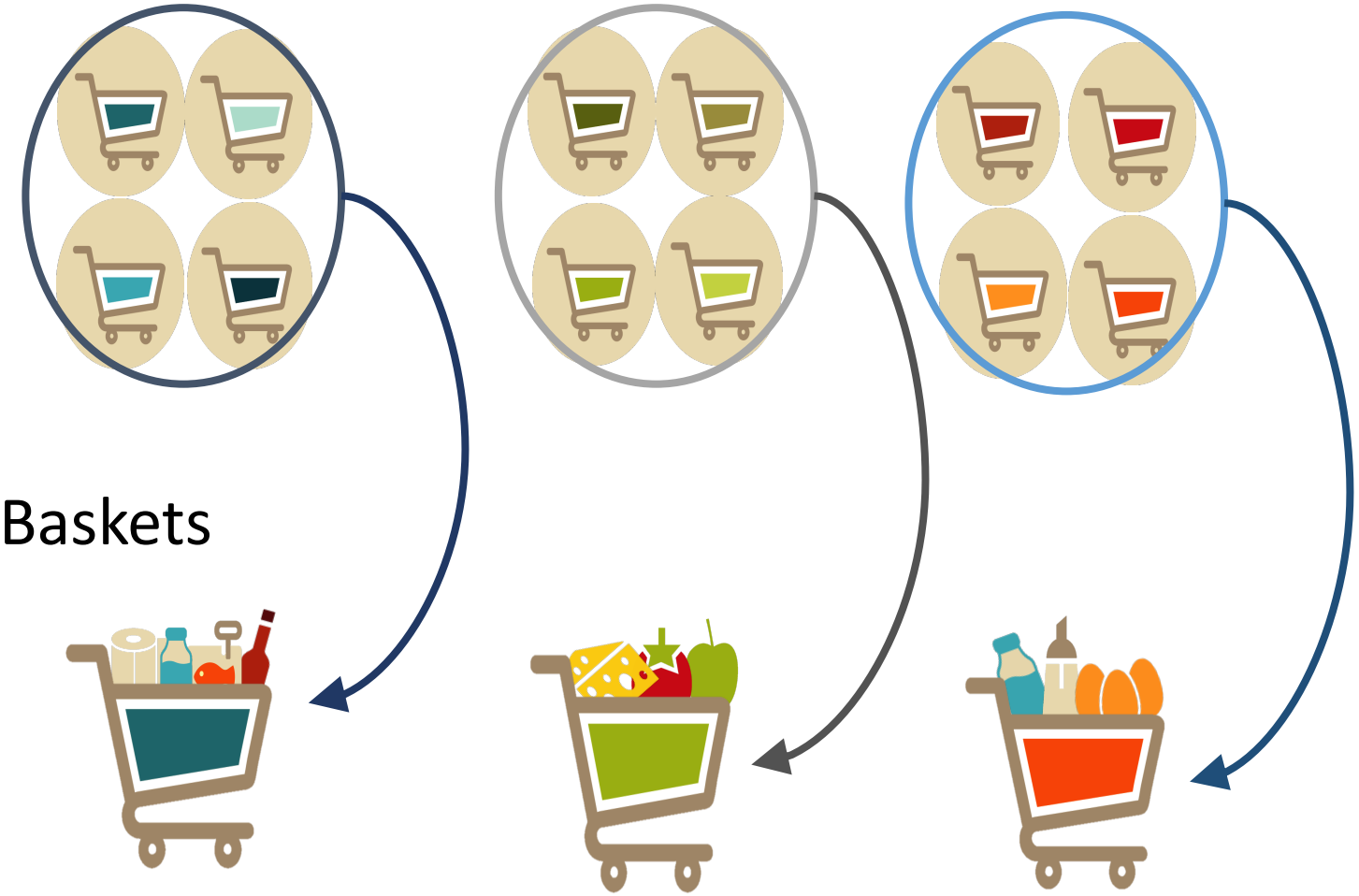
How It Works 1/3

How It Works 2/3



How It Works 3/3

- Clusters
- Representative Baskets



TX-Means Algorithm

```
TXMEANS(B: baskets):
• r <-- GETREPR(B);
• Q.push(B,r);
• While there is a cluster B,r to split in Q:
  • Remove common items from B;
  • B1, B2, r1, r2 <-- BISECTBASKET(B);
  • If BIC(B1,B2,r1,r2) > BIC(B,r) Then:
    • add B1,B2,r1,r2 to the clusters to split Q;
  • Else
    • add B,r to the clustering result C;
• Return C;
```

representative basket

bisecting schema

stopping criterion

Bisecting Schema

```
BISECTBASKET(B: baskets):
• SSE <-- inf;
• r1,r2 <-- select random initial baskets in B as representative;
• While True:
  • C1,C2 <-- assign baskets in B with respect to r1,r2;
  • r1_new <-- GETREPR(C1); r2_new <-- GETREPR(C2);
  • SSE_new <-- SSE(C1,C2,r1_new,r2_new);
  • If SSE_new >= SSE Then:
    • Return C1,C2,r1,r2;
  • r1,r2 <-- r1_new,r2_new;
```

overlap-based distance function: Jaccard coefficient

Get Representative Baskets

```
GETREPR(B: baskets):
- I <-- not common items in B;
- r <-- common items in B;
- While I is not empty:
  - Add to r the items with maximum frequency in I;
  - Calculate the distance between r and the baskets in B;
  - If the distance no longer decreases Then:
    - Return r;
  - Else
    - remove from I the items with maximum frequency;
- Return r;
```

overlap-based distance function (Jaccard coefficient)

Dealing with Big Datasets

- Clustering of a big individual transactional dataset B.
- TX-Means is scalable thanks to the following sampling strategy.

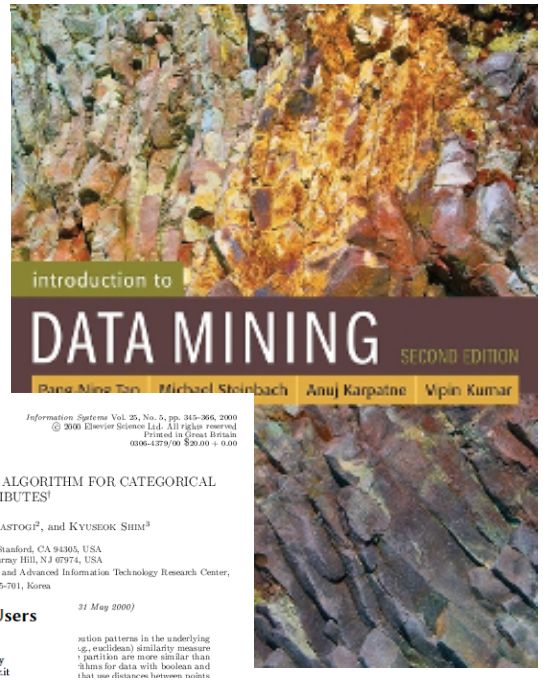
- Sampling strategy:
  - Random selection of a subset S of the baskets in B;
  - Run of TX-Means on the subset S and obtain clusters C and representative baskets R;
  - Assign the remaining baskets B/S to the clusters C using a nearest neighbor approach with respect to the representative baskets R.

References

- Advanced Clustering. Chapter 9.2.2. Introduction to Data Mining.
- Pelleg, Dan, and Andrew W. Moore. "X-means: Extending k-means with efficient estimation of the number of clusters." 2000.
- Guha, S., et al. ROCK: A robust clustering algorithm for categorical attributes. 2000.
- Yang, Y., et al. CLOPE: a fast and effective clustering algorithm for transactional data. 2002.
- Guidotti, R., et al. Clustering individual transactional data for masses of users. 2017.

X-means: Extending K-means with Efficient Estimation of the Number of Clusters

Dan Pelleg, Andrew Moore — School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

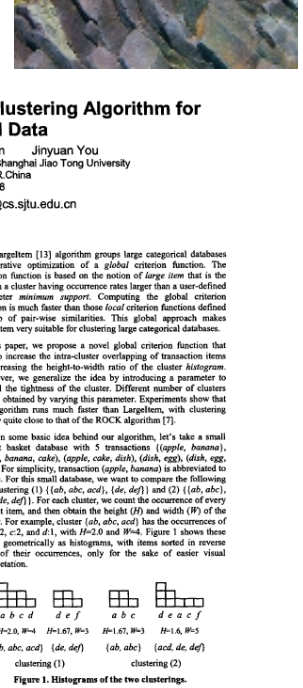


ROCK: A ROBUST CLUSTERING ALGORITHM FOR CATEGORICAL ATTRIBUTES

Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim

Clustering Individual Transactional Data for Masses of Users

Riccardo Guidotti, Anna Monreale, Mirco Nanni, Fosca Giannotti, Dino Pedreschi



CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data

Yiling Yang, Xudong Guan, Jinyuan You