UNIVERSITAT DE BARCELONA

Course. Introduction to Machine Learning
Work 1. Clustering Exercise

Session 2

Course 2021-2022

Dr. Maria Salamó Llorente

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,

University of Barcelona

UNIVERSITAT DE BARCELONA

Contents

OPTICS

With sklearn

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as **density-connected points** or based on an explicitly constructed density function
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - Need density parameters as termination condition
  - One scan
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - **OPTICS: Ankerst, et al (SIGMOD'99).**
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

OPTICS: a relative of DBSCAN

- DBSCAN is a Density-Based Clustering algorithm
- Reminder: In density based clustering we partition points into dense regions separated by not-so-dense regions.
- Important Questions:
  - How do we measure density?
  - What is a dense region?
- DBSCAN:
  - Density at point p: number of points within a circle of radius Eps
  - Dense Region: A circle of radius Eps that contains at least MinPts points

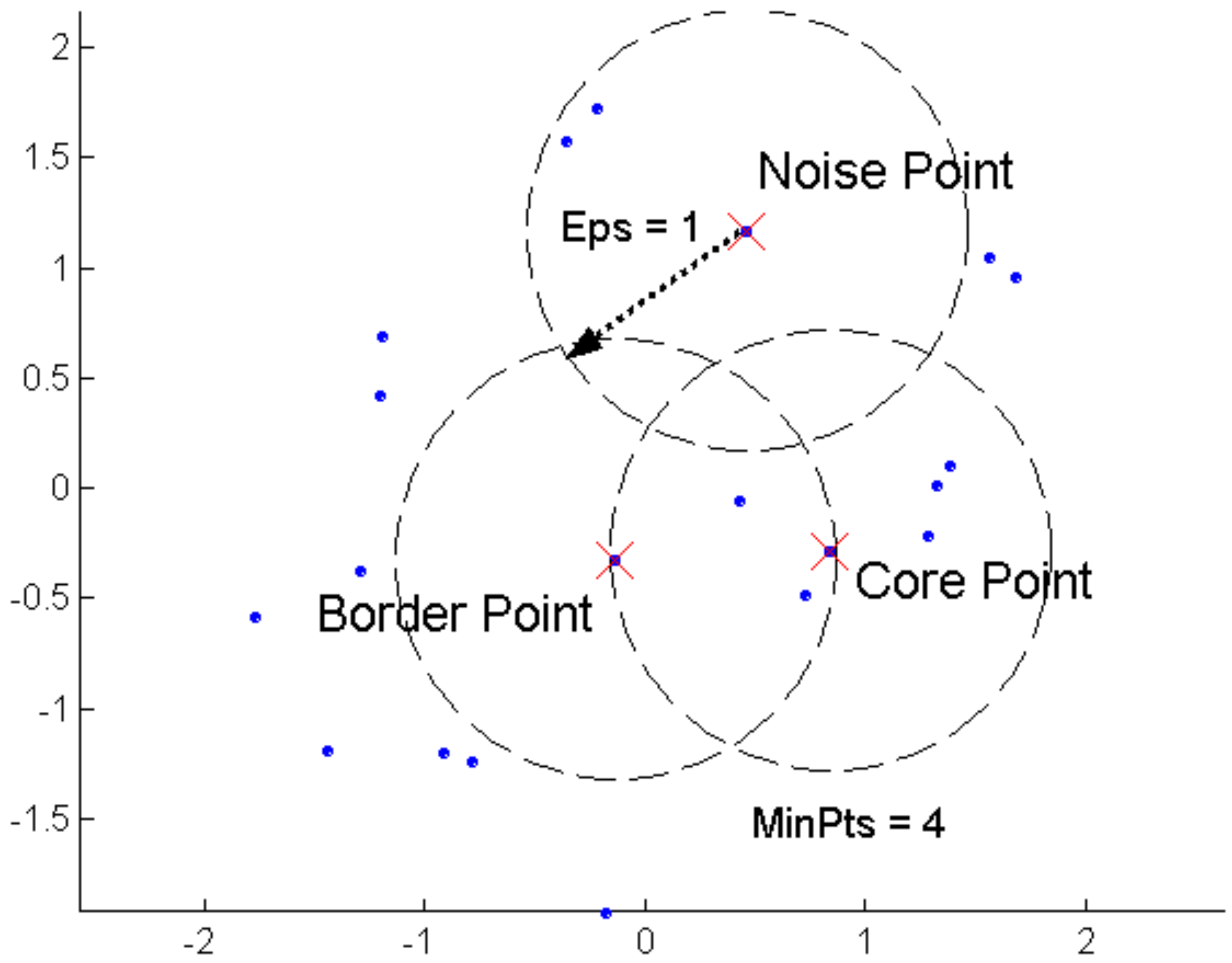
DBSCAN: Preliminary concepts

Characterization of points

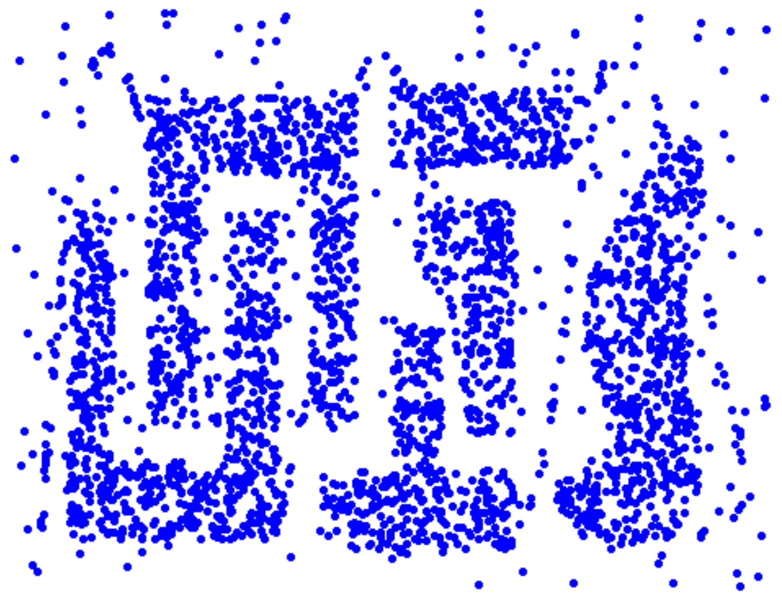
- **Epsilon parameter**, or Ɛ
  - Density = number of points within a specified radius (Eps)
  - For any point *p*, the epsilon defines a distance around the point
- **MinPts parameter** (minimum amount of points)
  - How many points must be within the Ɛ distance of a point p (including the point) to form a cluster
- **Core points**
  - A point is a **core point** if it has more than a specified number of points (*MinPts*) within its Ɛ distance (including itself)
  - These points belong in a dense region and are at the interior of a cluster
- **Border point**
  - A **border point** has fewer than *MinPts* within Ɛ, but is in the neighborhood of a core point
- **Noise point**
  - A **noise point** is any point that is not a core point or a border point

UNIVERSITAT DE BARCELONA

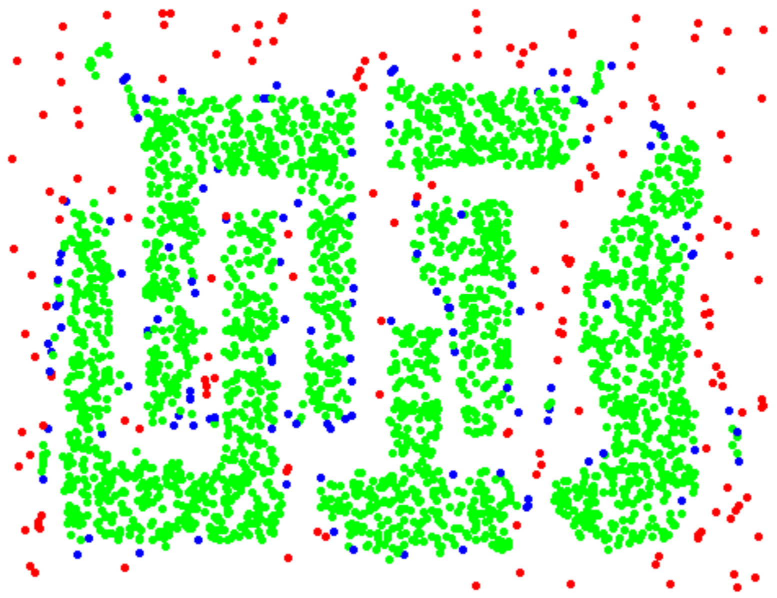
DBSCAN: Core, Border, and Noise Points



DBSCAN: Core, Border and Noise Points



Original Points



Point types: core, border and noise

Eps = 10, MinPts = 4

Parameter Estimation

- Parameters must be specified by the user
  - Ɛ = physical distance (radius),
  - *minPts* = desired minimum cluster size

minPts

- derived from the number of dimensions *D* in the data set, as *minPts* ≥ *D* + *1*
- *minPts* = *1* does not make sense, as then every point on its own will already be a cluster
- *minPts* must be chosen at least 3. Larger is better.
- larger the dataset, the larger the value of *minPts* should be chosen

Ɛ

- value can be chosen by using a k-distance graph
- If Ɛ is chosen much too small, a large part of the data will not be clustered
- If too high value, majority of objects will be in the same cluster
- In general, small values of Ɛ are preferable

OPTICS: A Cluster-Ordering Method

- **OPTICS**: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
  - Produces a special order of the database w.r.t. its density-based clustering structure
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure

Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: ordering points to identify the clustering structure. In Proceedings of the 1999 ACM SIGMOD international conference on Management of data (SIGMOD '99). Association for Computing Machinery, New York, NY, USA, 49–60. DOI:https://doi.org/10.1145/304182.304187

Note all the documents with this icon are in a zip file in campus virtual

UNIVERSITAT DE BARCELONA

OPTICS: Preliminary concepts

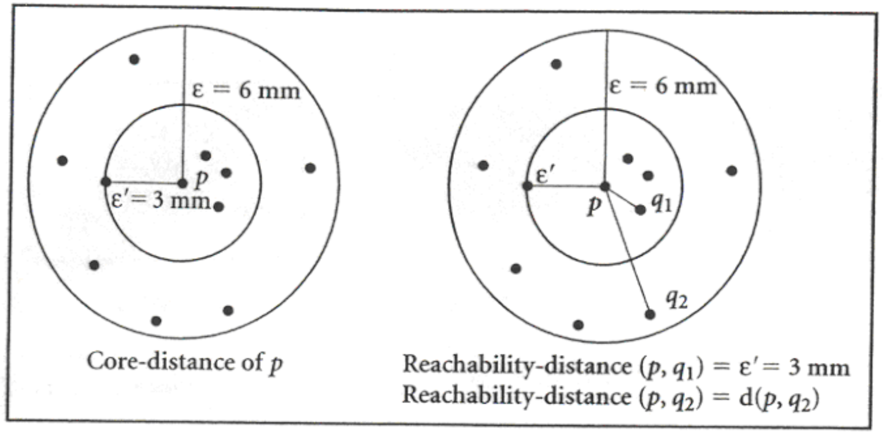
Characterization of points

- Use the preliminary concepts of DBSCAN
- **Core distance**
  - It is the minimum value of radius required to classify a given point as a core point.
  - Is undefined if the number of points in the neighborhood (including itself) is lower than the minimum of points required
- **Reachability distance**
  - Expresses the distance which is reachable from a core point.
  - The reachability distance between a point p and q is the maximum of the Core Distance of p and the Euclidean distance (or other distance metric) between p and q. Note that q should be a core point.

OPTICS: An Extension from DBSCAN

OPTICS was developed to overcome the difficulty of selecting appropriate parameter values for DBSCAN [Ankerst99].

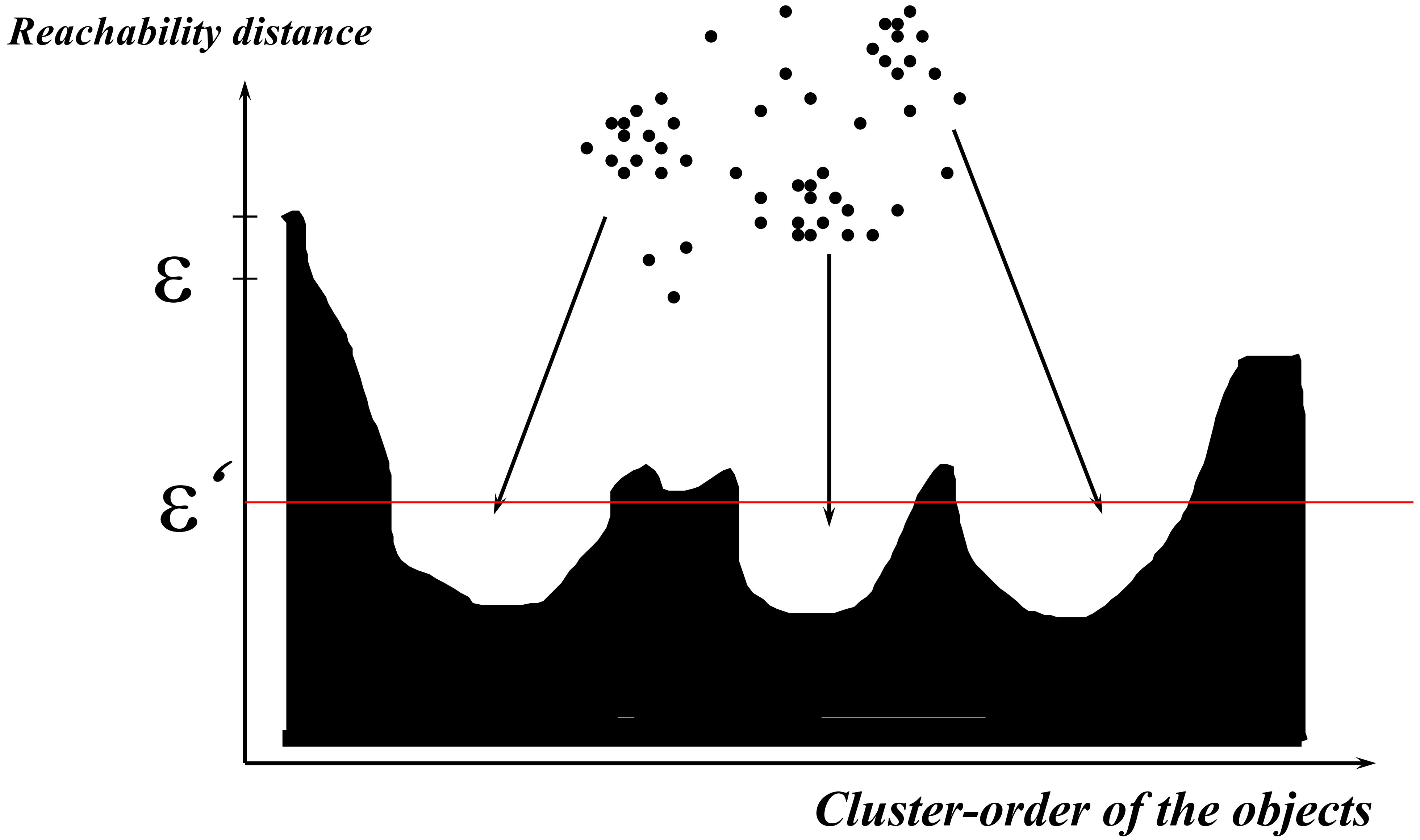
- The OPTICS algorithm finds clusters using the following steps:
  1) Create an ordering of the objects in a database, storing the core-distance and a suitable reachability distance for each object. Clusters with highest density will be finished first.
  2) Based on the ordering information produced by OPTICS, use another algorithm to extract clusters.
  3) Extract density-based clusters with respect to any distance e' that is smaller than the distance e used in generating the order.



ε = 6 mm
p
ε′ = 3 mm
Core-distance of p
ε = 6 mm
ε′
p
q1
q2
Reachability-distance (p, q1) = ε′ = 3 mm
Reachability-distance (p, q2) = d(p, q2)

OPTICS. The core distance of p is the distance e', between p and the fourth closest object. The reachability distance of q1 with respect to p is the core-distance of p (e'=3mm) since this is greater than the distance between p and q1. The reachability distance of q2 with respect to p is the distance between p and q2 since this is greater than the core-distance of p (e'=3mm). Adopted from [Ankerst99].

OPTICS Output: a reachability plot



K-Means

Implement your own code

UNIVERSITAT DE BARCELONA

K-Means basis

- It is a partitional algorithm that …
  - Assumes instances are **real-valued vectors**
  - Clusters based on ***centroids, center of gravity***, or **mean of points** in a cluster, ***c***:

$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

  - Reassignment of instances to clusters is **based on distance** to the current cluster centroids
    - Manhattan distance (L_1 norm), Euclidean distance (L_2 norm), Cosine similarity

Discussion on the K-Means method

- K-Means clustering often **terminates at a local optimal**
  - Initialization can be important to find high-quality clusters
- **Need to specify K**, the number of clusters, in advance
  - There are ways to automatically determine the "*best*" K
  - In practice, one often runs a range of values and selected the "*best*" K value
- **Sensitive to noisy data and outliers**
  - Variations: Using K-medians, K-medoids, etc.
- K-Means is applicable only to objects in a **continuous n-dimensional space**
  - Using the K-Modes for **categorical data**
- Non suitable to discover clusters with **non-convex shapes**
  - Using density-based clustering, kernel k-means, etc.

Variations of K-Means

- There are many variants of the K-Means methods, varying different aspects
  - Choosing better initial centroid estimates
    - K-Means++, Intelligent K-Means, Genetic K-Means
  - Choosing different representatives for the clusters
    - K-Medoids, K-Medians, K-Modes
  - Applying feature transformation techniques *(explained at the supervised part of the course)*
    - Weighted K-Means, Kernel K-Means

Initialization of K-Means

- Different initializations may generate rather different clustering results
- Original proposal (MacQueen,1967): selects the k seed randomly
  - Need to run the algorithm multiple times using different seeds
- There are many methods proposed for better initialization of K seeds
  - K-Means++ (Arthur and Vassilvitskii,2007):
    - The first centroid is selected randomly
    - The next centroid selected is the one that is farthest from the currently selected (selection is based on a weighted probability score).
    - The selection continues until K centroids are obtained

Some k-Means references

- MacQueen, J. B. (1967). **Some Methods for classification and Analysis of Multivariate Observations**. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297.

- Celebi, M. E., Kingravi, H. A., and Vela, P. A. (2013). **A comparative study of efficient initialization methods for the k-means clustering algorithm**. Expert Systems with Applications. 40 (1): 200–210.
- Arthur, D.; Vassilvitskii, S. (2007). **K-means++: the advantages of careful seeding**. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.

Note all the documents with this icon are in a zip file in campus virtual

K-Modes

K-Modes for categorical data

- K-Means cannot handle non-numerical (categorical) data
  - Mapping categorical value to 1/0 cannot generate quality clusters for high-dimensional data
- K-Modes is a variation of the *K-Means* Method (Huang'98)
  - Replacing means of clusters with <u>modes</u>
  - Using new dissimilarity measures to deal with categorical objects
  - Using a <u>frequency</u>-based method to update modes of clusters

K-Modes basis

- K-Modes: an extension to K-Means by replacing means with **modes**

$$\Phi(x_j, z_j) = 1 - n_j^r/n_l \text{ when } x_j = z_j \text{ ; } 1 \text{ when } x_j \neq z_j$$

where z_j is the categorical value of attribute j in Z_l, n_l is the number of objects in cluster l, and n_j^r is the number of objects whose attribute value is r

- Dissimilarity measure between object X and the center of a cluster Z
- The dissimilarity measure (distance function) is **frequency-based**

$$d(X_i, X_l) \equiv \sum_{j=1}^{m} \delta(x_{i,j}, x_{l,j})$$

where

$$\delta(x_{i,j}, x_{l,j}) = \begin{cases} 0, & x_{i,j} = x_{l,j} \\ 1, & x_{i,j} \neq x_{l,j} \end{cases}$$

K-Modes algorithm

- K-Modes deals with categorical attributes

```
Insert the first K objects into K new clusters.
Calculate the initial K modes for K clusters.
Repeat {
    For (each object O) {
        Calculate the similarity between object O and the
        modes of all clusters.
        Insert object O into the cluster C whose mode is the
        least dissimilar to object O.
    }
        Recalculate the cluster modes so that the cluster
        similarity between mode and objects is maximized.
} until (num_iterations or few objects change clusters).
```

K-Modes

- Algorithm is still based on iterative object cluster assignment and centroid update
- A **fuzzy k-modes** method is proposed to calculate a **fuzzy cluster membership** value for each object to each cluster
- A mixture of categorical and numerical data: Using a **K-prototype** method

References of K-Modes

- Zhexue Huang and Michael K. Ng. 2003. **A Note on K-Modes Clustering**. J. Classif. 20, 2 (September 2003), 257-261. DOI=http://dx.doi.org/10.1007/s00357-003-0014-4
- Anil Chaturvedi, Paul E. Green, and J. Douglas Caroll. 2001. **K-Modes Clustering**. J. Classif. 18, 1 (January 2001), 35-55. DOI=http://dx.doi.org/10.1007/s00357-001-0004-3
- Zengyou He, **Approximation algorithms for K-Modes clustering**. https://arxiv.org/pdf/cs/0603120.pdf
- Fuyuan Cao, Jive Liang, Deyu Li, Liang Bai, Chuangyin Dang. **A dissimilarity measure for the K-Modes clustering algorithm**. Knowledge-based Systems, Volume 26, 2012, ISSN 0950-7051.DOI= https://doi.org/10.1016/j.knosys.2011.07.011.

  http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.652.5571&rep=rep1&type=pdf

K-Medoids

K-Medoids basis

- The **k-Means algorithm is sensitive to outliers**!!
  - since an object with an extremely large value may substantially distort the distribution of the data
- **K-Medoids**:
  - Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located object** in a cluster

K-Medoids

- The K-Medoids clustering algorithm:
  - Select *K* points as the initial representative objects (i.e., as initial k-Medoids)
  - Repeat
    - Assigning each point to the cluster with the closest medoid
    - Randomly select a non-representative object o_i
    - Compute the total cost S of swapping the medoid m with o_i
    - If $S<0$, then swap m with o_i to form the new set of medoids

Discussion on K-Medoids Clustering

- K-Medoids Clustering: find representative objects (medoids) in clusters
- PAM (Partitioning Around Medoids)
  - Starts from an initial set of medoids, and
  - Iteratively replaces one of the medoids by one of the non-medoids if it improves the total sum of the squared errors (SSE) of the resulting clustering
  - PAM works effectively for small data sets but does not scale well for large data sets (due to the computational complexity)
  - Computational Complexity: PAM $O(K(n-K)^2)$ (quite expensive!)
- Efficiency improvements on PAM
  - CLARA (Kaufmann & Rousseeuw, 1987)
    - PAM on samples; $O(Ks^2 + K(n-k))$, s is the sample size
  - CLARANS (ng & Han, 1994): Randomized re-sampling, ensuring efficiency + quality

References K-Medoids

- R. T. Ng and Jiawei Han (2002), "**CLARANS: a method for clustering objects for spatial data mining**" in *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003-1016, Sep/Oct 2002. doi: 10.1109/TKDE.2002.1033770

- Kaufman, L. and Rousseeuw, P.J. (1987), **Clustering by means of Medoids**, in Statistical Data Analysis Based on the L_1 –Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416

- H.S. Park , C.H. Jun, **A simple and fast algorithm for K-medoids clustering**, Expert Systems with Applications, 36, (2) (2009), 3336–3341

- J. Xie and S. Jiang, "**A Simple and Fast Algorithm for Global K-means Clustering**", 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, 2010, pp. 36-40. doi: 10.1109/ETCS.2010.347

K-Prototypes

UNIVERSITAT DE BARCELONA

K-prototypes Algorithm

- To integrate the k-means and k-modes algorithms into the k-prototypes algorithm that is used to cluster the mixed-type objects

- The dissimilarity between two mixed-type objects X and Y, which are described by attributes $A_1^r, A_2^r, \ldots., A_p^r, A_{p+1}^c, \ldots, A_m^c$ (*m* is the attribute numbers the first *p* means numeric data, the rest means categorical data), can be measured by::

$$d_2(X,Y) = \sum_{j=1}^{p} (x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m} \delta(x_j, y_j)$$

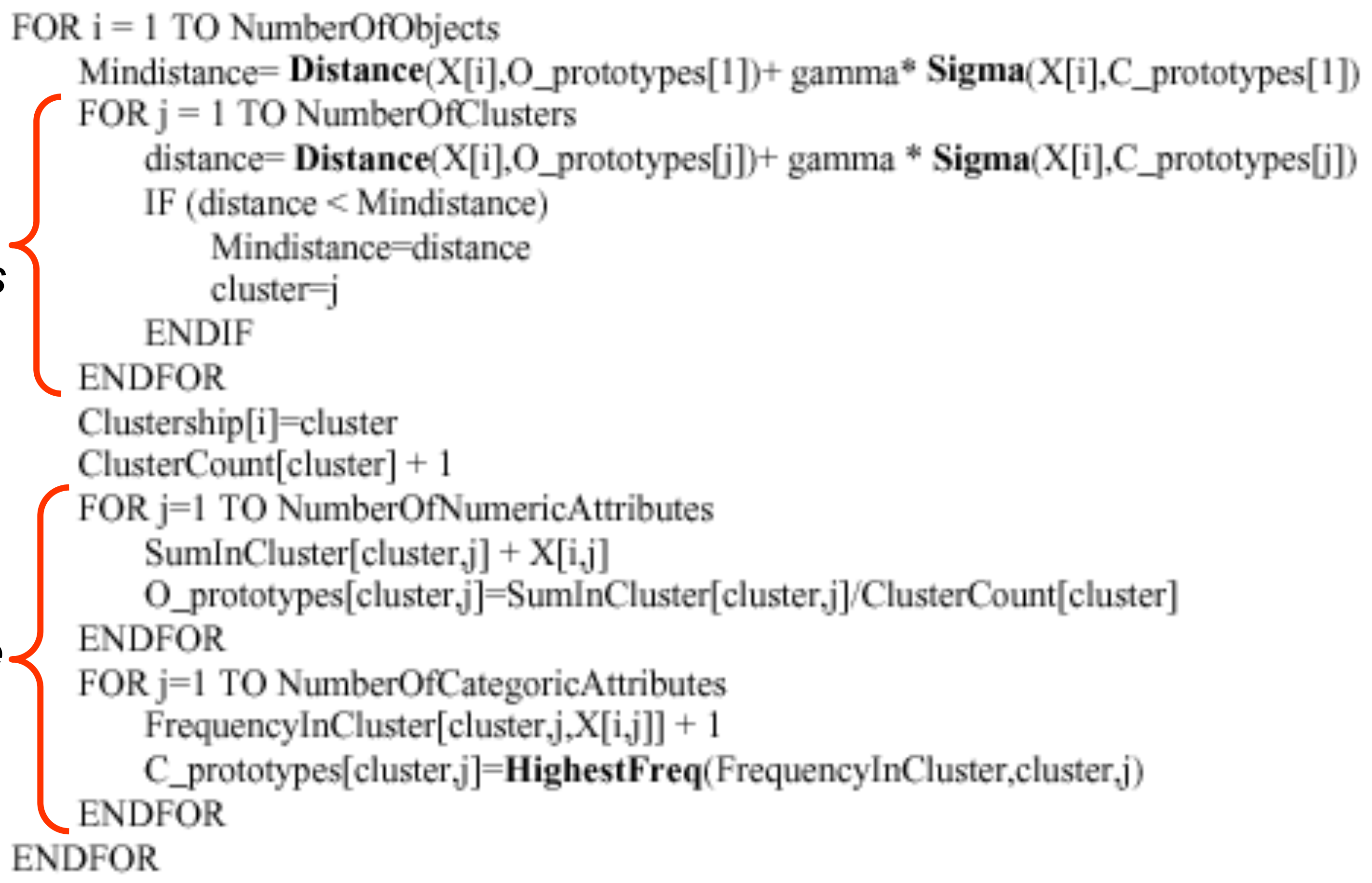
UNIVERSITAT DE BARCELONA

K-prototypes Algorithm(cont.)

$$d_2(X,Y) = \sum_{j=1}^{p}(x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m}\delta(x_j, y_j)$$

- The first term is the Euclidean distance measure on the numeric attributes and the second term is the simple matching dissimilarity measure on the categorical attributes
- The weight γ is used to avoid favoring either type of attribute

K-prototypes Algorithm(cont.)



```
FOR i = 1 TO NumberOfObjects
    Mindistance= Distance(X[i],O_prototypes[1])+ gamma* Sigma(X[i],C_prototypes[1])
```

Choose clusters

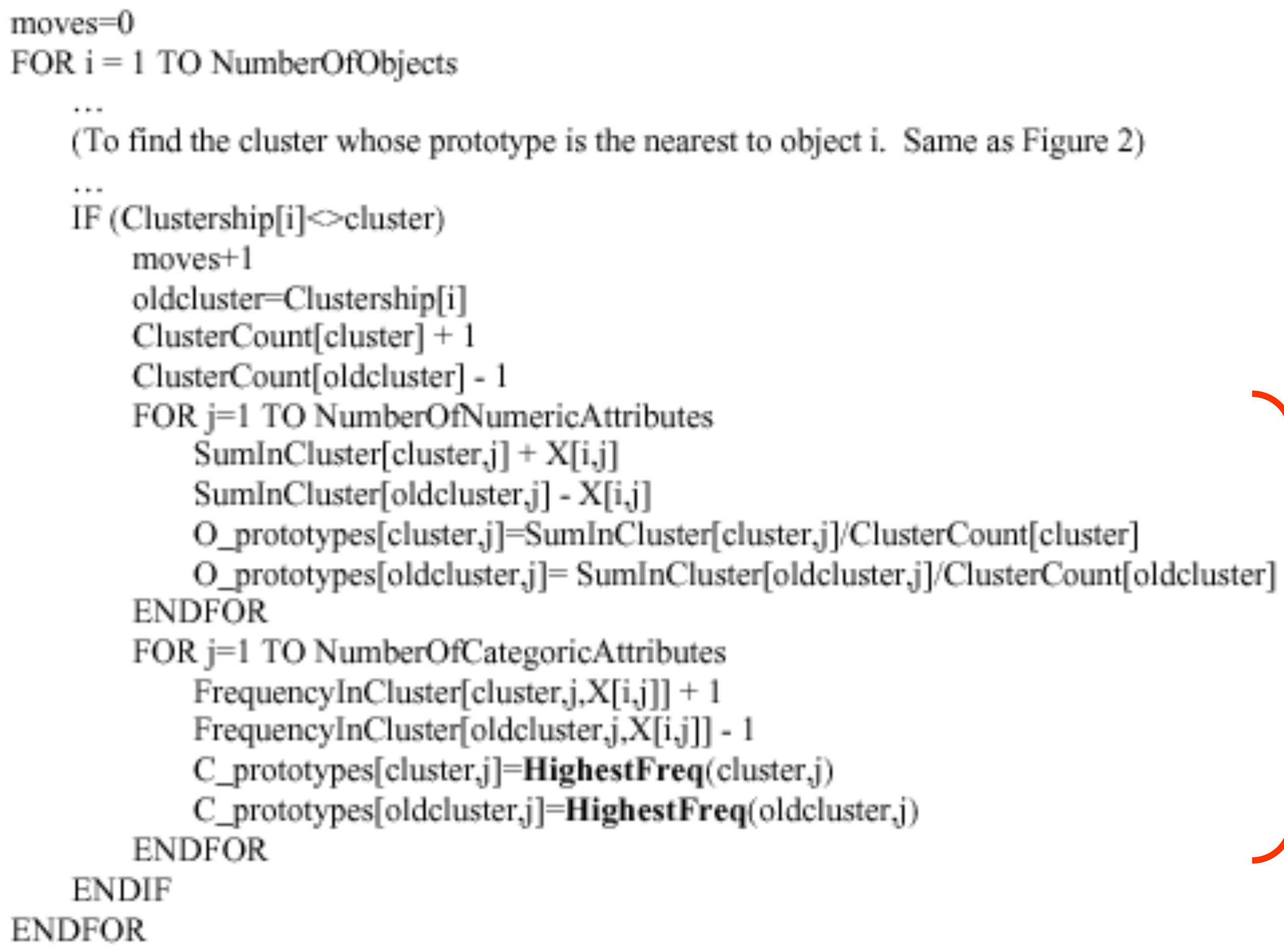
```
    FOR j = 1 TO NumberOfClusters
        distance= Distance(X[i],O_prototypes[j])+ gamma * Sigma(X[i],C_prototypes[j])
        IF (distance < Mindistance)
            Mindistance=distance
            cluster=j
        ENDIF
    ENDFOR
    Clustership[i]=cluster
    ClusterCount[cluster] + 1
```

Modify the mode

```
    FOR j=1 TO NumberOfNumericAttributes
        SumInCluster[cluster,j] + X[i,j]
        O_prototypes[cluster,j]=SumInCluster[cluster,j]/ClusterCount[cluster]
    ENDFOR
    FOR j=1 TO NumberOfCategoricAttributes
        FrequencyInCluster[cluster,j,X[i,j]] + 1
        C_prototypes[cluster,j]=HighestFreq(FrequencyInCluster,cluster,j)
    ENDFOR
ENDFOR
```

Figure 2. Initial allocation process.

K-prototypes Algorithm(cont.)



Modify the mode

```
moves=0
FOR i = 1 TO NumberOfObjects
    ...
    (To find the cluster whose prototype is the nearest to object i.  Same as Figure 2)
    ...
    IF (Clustership[i]<>cluster)
        moves+1
        oldcluster=Clustership[i]
        ClusterCount[cluster] + 1
        ClusterCount[oldcluster] - 1
        FOR j=1 TO NumberOfNumericAttributes
            SumInCluster[cluster,j] + X[i,j]
            SumInCluster[oldcluster,j] - X[i,j]
            O_prototypes[cluster,j]=SumInCluster[cluster,j]/ClusterCount[cluster]
            O_prototypes[oldcluster,j]= SumInCluster[oldcluster,j]/ClusterCount[oldcluster]
        ENDFOR
        FOR j=1 TO NumberOfCategoricAttributes
            FrequencyInCluster[cluster,j,X[i,j]] + 1
            FrequencyInCluster[oldcluster,j,X[i,j]] - 1
            C_prototypes[cluster,j]=HighestFreq(cluster,j)
            C_prototypes[oldcluster,j]=HighestFreq(oldcluster,j)
        ENDFOR
    ENDIF
ENDFOR
```

Figure 3. Reallocation process.

References of K-prototypes

- Zhexue Huang, **Clustering large datasets with mixed numerical and categorical values**. https://pdfs.semanticscholar.org/d42b/b5ad2d03be6d8fefa63d25d02c0711d19728.pdf
- Byoungwook Kim. **A Fast K-prototypes Algorithm Using Partial Distance Computation**. https://www.researchgate.net/publication/316348009_A_Fast_K-prototypes_Algorithm_Using_Partial_Distance_Computation

UNIVERSITAT DE BARCELONA

Course. Introduction to Machine Learning
Work 1. Clustering Exercise

Session 2

Course 2021-2022

Dr. Maria Salamó Llorente

Dept. Mathematics and Informatics,

Faculty of Mathematics and Informatics,

University of Barcelona