Unsupervised Learning
- Part 2 -

Road Map

- K-Medoids, k-modes and k-means++
- FastMap: multidimensional scaling
- Cluster evaluation
- Clusters and holes
- Fuzzy clustering: fuzzy C-means
- Summary

k-medoids

- The algorithms in this category are similar with k-means.
- The main differences from k-means are:
  - K-medoids uses a data point as center of a cluster (such a point is called a medoid). This is the cluster member best approximating the cluster center.
  - Stopping criterion is based not on SSD but on sum of pairwise dissimilarities (distances).
- The best known algorithm of this type is Partitioning Around Medoids (PAM)

PAM

Input:

- A dataset D = {P_1, P_2, …, P_m} containing **m** points in an **n**-dimensional space and a distance function between points in that space.
- **k**: the number of clusters to be obtained

Output:

- The k clusters obtained

PAM - Method

1. Randomly choose k points in D as initial medoids: {m_1, m_2, …, m_k}
2. REPEAT
3. FOR (i=1; i<=m; i++)
4. Assign P_i to the nearest medoid
5. END FOR
6. FOR (i=1; i<=k; i++)
7. FOR (j=1; j<=m; j++)
8. IF Pj is not a medoid THEN
9. Configuration(i, j) = swap P_j with m_i
10. Compute the cost of the new configuration
11. Reverse the swap
12. END IF
13 END FOR
14 END FOR
15. Select the configuration with the best cost (lowest)
16. UNTIL New configuration = Old configuration

Configuration cost

- The main idea is that each medoid may be swapped with any non-medoid point.
- If the new configuration is the best swap, a new medoid is appointed replacing an old one.
- The process continues until no better configuration is possible.
- The cost of a configuration is the sum of the distances between points and their medoids:

$$\text{cost} = \sum_{i=1}^{m} \text{Dist}(P_i - m_i)$$

k-modes

- k-modes is designed to be used for points having categorical (nominal or ordinal) attributes.
- The mode of a dataset is the most frequent value.
- This refers to a dataset containing atomic values.
- In clustering a point is characterized by a set of attributes, in some cases of different types, each attribute having a value from its domain.

k-modes

- In that case we must redefine the mode for applying the notion to a set of points.
- The definition starts with the expression returning the number of dissimilarities (like in the previous course) between two points X and Y in an n-dimensional space:

$$\mathbf{d(X,Y)} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \delta(\mathbf{x_i}, \mathbf{y_i})$$

where $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$ and $\delta(x_i, y_i) = 1$ if $x_i = y_i$ and 0 otherwise

The mode

- If $D = \{P_1, P_2, \ldots, P_m\}$ is a set containing **m** points with **n** attributes (categorical or not), the mode of D may be defined as a vector (with the same number of dimensions) $Q = (q_1, q_2, \ldots, q_n)$ that minimizes:

$$S(D, Q) = \sum_{i=1}^{m} d(P_i, Q)$$

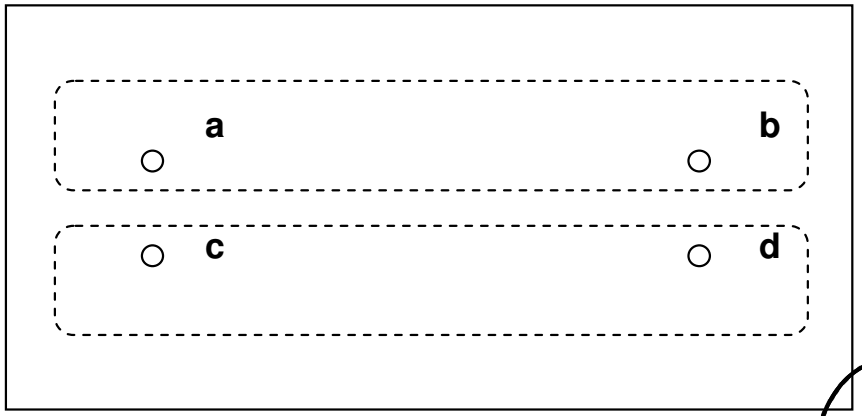
- Q is not necessarily a member of D. The mode of a set is not unique. For example, the mode of [a, b], [a, c], [c, b], and [b, c] is either [a, b] or [a, c].

k-means vs. k-modes

- The differences between k-means and k-modes are listed in the initial article ([Huamg 98]):
  1. Uses of a simple matching dissimilarity measure for categorical objects,
  2. Replaces means of clusters by modes, and
  3. Uses a frequency-based method for finding the modes.

K-means++

- One of the problems of k-means is that the algorithm is sensitive to the initial centroids.
- A bad choice may lead to bad clustering results, as in figure 1: if **a** and **c** are chosen for initial centroids the result is not the natural one:



k-means++

- K-means++ is not a new clustering algorithm but a method to select initial centroids:

1. The first centroid is selected randomly from the data points.
2. For each data point P, compute **d** = Dist(P, c), the distance between P and the nearest centroid already determined.
3. A new centroid is selected using a weighted probability distribution: the point is chosen with a probability proportional to d^2.
4. Repeat steps 2 and 3 until k centroids are selected.

- After initial centroid selection, usual k-means algorithm may be run for clustering the dataset.

Road Map

- K-Medoids, k-modes and k-means++
- FastMap: multidimensional scaling
- Cluster evaluation
- Clusters and holes
- Fuzzy clustering: fuzzy C-means
- Summary

FastMap

- There are cases when there is no Euclidean space and only the distances between two points are available (given as input or by a distance function specific to the dataset).
- In that case all the algorithms assuming the existence of coordinates and of a Euclidean space cannot be used.
- This paragraph presents a solution for solving the above problem: associate a Euclidian space with few dimensions with the dataset points.

FastMap

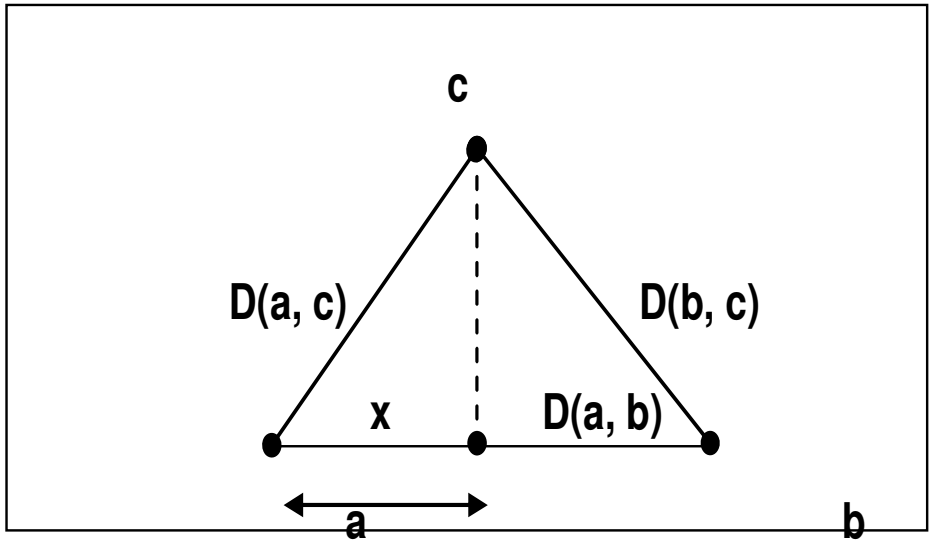
- The process of creating a Euclidian space knowing only the distances between any two points is called multidimensional scaling
- There are many algorithms for this, the most known being FastMap, MetricMap, and Landmark MDS (LMDS).
- These algorithms approximate classical MDS using a subset of the data and fitting the remainder to the solution.

FastMap

- FastMap is a recursive algorithm and at each step the following operations are done:
- Two distant points are selected as an axis. In the next figure these points are **a** and **b**.
- For every point c compute the coordinate **x** for this axis using the generalized Pythagoras theorem (also known as the law of cosine):

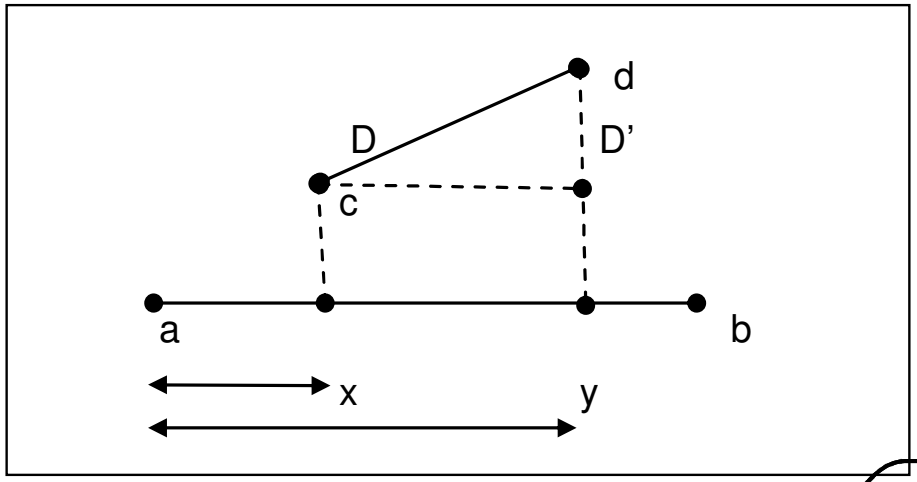
$$x = (D^2(a, c) + D^2(a, b) - D^2(b, c)) / (2 * D(a, b))$$

$$x = (D^2(a, c) + D^2(a, b) - D^2(b, c)) / (2 \cdot D(a, b))$$



$D'^2 = D^2 - (x - y)^2$

- Use for further axis not the original distances D between points but the remainder D' after subtracting the distance with respect with the coordinates already computed.



Weaknesses

- This process stops after computing the desired number of coordinates for every point or no more axes can be found.

Weakness:

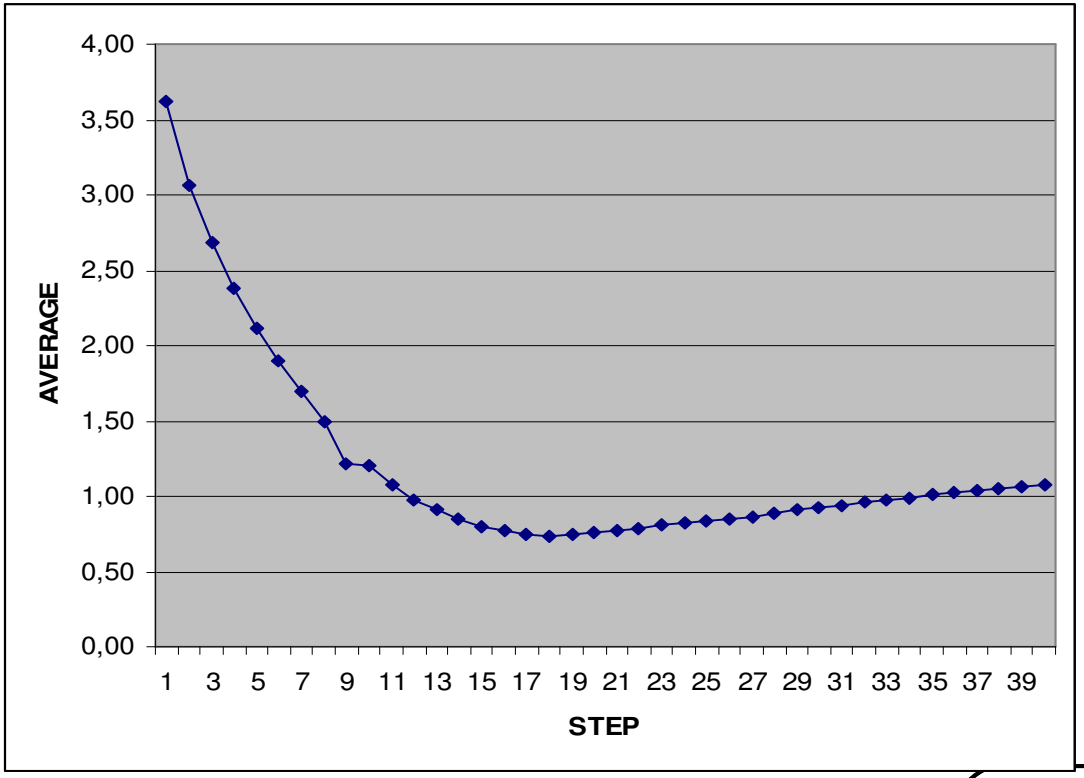
- For real data the problem is that the distance matrix is not a Euclidian one: the value for D'^2 may be negative value for.
- In that case the only way to continue is to assume D' is 0.
- But this assumption leads to propagated errors in computing real good coordinates for our purpose.

Experiments

- The algorithm was run with this assumption on a 2000 nodes matrix.
- The optimal number of axes was 18 (enough big value)
- The average of the differences between real distances and computed distances (computed from the resulting coordinates) was high: 0.78 after 18 steps, as shown in next figure.

Experiments

- Average of | Dreal – Dcomputed | over all pairs of nodes



Road Map

- K-Medoids, k-modes and k-means++
- FastMap: multidimensional scaling
- Cluster evaluation
- Clusters and holes
- Fuzzy clustering: fuzzy C-means
- Summary

Cluster evaluation

- After performing the clustering process, the result must be evaluated in order to validate it (or not).
- Because real clusters are not known for a test dataset, this is a hard problem.
- Some methods were developed for this purpose.
- These methods are designed not for evaluating the clustering results on a particular dataset but for evaluating the quality of the clustering algorithm.

Methods

Most used methods:

- User inspection
- Ground truth
- Cohesion and separation
- Indirect evaluation

User inspection

- In that case some experts are inspecting the results of the clustering algorithm and rate it.
- User inspection may include:
  - Evaluate cluster centroids
  - Evaluate distribution of points in clusters
  - Evaluate clusters by their representation (sometimes clusters may be represented as a decision tree for example).
  - Test some points to see if they really belong to the assigned cluster. This can be made when clustering documents: after clustering, some documents in each cluster are analyzed to see if they are in the same category.
- This method is hard to use for numerical data and huge volumes of information because the user inspection is based on the experience and intuition of the experts.
- Also, this method is subjective and may lead sometimes to a wrong verdict.

Ground truth

- In this case the input of the clustering algorithm is a labeled dataset.
- In this way we know in advance the cluster for each point and after running the clustering algorithm we can compare the real clusters with the results obtained.
- Evaluation may be made using some measures, as entropy and purity (known from previous chapter – supervised learning)

Entropy

- Remember that if we have a dataset D = {e_1, e_2, …, e_m} with examples labeled with classes from C = {c_1, c_2, …, c_n}, the entropy of D can be computed as:

$$\mathbf{entropy(D)} = -\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{Pr(c_i)\log_2 Pr\,(c_i)}$$

- After clustering, D is split in r disjoint subsets D_1, D_2, …, D_r. the combined entropy of these subsets is:

$$\mathbf{entropy(D, A_k)} = \sum_{\mathbf{i=1}}^{\mathbf{r}} \frac{\mathbf{count(D_i)}}{\mathbf{count(D)}} * \mathbf{entropy(D_i)}$$

Purity

- For each cluster, the purity of the cluster is the probability of the most present class:

$$\text{Purity}(D_i) = \max_j(\text{Pr}_i(c_j))$$

- We can compute a purity of the clustering process by combining the purities of resulting clusters:

$$\text{Purity}(D) = \sum_{i=1}^{r} \frac{|D_i|}{|D|} * \text{Purity}(D_i)$$

Ground truth

- These measures are usually used when comparing two clustering algorithms on the same labeled dataset.
- Other measures that can be used are precision, recall and F-score. The expressions for these measures were also presented in the previous chapter.
- The real problem is that an algorithm may perform well on a dataset and not so well on other dataset.

Cohesion and separation

- Other measures that can be used to evaluate the clustering algorithm are based on internal information:

1. Intra-cluster cohesion measures the compactness of the clusters.

- Using the sum of squares of the distances (SSD) from each point to its cluster we obtain a measure of this cohesion.

$$\mathbf{SSD} = \sum_{i=1}^{k} \sum_{\mathbf{p} \in \mathbf{Cluster\_i}} \mathbf{Dist}(\mathbf{p}, \mathbf{Centroid}(\mathbf{Cluster\_i})$$

- A small value is better than a bigger one.

Cohesion and separation

2. Inter-cluster separation (or isolation) measures how far are the clusters one from another.

- The distance between clusters may be computed in the known ways (single link, complete link, etc)

Indirect evaluation

- In many cases clustering is made in order to perform another task.
- Example: customers are grouped based on their buying habits for email marketing.
- If the primary goal (email marketing in our example) has no good results, it means that maybe the clustering was not so good.
- In this way a clustering algorithm can be rated based on another task results.

Road Map

- K-Medoids, k-modes and k-means++
- FastMap: multidimensional scaling
- Cluster evaluation
- Clusters and holes
- Fuzzy clustering: fuzzy C-means
- Summary

Clusters and holes

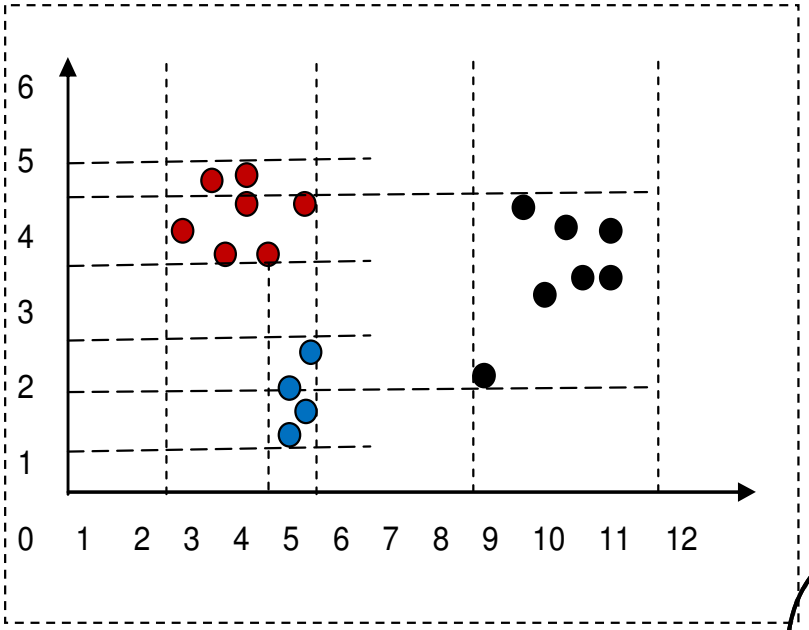
- The space between the clusters is empty.
- This portion of the space, containing no or very few points may be called hole.
- Discovering holes is sometimes very useful because a position in a hole says that a certain combination of attributes values is not possible (or the possibility of having that combination is very low).

Clusters and holes

- Example: in a 2D space with dimensions blood pressure and cholesterol level a hole defines a range of pairs (blood-pressure, cholesterol-level) with a zero or low probability of occurrence.
- There are several techniques for discovering regions that may be considered as holes.
- This paragraph presents two of them.

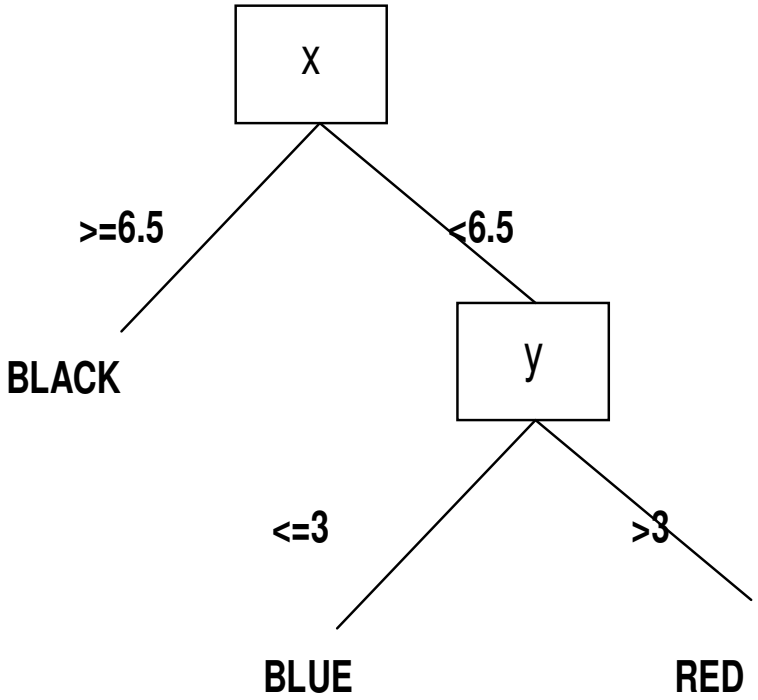
Decision tree clusters

- One of the possible representations of a set of clusters is a decision tree.
- Example:



Decision tree clusters

- The decision tree is:

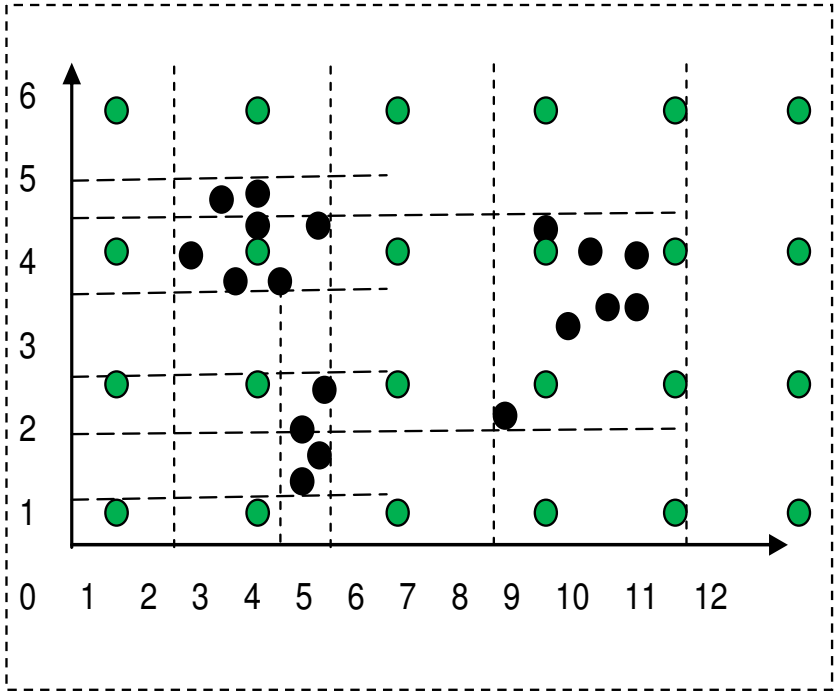


Decision tree clusters

- Because in holes there is no point, for using supervised learning to discover the holes the trick is the following:
  - Consider all points having the same class (called existing points, E)
  - Add uniformly another type of points (called non-existing points, N)
- Next figure is an illustration of this method

E and N points

- Green: N points, Black: E points



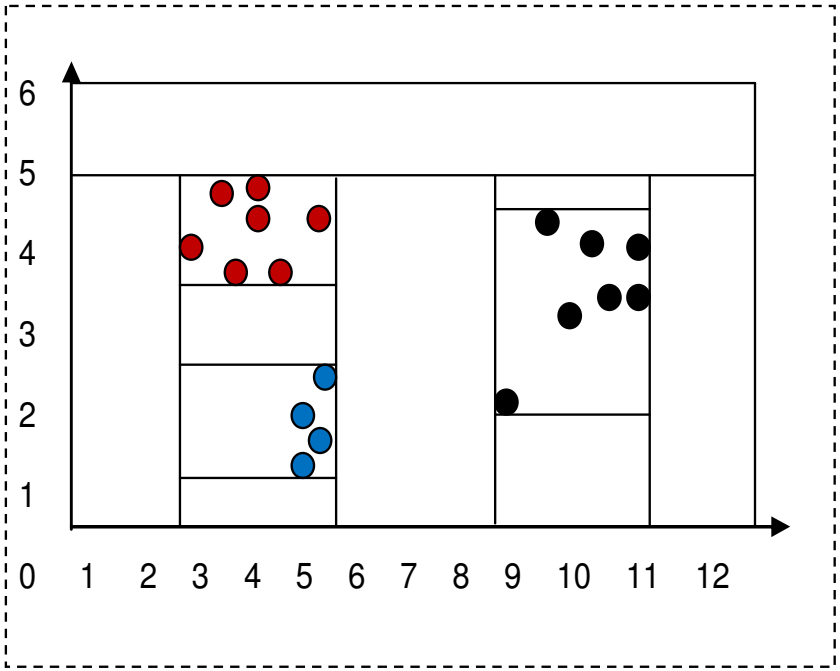
Processing

- A supervised learning algorithm can be used for building a decision tree for separating the two types of points: existing and non-existing.
- The decision tree is built using the best cut for each axis, and this best cut is based on the information gain.
- Because for computing the information gain only the probability for each type of points in a given region is needed, the non-existing points need not to be physically added but because their uniform spread the probability is proportional with the area of that region.

Processing

- For the existing points, the probability for each sub-region is computed by counting, as usual.
- The algorithm assumes that all regions are rectangular and the number of N points in each region is at least equal with the number of E points.
- After each split of a rectangle, if the inherited number of N points is less than the number of E points, their number is increased to the number of E points.
- The result is a decision tree splitting the space in rectangles, some of them being clusters and the others holes.

Result



Maximal hyper rectangles

- This is the second approach.
- The goal is to find the maximal hyper rectangles containing no or few data points.
- Note that the clusters in the previous example are contained in three rectangles:
  - Cluster 1 (red): x >= 2, x <=5 , y>=3.5, y<=5
  - Cluster 2 (black) : x>=8, x<=11, y>=2, y<=4.5
  - Cluster 3 (blue): x>=4, x<=5, y>=0.5, y<=2.5

FR and MHR

- Such a rectangle is called a ***filled region*** (FR).
- A maximal hyper rectangle is defined as follows:

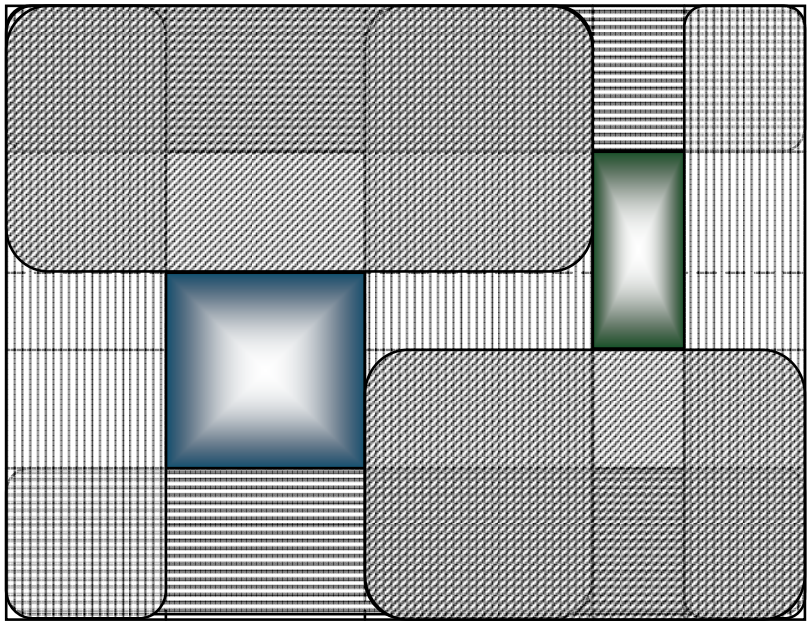
Definition: Given a k-dimensional continuous space S and n FRs in S, a ***maximal hyper-rectangle*** (MHR) in S is an empty HR that does not intersect (in a normal sense) with any FR, and has at least one FR lying on each of its 2k bounding surfaces. These FRs are called the ***bounding*** FRs of the MHR.

Algorithm

1. Let S be a k-dimensional continuous space and a set of **n** FRs (not always disjoint) in S,
2. Start with one MHR, occupying the entire space S.
3. Each FR is incrementally added to S. For each insertion, set of MHRs is updated:
   - All the existing MHRs that intersect with this FR must be removed from the set.
   - For each dimension two new hyper-rectangle bounds (lower and upper) are identified. If the new hyper-rectangles verify the MHR definition and are sufficiently large, insert them into the MHRs list.

Example

- Addition of a second FR (H2 after H1):



Road Map

- K-Medoids, k-modes and k-means++
- FastMap: multidimensional scaling
- Cluster evaluation
- Clusters and holes
- Fuzzy clustering: fuzzy C-means
- Summary

Fuzzy clustering

- In the case of soft clustering, any point belongs to more than one cluster and for each (point, cluster) pair there is a value of the membership level of that point to that cluster.
- One of the most known algorithms in this class is fuzzy C-means.

The model

Input:

- A dataset containing **n** elements (points), D = $\{e_1, e_2, \ldots, e_n\}$.
- The number of clusters **C**
- A level of cluster fuzziness, **m**

Output:

- A list of centroids $\{c_1, c_2, \ldots, c_C\}$
- A matrix **U** = $[u_{ij}]$, I = 1…n, j = 1…C, and u_{ij} = the level/degree of membership of element e_i to the cluster c_j.

The model

- The process is trying to minimize the objective function:

$$J(U, c_1, c_2, \ldots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2$$

where:

- ✓ u_{ij} and c_i are as described above.
- ✓ d_{ij} is the distance from the element e_i to the centroid c_j
- ✓ m is the fuzziness factor and in many cases the default value is 2.

If m is close or equal to 1, u_{ij} is close to 0 or 1 so a non-fuzzy solution is obtained (as in k-means). When m is increased from 2 to bigger values, u_{ij} have lower values and the clusters are fuzzier.

The algorithm

1. Close randomly initial cluster centers
2. REPEAT
3. Compute all d_{ij} values

$$u_{ij} = \frac{1}{\sum_{k=1..C} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}$$

4. Compute new values for the membership levels u_{ij}:
5. Compute new cluster centers cj:

$$c_j = \frac{\sum_{i=1..n} u_{ij}^m \cdot x_i}{\sum_{i=1..n} u_{ij}^m}$$

1. UNTIL (stopping criteria is met)

Stoping criteria

- The stopping criteria may include:

1. The number of iterations reached a given value.
2. The cluster centers movement after iteration **i** is below a certain threshold t:

$$\sum_{j=1..C} \left| C_j(i) - C_j(i-1) \right| < t$$

Summary

This course presented:

- K-Medoids, k-modes and k-means++ where k-medoids and k-modes are clustering algorithms and k-means++ is a method for determining a better than random set of initial cluster centers for k-means.
- FastMap: a multidimensional scaling algorithm to build a Euclidean space given the distances between any two points
- Cluster evaluation techniques. A method not included in this course but still important is silhouette (see [Rousseeuw 87])
- Clusters and holes: how to determine regions with no or few data points
- Fuzzy clustering and fuzzy C-means for performing soft clustering.

- Next week: Semi-supervised learning

References

- [Liu et al. 98] Bing Liu, Ke Wang, Lai-Fun Mun and Xin-Zhi Qi, "Using Decision Tree Induction for Discovering Holes in Data," Pacific Rim International Conference on Artificial Intelligence (PRICAI-98), 1998
- [Liu et al. 00] Bing Liu, Yiyuan Xia, Phlip S. Yu. "Clustering through decision tree construction." Proceedings of 2000 ACM CIKM International Conference on Information and Knowledge Management (ACM CIKM-2000), Washington, DC, USA, November 6-11, 2000
- [Liu 11] Bing Liu, 2011. Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data, Second Edition, Springer, chapter 4.
- [Huang 98] ZHEXUE HUANG, Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values, DMKD 2, 1998,
- http://www.cs.ust.hk/~qyang/Teaching/537/Papers/huang98extensions.pdf
- [Torgerson 52] Torgerson, W.S. (1952). Multidimensional Scaling: Theory and Method, Psychometrika, vol 17, pp. 401-419.
- [Faloustsos, Lin 95] Faloutsos, C., Lin K.I. (1995). FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data.

References

- [Wand et al, 99] Wang, J.T-L., Wang, X., Lin, K-I., Shasha, D., Shapiro, B.A., Zhang, K. (1999). Evaluating a class of distance-mapping algorithms for data mining and clustering, In: Proc of ACM KDD, pp. 307-311.
- [de Silva, Tenenbaum 04]de Silva, V., Tenenbaum J.B. (2004). Sparse multi-dimensional scaling using landmark points,
- http://pages.pomona.edu/~vds04747/public/publications.html, Manuscript, June 2004.
- [Yang et al 06] Yang, T., Liu, J., McMillan, L., Wang, W., (2006). A Fast Approximation to Multidimensional Scaling, In: Proceedings of the ECCV Workshop on Computation Intensive Methods for Computer Vision (CIMCV).
- [Platt 05] Platt, J.C., (2005). FastMap, MetricMap, and Landmark MDS are all Nyström Algorithms, In: 10th International Workshop on Artificial Intelligence and Statistics, pp. 261-268.
- [Bezdek 81] Bezdek, James C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers Norwell, MA, USA, ISBN 0-306-40671-3
- [Rousseeuw 87] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics 20: 53–65