Data Clustering

Nicolas PASQUIER

Laboratoire I3S (UMR-7271 UNS/CNRS)

Université Nice Sophia-Antipolis

http://www.i3s.unice.fr/~pasquier

mailto:nicolas.pasquier@unice.fr

Objectives of the Course

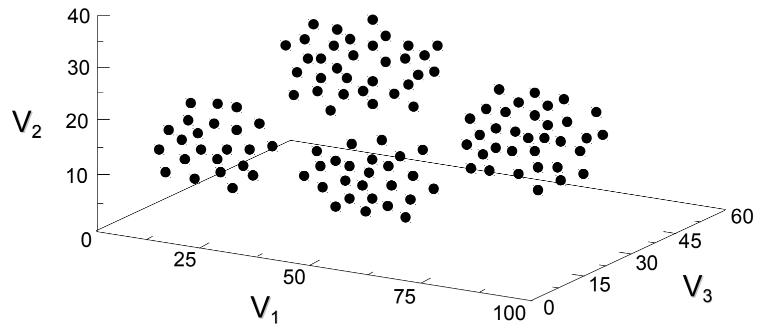
- Understand what is unsupervised classification
- Connect to the concept of data structure discovery
- Learn how to
  - Define a multi-dimensional data space for data clustering
  - Define a similarity measure in this multi-dimensional data space
  - Choose a relevant clustering algorithm regarding multi-dimensional data in input
  - Define an adequate parametrization for the chosen algorithm
- Comprehend the central notion of similarity measure and relate to the mathematical notion of distance measure
- Understand algorithmic approaches: Partitioning, hierarchical, density based, grid based, model based, and ensemble clustering

What is Clustering?

- A cluster (group) is a set of individuals (e.g. instances) or features (e.g. variables) that are
  - Similar among themselves within the group
  - Different from one group to another
- Clustering is the process of classifying individuals or objects in different groups
- Unsupervised context: No target variable or classes
- Thus, no prior knowledge of the number and type of “natural” clusters in the data space
- Subjective process that aims to discover inherent data structures in the data space to reveal coherent data groups
- Terminology: Segmentation, unsupervised learning, data partitioning

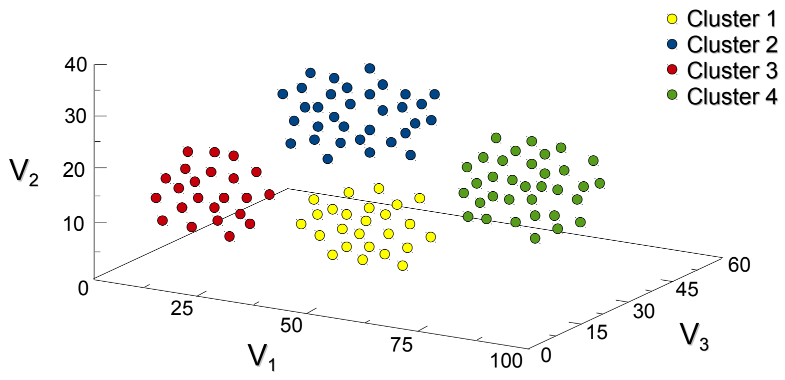
Example: Multi-dimensional Data Space

- Tri-dimensional data space: Dimensions are variables V_1, V_2, V_3
- Each dataset instance is represented as a dot



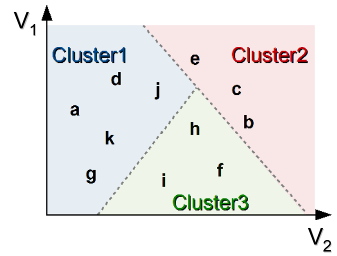
Example: Data Clustering

- Four “natural” clusters correspond each to a dense region of the data space separated by sparsely populated regions

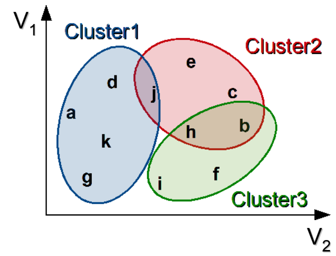


Data Clustering Typologies

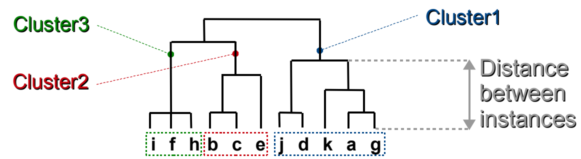
Exclusive clusters (partitions)



Overlapping clusters (fuzzy)



Hierarchical clustering (dendrogram)



Application Examples

- Fields of application: Astronomy, biology (bioinformatics, biodiversity, medicine, …), geography, geology, marketing, sociology, zoology, etc.
- Customer Relationship Management (CRM) segmentation
  - Distinguish segments of customers (similar behaviors)
  - Characterization of segments according to their purchasing behaviors
- Bioinformatics
  - Identify genes (genomics) and proteins (proteomics) participating to the same biological functions or processes
- Medical imaging
  - Differentiation between different tissue types

Application Examples

- Social network analysis
  - Identification and characterization of users communities according to their centers of interest or their opinions
- Spatial data analysis
  - Automatic data acquisition problems of volume (satellite images, medical equipment, etc.)
  - Identification of geographical areas with similar properties (e.g. climate, crops, habitats)
- Web Mining
  - Identify corpus or collections of web-based documents addressing the same topics

What is a Good Clustering?

- Assessing the quality of the discovered groups
  - Minimize intra-clusters variability (i.e. high similarity of individuals within clusters)
  - Maximize inter-clusters variability (i.e. low similarity of individuals between clusters)
- The similarity between two instances is assessed by comparing variable values of the instances, to calculate a distance between them
- The quality of the clustering result will depend on
  - The distance measure used
  - The algorithm configuration chosen to implement it

Distance Measure Definition

- Let X and Y be two vectors (instances), a function d() is a distance measure if and only if d(X,Y) satisfies the following properties (Anderberg, 1973)
  1. Non-negative: $d(X,Y) \geq 0$
  2. Reflexive: $d(X,X) = d(Y,Y) = 0$
  3. Commutative: $d(X,Y) = d(Y,X)$
  4. Triangular inequality: $d(X,Y) \leq d(X,W) + d(W,Y)$
- The definition of distance functions depends on the type of variables in the data (numerical, binary, etc.)
- It is difficult to define the notion of "sufficiently similar" to include two instances within the same group: There is typically a part of subjectivity in the decision

Data Structures

- Clustering algorithms receive as input a data matrix or a distance matrix (computed from the data matrix)
- Let D be a dataset of p variables and n instances
- Data matrix

$$\begin{bmatrix} V_{11} & \dots & V_{1f} & \dots & V_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ V_{i1} & \dots & V_{if} & \dots & V_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ V_{n1} & \dots & V_{nf} & \dots & V_{np} \end{bmatrix}$$

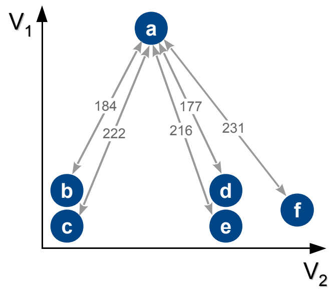
- Distance matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \dots & \dots & \dots & \dots & \\ d(n,1) & d(n,2) & \dots & d(n,n-1) & 0 \end{bmatrix}$$

Distance Matrix: Example

- Example bi-dimensional dataset of six instances D = {a, b, c, d, e, f} with numerical dimensions V_1 and V_2
- Compute the Euclidean distance measure for each pair of instances

Data space



Euclidean distance matrix

	a	b	c	d	e	f
a	0					
b	184	0				
c	222	45	0			
d	177	123	129	0		
e	216	128	121	46	0	
f	231	203	201	83	81	0

Types of Variables

- The distance measure is defined according to the variable types (semantics, not encoding)
- Numerical: Continuous values
  - Ex: Temperature ∈ ℤ, age ∈ ℕ, speed ∈ ℝ
  - 0-linear scales are peculiar cases: Exponential $\beta.e^{(\alpha.v)}$, logarithmic $\beta.\log(\alpha.n)$
- Binary: Two possible values
  - Ex: Gender ∈ {M, F}, married ∈ {true, false}, active ∈ {0, 1}
- Categorical (nominal): List of possible discrete values
  - Ex: Color ∈ {blue, green, …}, dept. number ∈ [01, 95]
- Ordinal: List of possible discrete ordered values
  - Ex: Humidity ∈ {low, medium, high}, ranking ∈ {1st, 2nd, 3rd, …}

Continuous Numerical Variables

- The most popular distance measure is the Minkowski distance
- Let X and Y be two instances: $X = \{x_1, x_2, \ldots, x_p\}$, $Y = \{y_1, y_2, \ldots, y_p\}$
- Minkowski distance is the generalization of Euclidean distance:

$$d(X,Y) = \sqrt[q]{|x_1-y_1|^q+|x_2-y_2|^q+\ldots+|x_p-y_p|^q}$$

where q is a positive 0-null integer

- Euclidean distance: q = 2

$$d(X,Y) = \sqrt{|x_1-y_1|^2+|x_2-y_2|^2+\ldots+|x_p-y_p|^2}$$

- Manhattan distance: q = 1

$$d(X,Y) = |x_1-y_1|+|x_2-y_2|+\ldots+|x_p-y_p|$$

Continuous Numerical Variables

- A weighted distance measure can be used to adapt the importance of each variable: Weight vector $W = \{w_1, w_2, \ldots, w_p\}$

$$d(X,Y) = \sqrt[q]{w_1|x_1-y_1|^q+w_2|x_2-y_2|^q+\ldots+w_p|x_p-y_p|^q}$$

- The Mahalanobis distance is a weighted distance measure that can be useful for outlier (exception) detection
- It can be defined as the measure of dissimilarity between two random vectors of the same distribution with covariance matrix Σ

$$d(X,Y) = \sqrt{(X-Y)^T \Sigma^{-1} (X-Y)}$$

- If the covariance matrix is the identity matrix, then this distance is the same as the Euclidean distance
- Mahalanobis distance gives less weight to the most noisy variables (assuming that each is a Gaussian random variable)

Data Normalization

- Important differences in the scales of variable values requires to normalize the variables, to equalize their influence on the process
- Calculation of z-scores: Measurements that are normalized by mean and deviation measures
- The data matrix is replaced by the z-score matrix for the clustering process
- Example p-dimensional data matrix of n instances

$$\begin{pmatrix} V_{11} & \ldots & V_{1f} & \ldots & V_{1p} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ V_{i1} & \ldots & V_{if} & \ldots & V_{ip} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ V_{n1} & \ldots & V_{nf} & \ldots & V_{np} \end{pmatrix}$$

Data Normalization by Z-score

- Average absolute deviation S_f of variable V_f

$$S_f = \frac{1}{n}\left(|v_{1f}-m_f|+|v_{2f}-m_f|+\ldots+|v_{nf}-m_f|\right)$$

where m_f is the mean of V_f: $m_f = \frac{1}{n}\left(v_{1f}+v_{2f}+\ldots+v_{nf}\right)$

- Computing the z-score of the v_{if} value:

$$z_{if} = \frac{v_{if}-m_f}{S_f}$$

- Z-score matrix

$$\begin{pmatrix} z_{11} & \ldots & z_{1f} & \ldots & z_{1p} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ z_{i1} & \ldots & z_{if} & \ldots & z_{ip} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ z_{n1} & \ldots & z_{nf} & \ldots & z_{np} \end{pmatrix}$$

Z-score Normalization: Example

Data matrix

Customer	Age	Income
C1	50	11000
C2	70	11100
C3	60	11122
C4	60	11074

m_{Age}=60
s_{Age} =5

m_{Income}=11074
s_{Income}=48

Z-scores

Customer	Age	Income
C1	-2	-0.5
C2	2	0.18
C3	0	0.32
C4	0	0

- Manhattan distance
  - d(C1,C2) = 120
  - d(C1,C3) = 132
- Conclusion: C1 is more similar to C2 than C3 ☹

- Manhattan distance
  - d(C1,C2) = 4.675
  - d(C1,C3) = 2.324
- Conclusion: C1 is more similar to C3 than C2 ☺

Binary Variables

- Variable with two distinct possible values
- Symmetric variable: Both values have equal weights for the distance
  - E.g. the gender of a person; coding male gender by 1 and female by 0 is the same as the reverse
- Asymmetric variable: One value is more frequent than the other
  - E.g. HIV test, that can be positive or negative (0 or 1) but two patients with a value of 1 for the test are more similar than two patients with 0 for the test
  - Generally, we code by 1 the least frequent modality

Binary Variables: Example

- Example binary matrix M

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Yes	No	Pos	Neg	Neg	Neg
Mary	F	Yes	No	Pos	Neg	Pos	Neg
Jim	M	Yes	Yes	Neg	Neg	Neg	Pos

- Gender is asymmetric, other variables are symmetric
- Yes and Pos ≡ 1, No and Neg ≡ 0
- Here, only asymmetric variables will be used to compute distances

Name	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	1	0	1	0	0	0
Mary	1	0	1	0	1	0
Jim	1	1	0	0	0	1

Binary Variables: Contingency table

- Contingency table for a pair X and Y of dataset instances: Number of co-occurrences for each combination of binary values

		Y: 1	Y: 0	sum
X	1	a	b	a+b
X	0	c	d	c+d
	sum	a+c	b+d	p

a: Number of variables with value '1' in both instances among the p binary variables

- For symmetric binary variables, the simple matching coefficient can be used to assess the distance between instances X and Y

$$d(X,Y) = \frac{b+c}{a+b+c+d}$$

- Number of dissimilar values divided by the total number of values

Asymmetric Binary Variables

- Co-occurrences of '0' are non informative and must be disregarded
- The Jaccard similarity coefficient disregards the d counts in the contingency matrix

$$d(X,Y) = \frac{b+c}{a+b+c}$$

- Example binary matrix M

$$d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(Mary, Jim) = \frac{1+2}{1+1+2} = 0.75$$

- The two most similar instances are Jack and Mary

Categorical Variables

- Generalization of the notion of binary variable
  - E.g. the color of an object: Color ∈ {blue, green, red, …}
- Method 1: Simple matching

$$d(X,Y) = \frac{p-m}{p}$$

where m is the number of pairings and p the number of variables

- Method 2: Use one binary variable for each categorical value

ID	Color	...
1	Blue	...
2	Green	...
3	Red	...
…	…	...

Binarization →

ID	Blue	Green	Red	...
1	1	0	0	...
2	0	1	0	...
3	0	0	1	...
…	…	…	…	...

- Use the Jaccard coefficient as the new variables are asymmetric

Ordinal Variables

- A categorical variable which possible values are ordered
  - Ex: Level ∈ {Low, Medium, High} where Low $<_p$ Medium $<_p$ High
- Method:
  1. Replace each value by its rank $r_i \in [1..N]$ where N is the number of values
  2. Compute z-scores to normalize ranks r_i in the interval [0.0, 1.0] and thus ensure independence vs. the number of states N
  3. Apply a Minkowski distance to the resulting z-scores (continuous var.)

ID	Level	…
1	Medium	…
2	Low	…
3	High	…
4	Medium	…
…	…	…

Ranks →

ID	Level	…
1	2	…
2	1	…
3	3	…
4	2	…
…	…	…

Z-scores →

ID	Level	…
1	0.5	…
2	0.0	…
3	1.0	…
4	0.5	…
…	…	…

Heterogeneous Datasets

- Let X = {X_1, …, X_f, …, X_p} and Y = {Y_1, …, Y_f, …, Y_p}
- Combine the different measurements using a weighted formula

$$d(X,Y) = \frac{\sum_{f=1}^{f=p} \delta_f(X,Y)\, d_f(X,Y)}{\sum_{f=1}^{f=p} \delta_f(X,Y)}$$

- If variable V_f is
  - Binary or categorical: If $X_f = Y_f$ then $d_f(X,Y) = 0$, otherwise $d_f(X,Y) = 1$
  - Continuous: Use a normalized distance measure
  - Ordinal: Compute z-scores z_{if} from ranks r_{if} and process result as a continuous variable
- $\delta_f(X,Y) = 0$ if $X_f = Y_f = 0$ and V_f is binary asymmetric (or a value is missing), otherwise $\delta_f(X,Y) = 1$

Clustering Algorithms

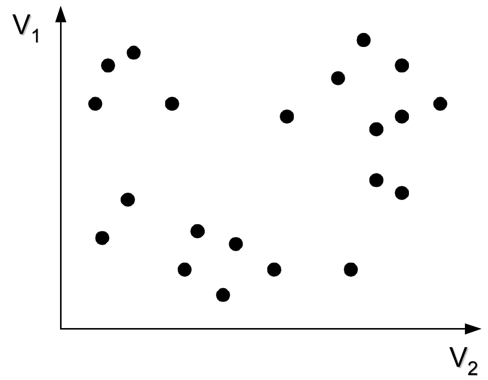
- Clustering a dataset is a NP-Hard problem (non-deterministic polynomial-time class)
- Clustering algorithms aim to provide a polynomial-time approximation of the optimal solution
- Different algorithmic approaches, that make use of different data space properties and grouping principles, have been proposed
  - Partitioning approaches
  - Hierarchical approaches
  - Density based approaches
  - Grid based approaches
  - Model (concepts) based approaches
  - Ensemble clustering (consensus based approaches)

Partitioning Approaches

- Principle: Partition the dataset instances into K groups where K is a user defined parameter
- K-means algorithm (H. Steinhaus, 1957)
  1. Initialization: Choose randomly K instances as initial centers, called <u>centroids</u>, of clusters to generate an initial partition of the dataset
  2. Iteration loop:
     a) For each instance, calculate its distance to the centroid of each cluster
     b) If required, re-assign each instance to the cluster which centroid is the nearest
     c) Re-calculate the centroid of each cluster as its barycenter
  3. Repeat the iteration loop if some instances where re-assigned

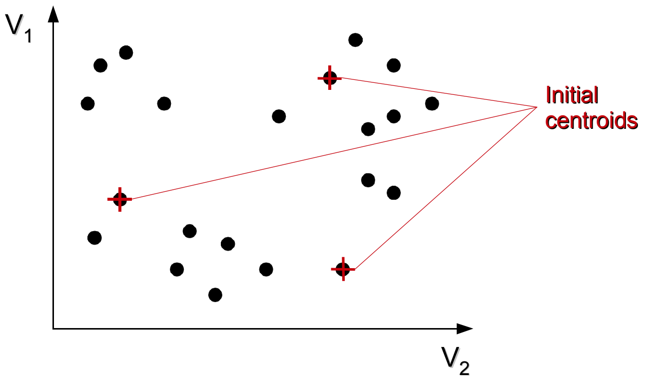
K-means Algorithm: Example

- Example bi-dimensional data space to partition into three clusters: Parameter K = 3



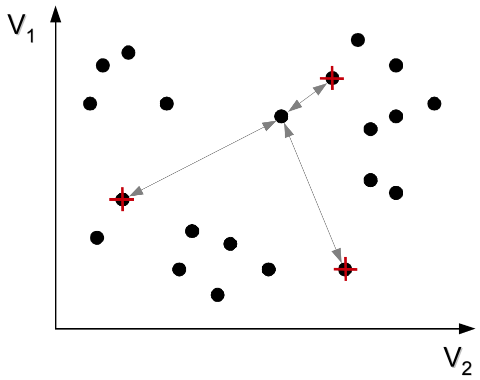
K-means Algorithm: Example

- Random initialization: 3 instances are chosen randomly as the initial centroids



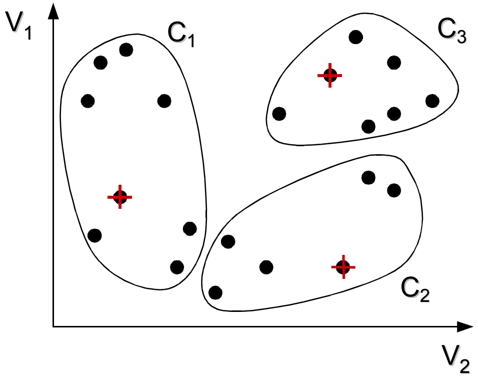
K-means Algorithm: Example

- Distance measure calculation: For each non-centroid instance, its distance to each centroid is calculated



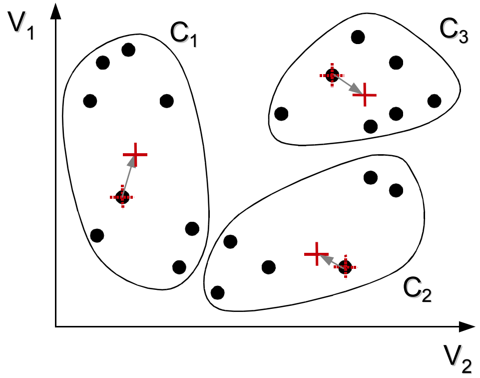
K-means Algorithm: Example

- Assignment of instances: Each non-centroid instance is assigned to the cluster which centroid is the nearest



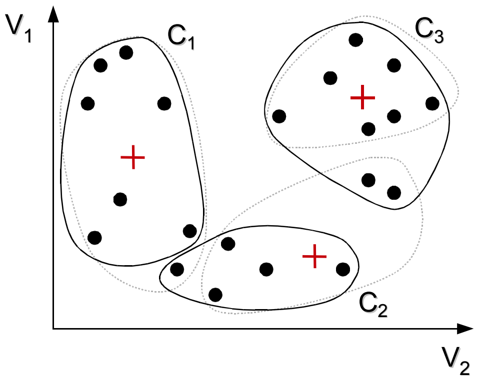
K-means Algorithm: Example

- Updating the centroids: The centroid of each cluster is re-calculated as the barycenter of the instances of the cluster (center of gravity)



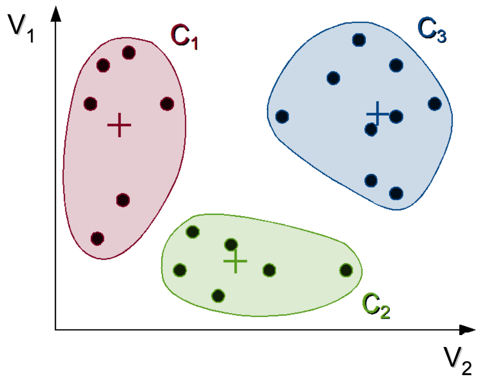
K-means Algorithm: Example

- Updating clusters: Distances between instances and centroids are re-calculated, and instances are re-assigned to another cluster if nearest



K-means Algorithm: Example

- The iterations stop when stability is reached, i.e. no re-assignment of instance is required



K-means Algorithm Properties

- Strengths
  - Efficient in term of computational time: Time complexity is O(K.N.T) for K clusters, N instances and T iterations (usually $K \ll T \ll N$)
  - Good scale-up properties with respect to the dataset size
- Weaknesses
  - Sensitive to noisy data and outliers
  - Can generate only convex clusters
  - Requires the end-user to define a priori the number K of clusters
  - Can process only variables for which the mean is calculable (unable to process discrete variables)
  - Non-deterministic: The result depends on the initial choice of centroids
- Some implementations try to improve the relevance of the chosen initial centers (assumptions on data value distributions)

Partitioning Algorithm Variants

- K-medoids algorithm (L. Kaufman & P.J. Rousseeuw, 1987)
  - Cluster centers are represented by <u>medoids</u> instead of centroids
  - The <u>medoid</u> is the most central instance of a cluster
  - Enables the processing of non-continuous data
  - Also known as the PAM (Partitioning Around Medoids) algorithm
- Strengths
  - Can process discrete variables (categorical, binary, etc.)
  - Each cluster is represented by a real instance
  - More robust to noisy data and outliers
- Weaknesses
  - Less efficient than K-means as it requires more computations

Hierarchical Clustering

- These algorithms give a hierarchical decomposition of the possible different clusters with a tree-like graphical representation named dendrogram
- Agglomeration approaches (e.g. AGNES, ROCK and UPGMA)
  - Start: Each instance is considered as a cluster
  - Iterations: Successively group the nearest clusters
  - Stop: A stopping condition is reached (measure < threshold or all instances in one cluster)
- Divisive approaches (e.g. DIANA, BIRCH and CURE)
  - Start: One cluster regroups all instances
  - Iterations: Successively divide the least compact clusters
  - Stop: A stopping condition is reached (measure < threshold or each instance constitutes a cluster)

Hierarchical Clustering: Example

- Example dataset D = {a, b, c, d, e} of five instances from which the following distance matrix is computed

Distance matrix

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00

- The clusters are constructed using an agglomeration approach

Hierarchical Clustering: Example

- Initialization: Each instance constitutes a cluster

Distance matrix

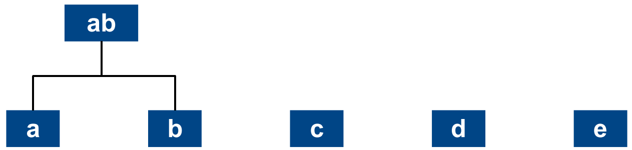
	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00

Hierarchical Clustering: Example

- Iteration: Merge the two nearest clusters
- They are identified as the minimal distance value in the distance matrix

Distance matrix

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00



Hierarchical Clustering: Example

- Iteration: Merge the two nearest clusters
- The distances d(ab,c), d(ab,d), d(ab,e), d(c,d), d(c,e) and d(d,e) are compared in the distance matrix

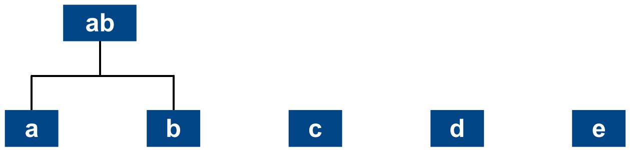
Distance matrix

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	0.21	0.00

d(ab,c) = avg(0.39,0.32) = 0.355

d(ab,d) = avg(0.43,0.34) = 0.385

d(ab,e) = avg(0.39,0.41) = 0.40

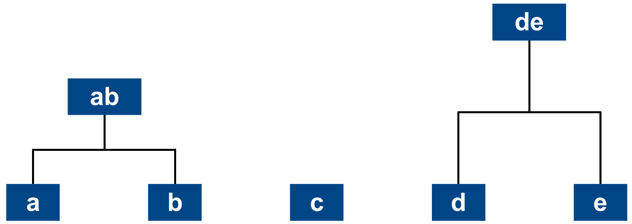


Hierarchical Clustering: Example

- Iteration: Merge the two nearest clusters
- Clusters {d} and {e} are merged since d(d,e) is the lowest distance

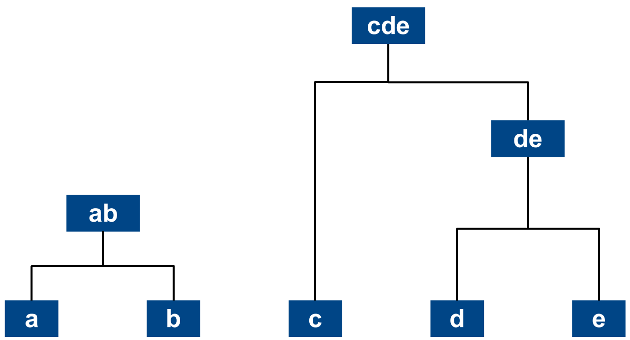
Distance matrix

	a	b	c	d	e
a	0.00				
b	0.18	0.00			
c	0.39	0.32	0.00		
d	0.43	0.34	0.25	0.00	
e	0.39	0.41	0.27	**0.21**	0.00



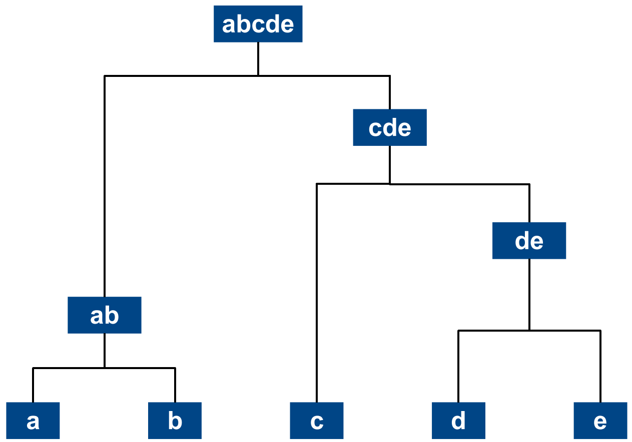
Hierarchical Clustering: Example

- Iteration: Merge the two nearest clusters
- The distances d(ab,c), d(ab,de) and d(c,de) are compared in the distance matrix: Clusters {c} and {de} are merged



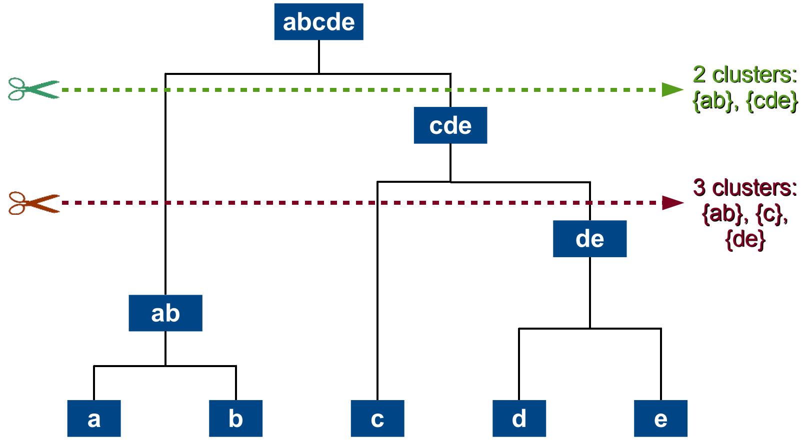
Hierarchical Clustering: Example

- Stop: Clusters {ab} and {cde} are merged into a unique cluster {abcde}



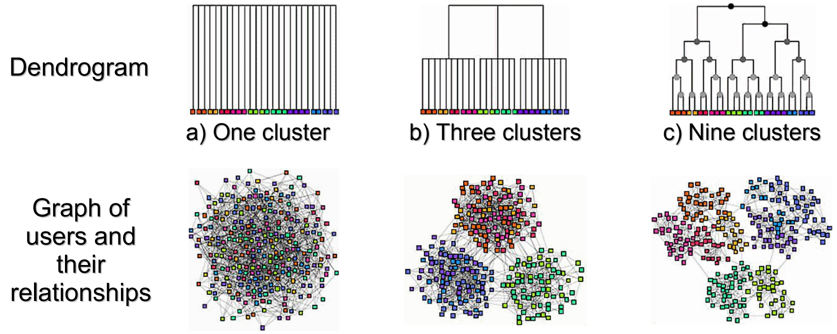
Hierarchical Clustering: Example

- The number of clusters obtained depends on the height at which one cuts the dendrogram



Dendrogram

- The dendrogram can provide groupings (clusters) at different levels of granularity
- Ex: Social network groups of users



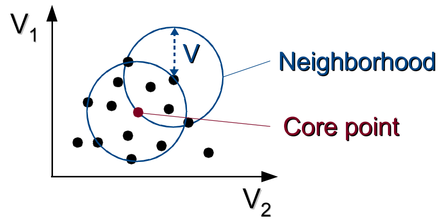
- The levels represent different types of relationships (acquaintance)

Hierarchical Clustering Properties

- Strengths
  - Do not require to define a priori the number K of clusters
  - Provides a hierarchical decomposition of clusters in a graphical representation that also depicts distances between them
- Weaknesses
  - Time complexity is $O(N^2.\log(N))$ for N instances
  - Poor scale-up properties with respect to the dataset size
  - Grouping instances in clusters is definitive: Erroneous decisions are impossible to correct later
  - Clusters tend to be of the same size

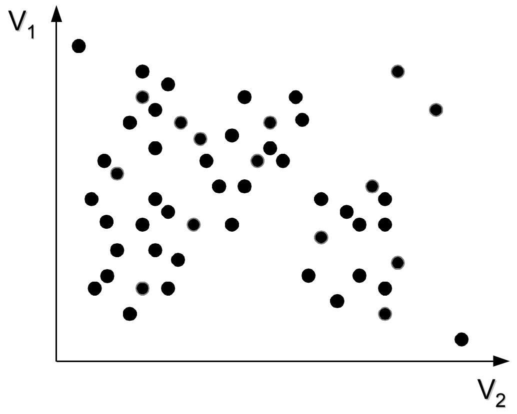
Density Based Approaches

- Principle: Identify dense regions of the data space that are separated by sparsely populated (non-dense) regions
- Definition: A <u>core point</u> in the multi-dimensional data space is a point which neighborhood size is at least equal to a threshold
- Parameters
  - V: Neighborhood distance in the data space
  - N: Minimal number of instances in the neighborhood of a core instance
- Example bi-dimensional data space
- Neighborhood size threshold N = 12



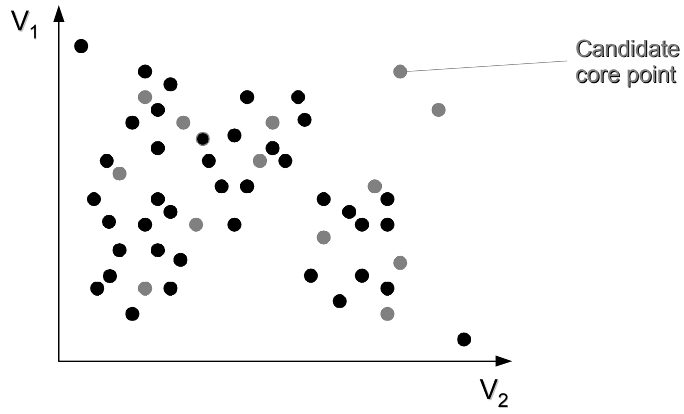
Density Based Approaches: Example

- Example bi-dimensional data space
- Parameter N = 3



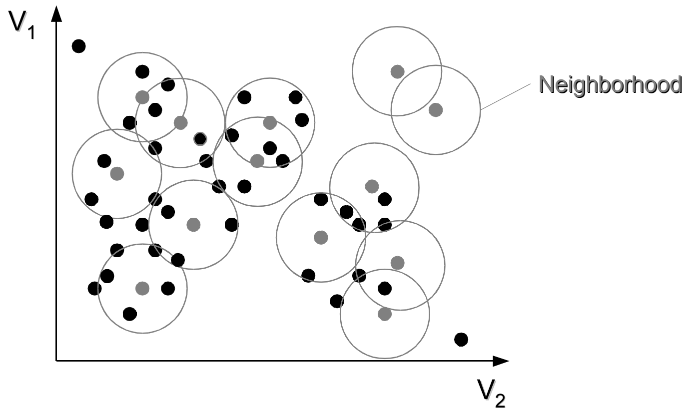
Density Based Approaches: Example

- Initialization: Random selection of candidate core points



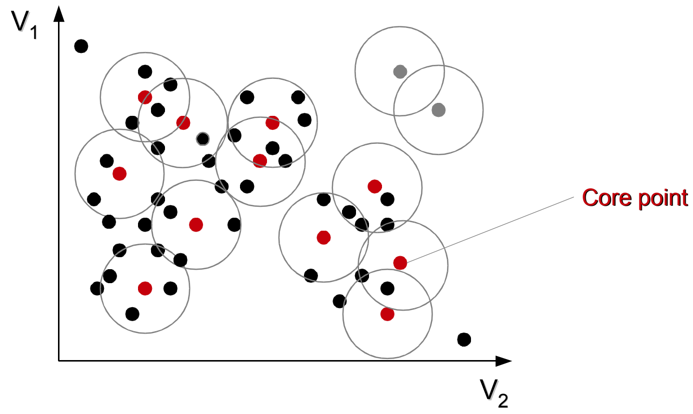
Density Based Approaches: Example

- Computing neighborhood of candidate core points



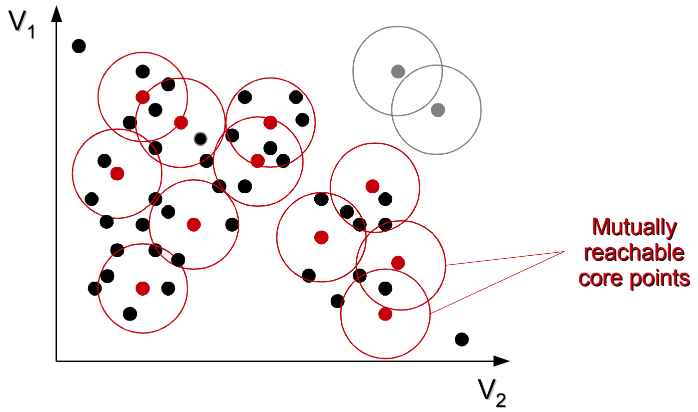
Density Based Approaches: Example

- Identifying core instances among candidates



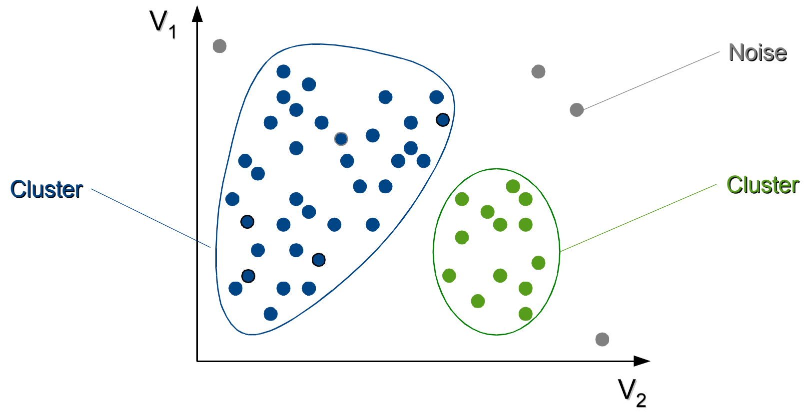
Density Based Approaches: Example

- Merging neighborhoods of core points that are mutually reachable (overlapping neighborhoods)



Density Based Approaches: Example

- Clusters can have different sizes and shapes, and noisy data is identified as isolated points

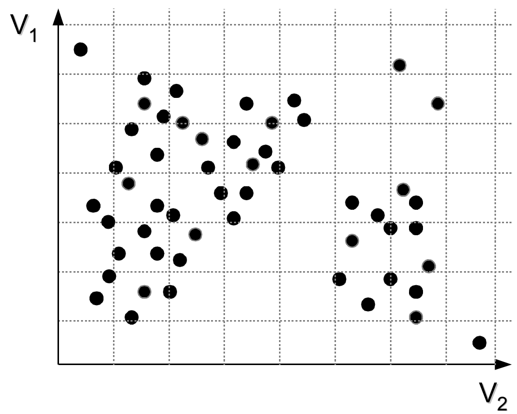


Density Based Approaches Properties

- Strengths
  - Do not require to define a priori the number K of clusters
  - Clusters can have different sizes and shapes
  - Robust to noisy data and outliers
- Weaknesses
  - Time complexity is $O(N^2)$ for N instances, or O(N.log(N)) if a spatial index data structure is used: Processing very large datasets can be costly
  - Less adequate to discrete variables than to continuous variables
- Representative algorithms
  - DBSCAN (Ester *et al.*, 1996)
  - DENCLUE (Hinneburg & Keim, 1998)
  - OPTICS (Ankerst *et al.*, 1999)

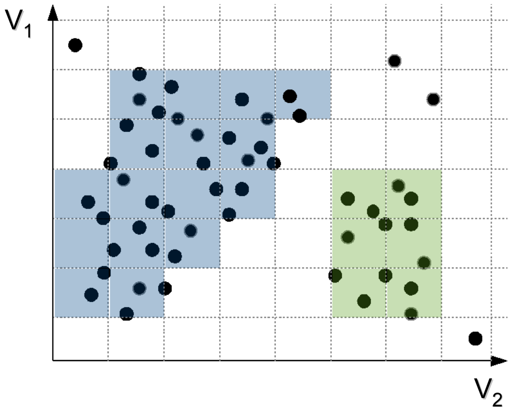
Grid Based Approaches: Example

- The multi-dimensional data space is divided into cells: Each variable is discretized in an equal-width manner
- The width of the discretization intervals determines the size and number of cells
- It can be parametrized to adapt to the variable distribution



Grid Based Approaches: Example

- Dense cells are cells that contain a number of instances at least equal to a minimum user-defined threshold (neighborhood size)
- Adjacent dense cells are merged to form clusters
- Noisy data and outlier are disregarded as they are contained in sparsely populated cells



Grid Based Approaches: Example

- Strengths
  - Similar to density based approaches: No parameter K, clusters with arbitrary sizes and shapes, and robustness to noisy data and outliers
  - The parametrization of the grid resolution is used to adapt the process to the data: Trade-off between efficiency and accuracy of the result
  - Good scale-up properties with respect to the number of instances and dimensions
- Weaknesses
  - Continuous variables must be discretized, which can be difficult to automatically optimize
- Representative algorithms:
  - STING (Wang *et al.*, 1997), CLIQUE (Agrawal *et al.*,1998), WaveCluster (Sheikholeslami *et al.*, 1998)

Model Based Approaches

- Generates a hierarchical decomposition of clusters
- A graphical representation depicts the hierarchy in a tree-like or lattice structure
- Each cluster provides a summarized description of the variable values that distinguish the cluster from other clusters
- The description can take different forms, e.g. probabilities, distributions, populations
- Terminology: Conceptual clustering, bi-clustering or co-clustering

Model Based Approaches: Example

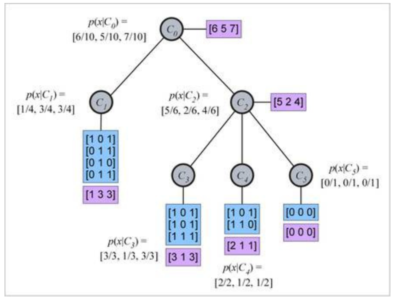
- Clusters are represented as nodes in a tree-like diagram where edges represent inclusion relationships

Tri-dimensional dataset

ID	V_1	V_2	V_3
1	1	0	1
2	0	1	1
3	1	0	1
4	1	0	1
5	0	0	0
6	1	1	0
7	1	0	1
8	0	1	0
9	1	1	1
10	0	1	1

Clustering

Probabilistic clustering model



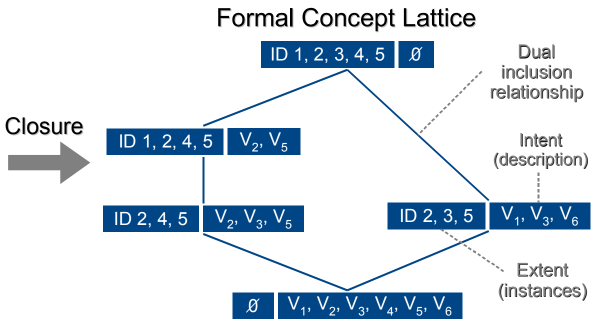
- Descriptive statistics (probabilities) of values for each variable characterize each cluster from C_0 (root) to C_5 (leaf node)

Bi-clustering Approaches: Example

- Clusters, called bi-clusters, are pairs: List of instances and list of their common features (variable values)
- E.g. closed sets in the data matrix (maximal rectangles) are bi-clusters called formal concepts
- They are represented as nodes in the formal concept lattice

Data matrix

ID	V_1	V_2	V_3	V_4	V_5	V_6
1	0	1	0	0	1	0
2	1	1	1	0	1	1
3	1	0	1	1	0	1
4	0	1	1	0	1	0
5	1	1	1	0	1	1

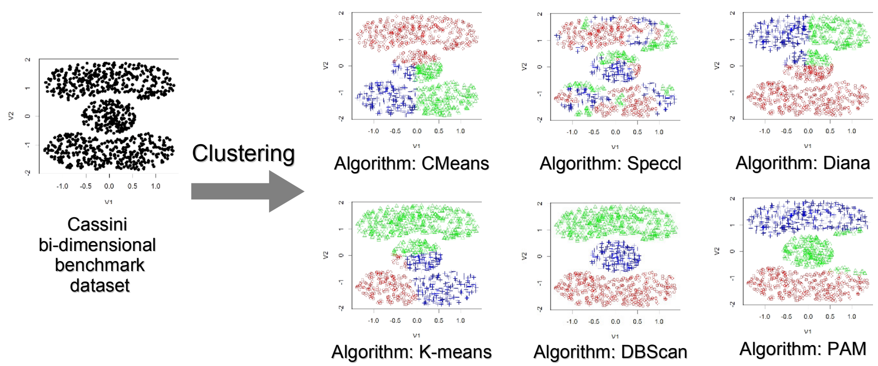


Model Based Approaches Properties

- Strengths
  - A characterization (distinctive description) of each cluster is provided
  - Subjacent theoretical problems are well-known and efficient algorithmic solutions exist
  - E.g. the Category Utility measure to maximize (resp. minimize) the probability that two instances in the same (resp. different) cluster(s) have common values (quadratic function optimization)
  - E.g. efficient level-wise algorithms for extracting closed sets
- Weaknesses
  - Continuous variables must be discretized (difficult to optimize)
- Representative algorithms:
  - Cobweb (Fischer, 1987), SUBDUE (Jonyer *et al*., 2001), CC (Cheng *et al*., 2000), GCF (Talavera *et al*., 2001), FIST (Mondal *et al*., 2011)

Ensemble Clustering

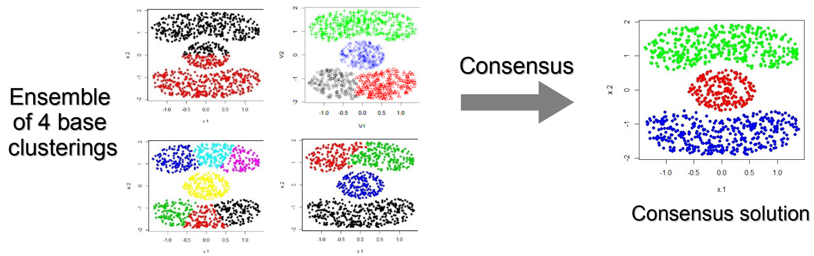
- Different algorithmic configurations (algorithm and parametrization) can generate different clustering results



- Problem: Evaluation measures assess clusters relevance vs. specific properties that can be inappropriate for the data space

Ensemble Clustering

- Combine multiple base clusterings into a new solution that provides better partitioning of the dataset instances



- Different algorithmic approaches to combine base clusterings: Graph based, majority voting based, co-association matrix based, distance based, fragments based, closed sets based
- The consensus solution is constructed to be as similar as possible to the ensemble (set of base clusterings)

Ensemble Clustering

- Strengths
  - Can discover clusters of arbitrary sizes and shapes
  - Can discover clusters corresponding to different types of structures in the data space (dense areas, convex groups, etc.)
  - Robust to noisy data and outliers
- Weaknesses
  - Requires to apply different clustering algorithmic configurations to construct the ensemble, which may be costly for very large datasets
- Representative algorithms:
  - Evidence Accumulation (Fred *et al.*, 2002), CSPA, HGPA, MCLA (Strehl *et al.*, 2003), HBGF (Fern *et al.*, 2004), WClustering (Li *et al.*, 2008), WSBPA (Domeniconi *et al.*, 2009), Probability Accumulation (Wang *et al.*, 2009), MultiCons (Al-Najdi *et al.*, 2016)

Bibliography and References

- Bibliography

  - C. C. Aggarwal & C. K. Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, August 2013.
  - G. Gan, C. Ma & J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. SIAM Publisher, July 2007.
  - M. J. Zaki & W. Meira. *Data Mining and Analysis – Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.

- Web sites
  - KDNuggets: Business Analytics, Big Data, Data Mining, Data Science, and Machine Learning. http://www.kdnuggets.com/
  - R and Data Mining: Documents, examples, tutorials and resources on R and data mining. http://www.rdatamining.com/
  - CRAN Task View: Cluster Analysis & Finite Mixture Models. https://cran.r-project.org/web/views/Cluster.html