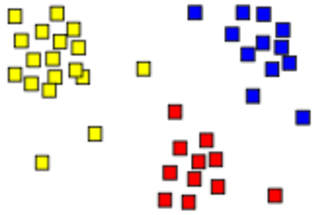
WIKIPEDIA

Cluster analysis



The result of a cluster analysis shown as the coloring of the squares into three clusters.

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances between cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including parameters such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

Besides the term *clustering*, there are a number of terms with similar meanings, including *automatic classification*, *numerical taxonomy*, *botryology* (from Greek βότρυς "grape") and *typological analysis*. The subtle differences are often in the use of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest.

Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert Tryon in 1939[1][2] and famously used by Cattell beginning in 1943[3] for trait theory classification in personality psychology.

Contents

Definition

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms.[4] There is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these "cluster models" is key to understanding the differences between the various algorithms. Typical cluster models include:

- *Connectivity models*: for example, hierarchical clustering builds models based on distance connectivity.
- *Centroid models*: for example, the k-means algorithm represents each cluster by a single mean vector.
- *Distribution models*: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the expectation-maximization algorithm.
- *Density models*: for example, DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- *Subspace models*: in biclustering (also known as co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
- *Group models*: some algorithms do not provide a refined model for their results and just provide the grouping information.
- *Graph-based models*: a clique, that is, a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques, as in the HCS clustering algorithm.
- *Neural models*: the most well known unsupervised neural network is the self-organizing map and these models can usually be characterized as similar to one or more of the above models, and including subspace models when neural networks implement a form of Principal Component Analysis or Independent Component Analysis.

A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example, a hierarchy of clusters embedded in each other. Clusterings can be roughly distinguished as:

- *Hard clustering*: each object belongs to a cluster or not
- *Soft clustering* (also: *fuzzy clustering*): each object belongs to each cluster to a certain degree (for example, a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- *Strict partitioning clustering*: each object belongs to exactly one cluster
- *Strict partitioning clustering with outliers*: objects can also belong to no cluster, and are considered outliers
- *Overlapping clustering* (also: *alternative clustering*, *multi-view clustering*): objects may belong to more than one cluster; usually involving hard clusters
- *Hierarchical clustering*: objects that belong to a child cluster also belong to the parent cluster
- *Subspace clustering*: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap

Algorithms

Clustering algorithms can be categorized based on their cluster model, as listed above. The following overview will only list the most prominent examples of clustering algorithms, as there are possibly over 100 published clustering algorithms. Not all provide models for their clusters and can thus not easily be categorized. An overview of algorithms explained in Wikipedia can be found in the list of statistics algorithms.

There is no objectively "correct" clustering algorithm, but as it was noted, "clustering is in the eye of the beholder."[4] The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another. It should be noted that an algorithm that is designed for one kind of model will generally fail on a data set that contains a radically different kind of model.[4] For example, k-means cannot find non-convex clusters.[4]

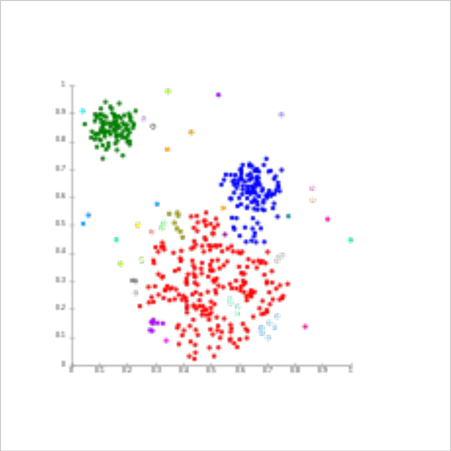
Connectivity-based clustering (hierarchical clustering)

Connectivity-based clustering, also known as *hierarchical clustering*, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

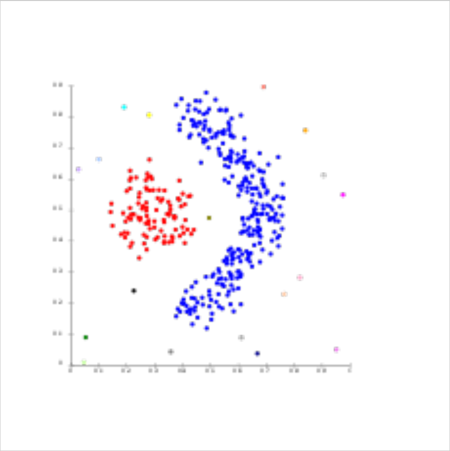
Connectivity-based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances) or UPGMA ("Unweighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

These methods will not produce a unique partitioning of the data set, but a hierarchy from which the user still needs to choose appropriate clusters. They are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as "chaining phenomenon", in particular with single-linkage clustering). In the general case, the complexity is $\mathcal{O}(n^3)$ for agglomerative clustering and $\mathcal{O}(2^{n-1})$ for divisive clustering,[5] which makes them too slow for large data sets. For some special cases, optimal efficient methods (of complexity $\mathcal{O}(n^2)$) are known: SLINK[6] for single-linkage and CLINK[7] for complete-linkage clustering. In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering.

Linkage clustering examples



Single-linkage on Gaussian data. At 35 clusters, the biggest cluster starts fragmenting into smaller parts, while before it was still connected to the second largest due to the single-link effect.



Single-linkage on density-based clusters. 20 clusters extracted, most of which contain single elements, since linkage clustering does not have a notion of "noise".

Centroid-based clustering

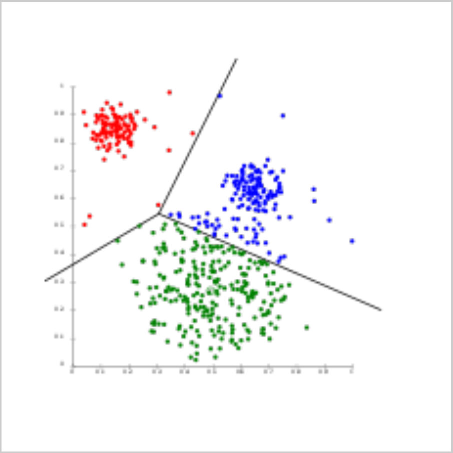
In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to *k*, *k*-means clustering gives a formal definition as an optimization problem: find the *k* cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximate method is Lloyd's algorithm,[8] often just referred to as "*k-means algorithm*" (although another algorithm introduced this name). It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of *k*-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (*k*-medoids), choosing medians (*k*-medians clustering), choosing the initial centers less randomly (*k*-means++) or allowing a fuzzy cluster assignment (fuzzy c-means).

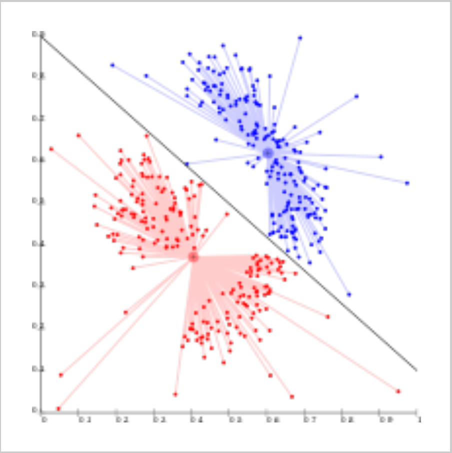
Most *k*-means-type algorithms require the number of clusters - *k* - to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders of clusters (which is not surprising since the algorithm optimizes cluster centers, not cluster borders).

K-means has a number of interesting theoretical properties. First, it partitions the data space into a structure known as a Voronoi diagram. Second, it is conceptually close to nearest neighbor classification, and as such is popular in machine learning. Third, it can be seen as a variation of model based clustering, and Lloyd's algorithm as a variation of the Expectation-maximization algorithm for this model discussed below.

k-means clustering examples



K-means separates data into Voronoi-cells, which assumes equal-sized clusters (not adequate here)



K-means cannot represent density-based clusters

Distribution-based clustering

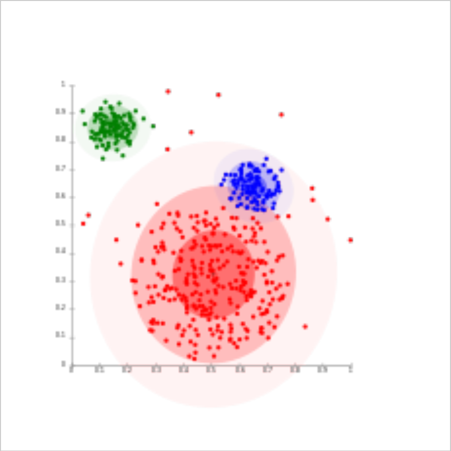
The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A convenient property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.

While the theoretical foundation of these methods is excellent, they suffer from one key problem known as overfitting, unless constraints are put on the model complexity. A more complex model will usually be able to explain the data better, which makes choosing the appropriate model complexity inherently difficult.

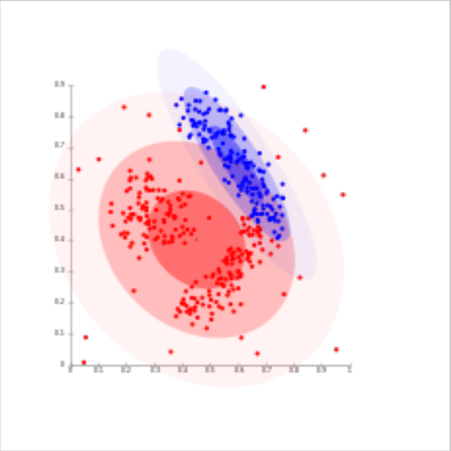
One prominent method is known as Gaussian mixture models (using the expectation-maximization algorithm). Here, the data set is usually modeled with a fixed (to avoid overfitting) number of Gaussian distributions that are initialized randomly and whose parameters are iteratively optimized to better fit the data set. This will converge to a local optimum, so multiple runs may produce different results. In order to obtain a hard clustering, objects are often then assigned to the Gaussian distribution they most likely belong to; for soft clusterings, this is not necessary.

Distribution-based clustering produces complex models for clusters that can capture correlation and dependence between attributes. However, these algorithms put an extra burden on the user: for many real data sets, there may be no concisely defined mathematical model (e.g. assuming Gaussian distributions is a rather strong assumption on the data).

Expectation-maximization (EM) clustering examples



On Gaussian-distributed data, EM works well, since it uses Gaussians for modelling clusters



Density-based clusters cannot be modeled using Gaussian distributions

Density-based clustering

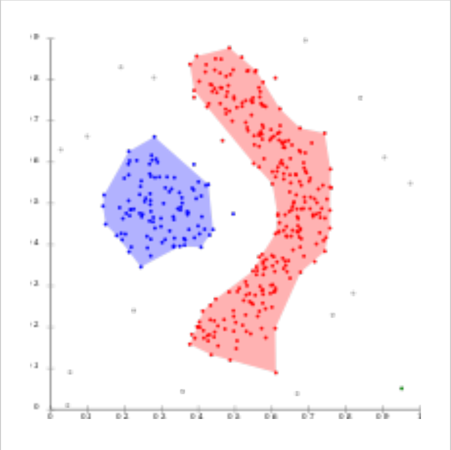
In density-based clustering,[9] clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

The most popular[10] density based clustering method is DBSCAN.[11] In contrast to many newer methods, it features a well-defined cluster model called "density-reachability". Similar to linkage based clustering, it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times. OPTICS[12] is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter ε, and produces a hierarchical result related to that of linkage clustering. DeLi-Clu,[13] Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the ε parameter entirely and offering performance improvements over OPTICS by using an R-tree index.

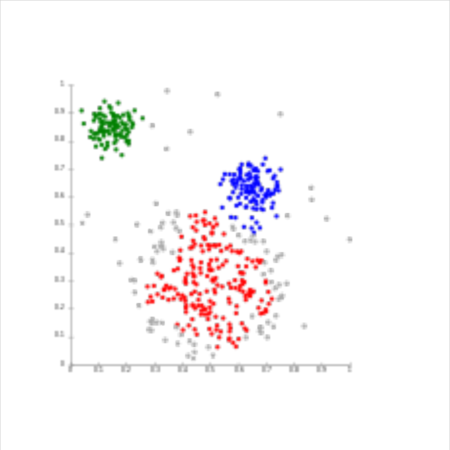
The key drawback of DBSCAN and OPTICS is that they expect some kind of density drop to detect cluster borders. On data sets with, for example, overlapping Gaussian distributions - a common use case in artificial data - the cluster borders produced by these algorithms will often look arbitrary, because the cluster density decreases continuously. On a data set consisting of mixtures of Gaussians, these algorithms are nearly always outperformed by methods such as EM clustering that are able to precisely model this kind of data.

Mean-shift is a clustering approach where each object is moved to the densest area in its vicinity, based on kernel density estimation. Eventually, objects converge to local maxima of density. Similar to k-means clustering, these "density attractors" can serve as representatives for the data set, but mean-shift can detect arbitrary-shaped clusters similar to DBSCAN. Due to the expensive iterative procedure and density estimation, mean-shift is usually slower than DBSCAN or k-Means. Besides that, the applicability of the mean-shift algorithm to multidimensional data is hindered by the unsmooth behaviour of the kernel density estimate, which results in over-fragmentation of cluster tails.[13]

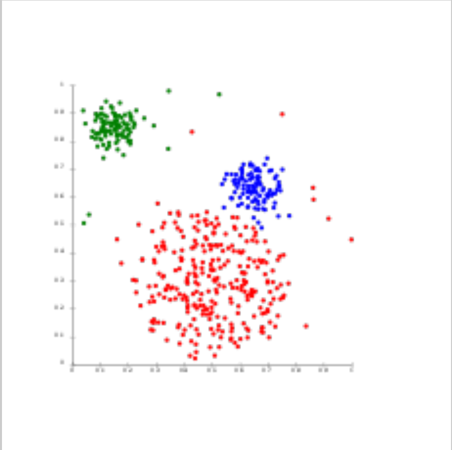
Density-based clustering examples



Density-based clustering with DBSCAN.



DBSCAN assumes clusters of similar density, and may have problems separating nearby clusters



OPTICS is a DBSCAN variant that handles different densities much better

Recent developments

In recent years, considerable effort has been put into improving the performance of existing algorithms.[14][15] Among them are *CLARANS* (Ng and Han, 1994),[16] and *BIRCH* (Zhang et al., 1996).[17] With the recent need to process larger and larger data sets (also known as big data), the willingness to trade semantic meaning of the generated clusters for performance has been increasing. This led to the development of pre-clustering methods such as canopy clustering, which can process huge data sets efficiently, but the resulting "clusters" are merely a rough pre-partitioning of the data set to then analyze the partitions with existing slower methods such as k-means clustering. Various other approaches to clustering have been tried such as seed based clustering.[18]

For high-dimensional data, many of the existing methods fail due to the curse of dimensionality, which renders particular distance functions problematic in high-dimensional spaces. This led to new clustering algorithms for high-dimensional data that focus on subspace clustering (where only some attributes are used, and cluster models include the relevant attributes for the cluster) and correlation clustering that also looks for arbitrary rotated ("correlated") subspace clusters that can be modeled by giving a correlation of their attributes.[19] Examples for such clustering algorithms are CLIQUE[20] and SUBCLU.[21]

Ideas from density-based clustering methods (in particular the DBSCAN/OPTICS family of algorithms) have been adopted to subspace clustering (HiSC,[22] hierarchical subspace clustering and DiSH[23]) and correlation clustering (HiCO,[24] hierarchical correlation clustering, 4C[25] using "correlation connectivity" and ERiC[26] exploring hierarchical density-based correlation clusters).

Several different clustering systems based on mutual information have been proposed. One is Marina Meilă's *variation of information* metric;[27] another provides hierarchical clustering.[28] Using genetic algorithms, a wide range of different fit-functions can be optimized, including mutual information.[29] Also message passing algorithms, a recent development in computer science and statistical physics, has led to the creation of new types of clustering algorithms.[30]

Evaluation and assessment

Evaluation (or "validation") of clustering results is as difficult as the clustering itself.[31] Popular approaches involve "*internal*" evaluation, where the clustering is summarized to a single quality score, "*external*" evaluation, where the clustering is compared to an existing "ground truth" classification, "*manual*" evaluation by a human expert, and "*indirect*" evaluation by evaluating the utility of the clustering in its intended application.[32]

Internal evaluation measures suffer from the problem that they represent functions that themselves can be seen as a clustering objective. For example, one could cluster the data set by the Silhouette coefficient; except that there is no known efficient algorithm for this. By using such an internal measure for evaluation, we rather compare the similarity of the optimization problems,[32] and not necessarily how useful the clustering is.

External evaluation has similar problems: if we have such "ground truth" labels, then we would not need to cluster; and in practical applications we usually do not have such labels. On the other hand, the labels only reflect one possible partitioning of the data set, which does not imply that there does not exist a different, and maybe even better, clustering.

Neither of these approaches can therefore ultimately judge the actual quality of a clustering, but this needs human evaluation,[32] which is highly subjective. Nevertheless, such statistics can be quite informative in identifying bad clusterings,[33] but one should not dismiss subjective human evaluation.[33]

Internal evaluation

When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. Those that use a gold standard are called external measures and are discussed in the next section - although when they are symmetric they may also be used as measures between two clusters for internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications.[34] Additionally, this evaluation is biased towards algorithms that use the same cluster model. For example, k-means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering.

Therefore, the internal evaluation measures are best suited to get some insight into situations where one algorithm performs better than another, but this shall not imply that one algorithm produces more valid results than another.[4] Validity as measured by such an index depends on the claim that this kind of structure exists in the data set. An algorithm designed for some kind of models has no chance if the data set contains a radically different set of models, or if the evaluation measures a radically different criterion.[4] For example, k-means clustering can only find convex clusters, and many evaluation indexes assume convex clusters. On a data set with non-convex clusters neither the use of k-means, nor of an evaluation criterion that assumes convexity, is sound.

The following methods can be used to assess the quality of clustering algorithms based on internal criterion:

- **Davies–Bouldin index**

  The Davies–Bouldin index can be calculated by the following formula:

  $$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

  where n is the number of clusters, c_x is the centroid of cluster x, σ_x is the average distance of all elements in cluster x to centroid c_x, and $d(c_i, c_j)$ is the distance between centroids c_i and c_j. Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies–Bouldin index, the clustering algorithm that produces a collection of clusters with the smallest Davies–Bouldin index is considered the best algorithm based on this criterion.

- **Dunn index**

  The Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula:[35]

  $$D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)} \,,$$

  where *d*(*i*,*j*) represents the distance between clusters *i* and *j*, and *d* '(*k*) measures the intra-cluster distance of cluster *k*. The inter-cluster distance *d*(*i*,*j*) between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, the intra-cluster distance *d* '(*k*) may be measured in a variety ways, such as the maximal distance between any pair of elements in cluster *k*. Since internal criterion seek clusters with high intra-cluster similarity and low inter-cluster similarity, algorithms that produce clusters with high Dunn index are more desirable.

- Silhouette coefficient

  The silhouette coefficient contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers. This index works well with k-means clustering, and is also used to determine the optimal number of clusters.

External evaluation

In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by (expert) humans. Thus, the benchmark sets can be thought of as a gold standard for evaluation.[31] These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. However, it has recently been discussed whether this is adequate for real data, or only on synthetic data sets with a factual ground truth, since classes can contain internal structure, the attributes present may not allow separation of clusters or the classes may contain anomalies.[36] Additionally, from a knowledge discovery point of view, the reproduction of known knowledge may not necessarily be the intended result.[36] In the special scenario of constrained clustering, where meta information (such as class labels) is used already in the clustering process, the hold-out of information for evaluation purposes is non-trivial.[37]

A number of measures are adapted from variants used to evaluate classification tasks. In place of counting the number of times a class was correctly assigned to a single data point (known as true positives), such *pair counting* metrics assess whether each pair of data points that is truly in the same cluster is predicted to be in the same cluster.[31]

Some of the measures of quality of a cluster algorithm using external criterion include:

- **Purity**: Purity is a measure of the extent to which clusters contain a single class.[34] Its calculation can be thought of as follows: For each cluster, count the number of data points from the most common class in said cluster. Now take the sum over all clusters and divide by the total number of data points. Formally, given some set of clusters M and some set of classes D, both partitioning N data points, purity can be defined as:

  $$\frac{1}{N}\sum_{m \in M}\max_{d \in D}|m \cap d|$$

  Note that this measure doesn't penalise having many clusters. So for example, a purity score of 1 is possible by putting each data point in its own cluster.

- **Rand measure** (William M. Rand 1971)[38]

  The Rand index computes how similar the clusters (returned by the clustering algorithm) are to the benchmark classifications. One can also view the Rand index as a measure of the percentage of correct decisions made by the algorithm. It can be computed using the following formula:
  $$RI = \frac{TP + TN}{TP + FP + FN + TN}$$
  where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. One issue with the Rand index is that false positives and false negatives are equally weighted. This may be an undesirable characteristic for some clustering applications. The F-measure addresses this concern, as does the chance-corrected adjusted Rand index.

- **F-measure**

  The F-measure can be used to balance the contribution of false negatives by weighting recall through a parameter $\beta \geq 0$. Let **precision** and **recall** (both external evaluation measures in themselves) be defined as follows:
  $$P = \frac{TP}{TP + FP}$$
  $$R = \frac{TP}{TP + FN}$$
  where P is the precision rate and R is the recall rate. We can calculate the F-measure by using the following formula:[34]
  $$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$
  Notice that when $\beta = 0$, $F_0 = P$. In other words, recall has no impact on the F-measure when $\beta = 0$, and increasing β allocates an increasing amount of weight to recall in the final F-measure.
  Also note that TN is not taken into account and can vary from 0 upward without bound.

- **Jaccard index**

  The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. An index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:
  $$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$
  This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.
  Also note that TN is not taken into account and can vary from 0 upward without bound.

- **Dice index**

  The Dice symmetric measure doubles the weight on TP while still ignoring TN and is equivalent to F1 - the F-measure with $\beta = 1$:

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{2TP}{2TP+FP+FN}$$

- **Fowlkes–Mallows index** (E. B. Fowlkes & C. L. Mallows 1983)[39]

  The Fowlkes-Mallows index computes the similarity between the clusters returned by the clustering algorithm and the benchmark classifications. The higher the value of the Fowlkes-Mallows index the more similar the clusters and the benchmark classifications are. It can be computed using the following formula:

  $$FM=\sqrt{\frac{TP}{TP+FP}\cdot\frac{TP}{TP+FN}}$$

  where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. The FM index is the geometric mean of the precision and recall P and R, and is thus also known as the G-measure, while the F-measure is their harmonic mean.[40][41] Moreover, precision and recall are also known as Wallace's indices B^I and B^{II}.[42] Chance normalized versions of recall, precision and G-measure correspond to Informedness, Markedness and Matthews Correlation and relate strongly to Kappa.[43]

- The **mutual information** is an information theoretic measure of how much information is shared between a clustering and a ground-truth classification that can detect a non-linear similarity between two clusterings. Normalized mutual information is a family of corrected-for-chance variants of this that has a reduced bias for varying cluster numbers.[31]
- **Confusion matrix**

  A confusion matrix can be used to quickly visualize the results of a classification (or clustering) algorithm. It shows how different a cluster is from the gold standard cluster.

Cluster tendency

To measure cluster tendency is to measure to what degree clusters exist in the data to be clustered, and may be performed as an initial test, before attempting clustering. One way to do this is to compare the data against random data. On average, random data should not have clusters.

- **Hopkins statistic**

  There are multiple formulations of the Hopkins Statistic.[44] A typical one is as follows.[45] Let X be the set of n data points in d dimensional space. Consider a random sample (without replacement) of $m \ll n$ data points with members x_i. Also generate a set Y of m uniformly randomly distributed data points. Now define two distance measures, u_i to be the distance of $y_i \in Y$ from its nearest neighbor in X and w_i to be the distance of $x_i \in X$ from its nearest neighbor in X. We then define the Hopkins statistic as:

  $$H=\frac{\sum_{i=1}^m u_i^d}{\sum_{i=1}^m u_i^d+\sum_{i=1}^m w_i^d}\,,$$

  With this definition, uniform random data should tend to have values near to 0.5, and clustered data should tend to have values nearer to 1.
  However, data containing just a single Gaussian will also score close to 1, as this statistic measures deviation from a *uniform* distribution, not multimodality, making this statistic largely useless in application (as real data never is remotely uniform).

Applications

Biology, computational biology and bioinformatics

Plant and animal ecology

cluster analysis is used to describe and to make spatial and temporal comparisons of communities (assemblages) of organisms in heterogeneous environments; it is also used in plant systematics to generate artificial phylogenies or clusters of organisms (individuals) at the species, genus or higher level that share a number of attributes

Transcriptomics

clustering is used to build groups of genes with related expression patterns (also known as coexpressed genes) as in HCS clustering algorithm . Often such groups contain functionally related proteins, such as enzymes for a specific pathway, or genes that are co-regulated. High throughput experiments using expressed sequence tags (ESTs) or DNA microarrays can be a powerful tool for genome annotation, a general aspect of genomics.

Sequence analysis

clustering is used to group homologous sequences into gene families. This is a very important concept in bioinformatics, and evolutionary biology in general. See evolution by gene duplication.

High-throughput genotyping platforms

clustering algorithms are used to automatically assign genotypes.

Human genetic clustering

The similarity of genetic data is used in clustering to infer population structures.

Medicine

Medical imaging

On PET scans, cluster analysis can be used to differentiate between different types of tissue in a three-dimensional image for many different purposes.[46]

Analysis of antimicrobial activity

Cluster analysis can be used to analyse patterns of antibiotic resistance, to classify antimicrobial compounds according to their mechanism of action, to classify antibiotics according to their antibacterial activity.

IMRT segmentation

Clustering can be used to divide a fluence map into distinct regions for conversion into deliverable fields in MLC-based Radiation Therapy.

Business and marketing

Market research

Cluster analysis is widely used in market research when working with multivariate data from surveys and test panels. Market researchers use cluster analysis to partition the general population of consumers into market segments and to better understand the relationships between different groups of consumers/potential customers, and for use in market segmentation, Product positioning, New product development and Selecting test markets.

Grouping of shopping items

Clustering can be used to group all the shopping items available on the web into a set of unique products. For example, all the items on eBay can be grouped into unique products. (eBay doesn't have the concept of a SKU)

World wide web

Social network analysis

In the study of social networks, clustering may be used to recognize communities within large groups of people.

Search result grouping

In the process of intelligent grouping of the files and websites, clustering may be used to create a more relevant set of search results compared to normal search engines like Google. There are currently a number of web based clustering tools such as Clusty.

Slippy map optimization

Flickr's map of photos and other map sites use clustering to reduce the number of markers on a map. This makes it both faster and reduces the amount of visual clutter.

Computer science

Software evolution

Clustering is useful in software evolution as it helps to reduce legacy properties in code by reforming functionality that has become dispersed. It is a form of restructuring and hence is a way of direct preventative maintenance.

Image segmentation

Clustering can be used to divide a digital image into distinct regions for border detection or object recognition.[47]

Evolutionary algorithms

Clustering may be used to identify different niches within the population of an evolutionary algorithm so that reproductive opportunity can be distributed more evenly amongst the evolving species or subspecies.

Recommender systems

Recommender systems are designed to recommend new items based on a user's tastes. They sometimes use clustering algorithms to predict a user's preferences based on the preferences of other users in the user's cluster.

Markov chain Monte Carlo methods

Clustering is often utilized to locate and characterize extrema in the target distribution.

Anomaly detection

Anomalies/outliers are typically - be it explicitly or implicitly - defined with respect to clustering structure in data.

Social science

Crime analysis

Cluster analysis can be used to identify areas where there are greater incidences of particular types of crime. By identifying these distinct areas or "hot spots" where a similar crime has happened over a period of time, it is possible to manage law enforcement resources more effectively.

Educational data mining

Cluster analysis is for example used to identify groups of schools or students with similar properties.

Typologies

From poll data, projects such as those undertaken by the Pew Research Center use cluster analysis to discern typologies of opinions, habits, and demographics that may be useful in politics and marketing.

Others

Field robotics

Clustering algorithms are used for robotic situational awareness to track objects and detect outliers in sensor data.[48]

Mathematical chemistry

To find structural similarity, etc., for example, 3000 chemical compounds were clustered in the space of 90 topological indices.[49]

Climatology

To find weather regimes or preferred sea level pressure atmospheric patterns.[50]

Petroleum geology

Cluster analysis is used to reconstruct missing bottom hole core data or missing log curves in order to evaluate reservoir properties.

Physical geography

The clustering of chemical properties in different sample locations.

See also

Specialized types of cluster analysis

- Balanced clustering
- Clustering high-dimensional data
- Conceptual clustering
- Consensus clustering
- Constrained clustering
- Data stream clustering
- HCS clustering
- Sequence clustering
- Spectral clustering

Techniques used in cluster analysis

- Artificial neural network (ANN)
- Nearest neighbor search
- Neighbourhood components analysis
- Latent class analysis
- Affinity propagation

Data projection and preprocessing

- Dimension reduction
- Principal component analysis
- Multidimensional scaling

Other

- Cluster-weighted modeling
- Curse of dimensionality
- Determining the number of clusters in a data set

- Parallel coordinates
- Structured data analysis

References

1. Bailey, Ken (1994). "Numerical Taxonomy and Cluster Analysis". *Typologies and Taxonomies*. p. 34. ISBN 9780803952591.
2. Tryon, Robert C. (1939). *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality*. Edwards Brothers.
3. Cattell, R. B. (1943). "The description of personality: Basic traits resolved into clusters". *Journal of Abnormal and Social Psychology*. **38** (4): 476–506. doi:10.1037/h0054116 (https://doi.org/10.1037%2Fh0054116).
4. Estivill-Castro, Vladimir (20 June 2002). "Why so many clustering algorithms — A Position Paper". *ACM SIGKDD Explorations Newsletter*. **4** (1): 65–75. doi:10.1145/568574.568575 (https://doi.org/10.1145%2F568574.568575).
5. Everitt, Brian (2011). *Cluster analysis*. Chichester, West Sussex, U.K: Wiley. ISBN 9780470749913.
6. Sibson, R. (1973). "SLINK: an optimally efficient algorithm for the single-link cluster method" (http://www.cs.gsu.edu/~wkim/index_files/papers/sibson.pdf) (PDF). *The Computer Journal*. British Computer Society. **16** (1): 30–34. doi:10.1093/comjnl/16.1.30 (https://doi.org/10.1093%2Fcomjnl%2F16.1.30).
7. Defays, D. (1977). "An efficient algorithm for a complete link method". *The Computer Journal*. British Computer Society. **20** (4): 364–366. doi:10.1093/comjnl/20.4.364 (https://doi.org/10.1093%2Fcomjnl%2F20.4.364).
8. Lloyd, S. (1982). "Least squares quantization in PCM". *IEEE Transactions on Information Theory*. **28** (2): 129–137. doi:10.1109/TIT.1982.1056489 (https://doi.org/10.1109%2FTIT.1982.1056489).
9. Kriegel, Hans-Peter; Kröger, Peer; Sander, Jörg; Zimek, Arthur (2011). "Density-based Clustering" (http://wires.wiley.com/WileyCDA/WiresArticle/wisId-WIDM30.html). *WIREs Data Mining and Knowledge Discovery*. **1** (3): 231–240. doi:10.1002/widm.30 (https://doi.org/10.1002%2Fwidm.30).
10. Microsoft academic search: most cited data mining articles (http://academic.research.microsoft.com/CSDirectory/paper_category_7.htm) Archived (https://web.archive.org/web/20100421170848/http://academic.research.microsoft.com/CSDirectory/Paper_category_7.htm) 2010-04-21 at the Wayback Machine.: DBSCAN is on rank 24, when accessed on: 4/18/2010
11. Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press. pp. 226–231. CiteSeerX 10.1.1.71.1980 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.1980). ISBN 1-57735-004-9.
12. Ankerst, Mihael; Breunig, Markus M.; Kriegel, Hans-Peter; Sander, Jörg (1999). "OPTICS: Ordering Points To Identify the Clustering Structure". *ACM SIGMOD international conference on Management of data*. ACM Press. pp. 49–60. CiteSeerX 10.1.1.129.6542 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.129.6542).
13. Achtert, E.; Böhm, C.; Kröger, P. (2006). "DeLi-Clu: Boosting Robustness, Completeness, Usability, and Efficiency of Hierarchical Clustering by a Closest Pair Ranking". *LNCS: Advances in Knowledge Discovery and Data Mining*. Lecture Notes in Computer Science. **3918**: 119–128. doi:10.1007/11731139_16 (https://doi.org/10.1007%2F11731139_16). ISBN 978-3-540-33206-0.
14. Sculley, D. (2010). *Web-scale k-means clustering*. Proc. 19th WWW.
15. Huang, Z. (1998). "Extensions to the *k*-means algorithm for clustering large data sets with categorical values". *Data Mining and Knowledge Discovery*. **2**: 283–304.
16. R. Ng and J. Han. "Efficient and effective clustering method for spatial data mining". In: Proceedings of the 20th VLDB Conference, pages 144-155, Santiago, Chile, 1994.
17. Tian Zhang, Raghu Ramakrishnan, Miron Livny. "An Efficient Data Clustering Method for Very Large Databases." In: Proc. Int'l Conf. on Management of Data, ACM SIGMOD, pp. 103–114.
18. Can, F.; Ozkarahan, E. A. (1990). "Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases". *ACM Transactions on Database Systems*. **15** (4): 483–517. doi:10.1145/99935.99938 (https://doi.org/10.1145%2F99935.99938).
19. Kriegel, Hans-Peter; Kröger, Peer; Zimek, Arthur (July 2012). "Subspace clustering". *Wiley Interdisciplinary Reviews: Data*

19. Kriegel, Hans-Peter; Kröger, Peer; Zimek, Arthur (July 2012). "Subspace clustering". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. **2** (4): 351–364. doi:10.1002/widm.1057 (https://doi.org/10.1002%2Fwidm.1057).
20. Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. (2005). "Automatic Subspace Clustering of High Dimensional Data". *Data Mining and Knowledge Discovery*. **11**: 5–33. doi:10.1007/s10618-005-1396-1 (https://doi.org/10.1007%2Fs10618-005-1396-1).
21. Karin Kailing, Hans-Peter Kriegel and Peer Kröger. *Density-Connected Subspace Clustering for High-Dimensional Data*. In: *Proc. SIAM Int. Conf. on Data Mining (SDM'04)*, pp. 246-257, 2004.
22. Achtert, E.; Böhm, C.; Kriegel, H.-P.; Kröger, P.; Müller-Gorman, I.; Zimek, A. (2006). "Finding Hierarchies of Subspace Clusters". *LNCS: Knowledge Discovery in Databases: PKDD 2006*. Lecture Notes in Computer Science. **4213**: 446–453. doi:10.1007/11871637_42 (https://doi.org/10.1007%2F11871637_42). ISBN 978-3-540-45374-1.
23. Achtert, E.; Böhm, C.; Kriegel, H. P.; Kröger, P.; Müller-Gorman, I.; Zimek, A. (2007). "Detection and Visualization of Subspace Cluster Hierarchies". *LNCS: Advances in Databases: Concepts, Systems and Applications*. Lecture Notes in Computer Science. **4443**: 152–163. doi:10.1007/978-3-540-71703-4_15 (https://doi.org/10.1007%2F978-3-540-71703-4_15). ISBN 978-3-540-71702-7.
24. Achtert, E.; Böhm, C.; Kröger, P.; Zimek, A. (2006). "Mining Hierarchies of Correlation Clusters". *Proc. 18th International Conference on Scientific and Statistical Database Management (SSDBM)*: 119–128. doi:10.1109/SSDBM.2006.35 (https://doi.org/10.1109%2FSSDBM.2006.35). ISBN 0-7695-2590-3.
25. Böhm, C.; Kailing, K.; Kröger, P.; Zimek, A. (2004). "Computing Clusters of Correlation Connected objects". *Proceedings of the 2004 ACM SIGMOD international conference on Management of data - SIGMOD '04*. p. 455. doi:10.1145/1007568.1007620 (https://doi.org/10.1145%2F1007568.1007620). ISBN 1581138598.
26. Achtert, E.; Bohm, C.; Kriegel, H. P.; Kröger, P.; Zimek, A. (2007). "On Exploring Complex Relationships of Correlation Clusters". *19th International Conference on Scientific and Statistical Database Management (SSDBM 2007)*. p. 7. doi:10.1109/SSDBM.2007.21 (https://doi.org/10.1109%2FSSDBM.2007.21). ISBN 0-7695-2868-6.
27. Meilă, Marina (2003). "Comparing Clusterings by the Variation of Information". *Learning Theory and Kernel Machines*. Lecture Notes in Computer Science. **2777**: 173–187. doi:10.1007/978-3-540-45167-9_14 (https://doi.org/10.1007%2F978-3-540-45167-9_14). ISBN 978-3-540-40720-1.
28. Kraskov, Alexander; Stögbauer, Harald; Andrzejak, Ralph G.; Grassberger, Peter (1 December 2003). "Hierarchical Clustering Based on Mutual Information". arXiv:q-bio/0311039 (https://arxiv.org/abs/q-bio/0311039).
29. Auffarth, B. (July 18–23, 2010). "Clustering by a Genetic Algorithm with Biased Mutation Operator". *Wcci Cec*. IEEE. CiteSeerX 10.1.1.170.869 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.170.869).
30. Frey, B. J.; Dueck, D. (2007). "Clustering by Passing Messages Between Data Points". *Science*. **315** (5814): 972–976. Bibcode:2007Sci...315..972F (http://adsabs.harvard.edu/abs/2007Sci...315..972F). doi:10.1126/science.1136800 (https://doi.org/10.1126%2Fscience.1136800). PMID 17218491 (https://www.ncbi.nlm.nih.gov/pubmed/17218491).
31. Pfitzner, Darius; Leibbrandt, Richard; Powers, David (2009). "Characterization and evaluation of similarity measures for pairs of clusterings". *Knowledge and Information Systems*. Springer. **19**: 361–394. doi:10.1007/s10115-008-0150-6 (https://doi.org/10.1007%2Fs10115-008-0150-6).
32. Feldman, Ronen; Sanger, James (2007-01-01). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge Univ. Press. ISBN 0521836573. OCLC 915286380 (https://www.worldcat.org/oclc/915286380).
33. Weiss, Sholom M.; Indurkhya, Nitin; Zhang, Tong; Damerau, Fred J. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer. ISBN 0387954333. OCLC 803401334 (https://www.worldcat.org/oclc/803401334).
34. Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich. *Introduction to Information Retrieval*. Cambridge University Press. ISBN 978-0-521-86571-5.
35. Dunn, J. (1974). "Well separated clusters and optimal fuzzy partitions". *Journal of Cybernetics*. **4**: 95–104. doi:10.1080/01969727408546059 (https://doi.org/10.1080%2F01969727408546059).
36. Färber, Ines; Günnemann, Stephan; Kriegel, Hans-Peter; Kröger, Peer; Müller, Emmanuel; Schubert, Erich; Seidl, Thomas; Zimek, Arthur (2010). "On Using Class-Labels in Evaluation of Clusterings" (http://eecs.oregonstate.edu/research/multiclust/Evaluation-4.pdf) (PDF). In Fern, Xiaoli Z.; Davidson, Ian; Dy, Jennifer. *MultiClust: Discovering, Summarizing, and Using Multiple Clusterings*. ACM SIGKDD.
37. Pourrajabi, M.; Moulavi, D.; Campello, R. J. G. B.; Zimek, A.; Sander, J.; Goebel, R. (2014). "Model Selection for Semi-Supervised Clustering". *Proceedings of the 17th International Conference on Extending Database Technology (EDBT)*,. pp. 331–

342. doi:10.5441/002/edbt.2014.31 (https://doi.org/10.5441%2F002%2Fedbt.2014.31).

38. Rand, W. M. (1971). "Objective criteria for the evaluation of clustering methods". *Journal of the American Statistical Association*. American Statistical Association. **66** (336): 846–850. doi:10.2307/2284239 (https://doi.org/10.2307%2F2284239). JSTOR 2284239 (https://www.jstor.org/stable/2284239).
39. E. B. Fowlkes & C. L. Mallows (1983), "A Method for Comparing Two Hierarchical Clusterings", Journal of the American Statistical Association 78, 553–569.
40. Powers, David (2003). *Recall and Precision versus the Bookmaker*. International Conference on Cognitive Science. pp. 529–534.
41. Arabie, P. "Comparing partitions". *Journal of Classification*. **2** (1): 1985.
42. Wallace, D. L. (1983). "Comment". *Journal of the American Statistical Association*. **78** (383): 569–579. doi:10.1080/01621459.1983.10478009 (https://doi.org/10.1080%2F01621459.1983.10478009).
43. Powers, David (2012). *The Problem with Kappa*. European Chapter of the Association for Computational Linguistics. pp. 345–355.
44. Hopkins, Brian; Skellam, John Gordon (1954). "A new method for determining the type of distribution of plant individuals". *Annals of Botany*. Annals Botany Co. **18** (2): 213–227. doi:10.1093/oxfordjournals.aob.a083391 (https://doi.org/10.1093%2Foxfordjournals.aob.a083391).
45. Banerjee, A. (2004). "Validating clusters using the Hopkins statistic". *IEEE International Conference on Fuzzy Systems*. **1**: 149–153. doi:10.1109/FUZZY.2004.1375706 (https://doi.org/10.1109%2FFUZZY.2004.1375706). ISBN 0-7803-8353-2.
46. Filipovych, Roman; Resnick, Susan M.; Davatzikos, Christos (2011). "Semi-supervised Cluster Analysis of Imaging Data" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3008313). *NeuroImage*. **54** (3): 2185–2197. doi:10.1016/j.neuroimage.2010.09.074 (https://doi.org/10.1016%2Fj.neuroimage.2010.09.074). PMC 3008313 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3008313). PMID 20933091 (https://www.ncbi.nlm.nih.gov/pubmed/20933091).
47. Bewley, A., & Upcroft, B. (2013). Advantages of Exploiting Projection Structure for Segmenting Dense 3D Point Clouds. In Australian Conference on Robotics and Automation [1] (http://www.araa.asn.au/acra/acra2013/papers/pap148s1-file1.pdf)
48. Bewley, A.; et al. "Real-time volume estimation of a dragline payload". *IEEE International Conference on Robotics and Automation*. **2011**: 1571–1576.
49. Basak, S.C.; Magnuson, V.R.; Niemi, C.J.; Regal, R.R. (1988). "Determining Structural Similarity of Chemicals Using Graph Theoretic Indices". *Discr. Appl. Math*. **19**: 17–44. doi:10.1016/0166-218x(88)90004-2 (https://doi.org/10.1016%2F0166-218x%2888%2990004-2).
50. Huth, R.; et al. (2008). "Classifications of Atmospheric Circulation Patterns: Recent Advances and Applications". *Ann. N.Y. Acad. Sci*. **1146**: 105–152.

External links

- Data Mining (https://curlie.org/Computers/Software/Databases/Data_Mining) at Curlie (based on DMOZ)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=833494059"

This page was last edited on 31 March 2018, at 22:20.