Review on Semantic Document Clustering

SK Ahammad Fahad[1], Wael M.S. Yafooz[2]

[1]*Faculty of Computer and Information Technology, Al-Madinah International University (MEDIU), Malaysia, fahad.wasd@gmail.com*

[2]*Faculty of Computer and Information Technology, Al-Madinah International University (MEDIU), Malaysia, Wael.mohamed@mediu.edu.my*

Received 27 February 2017; accepted 21 March 2017

Abstract

Now the age of information technology, the textual document is spontaneously increasing over online or offline. In those articles contain Product information to a company profile. A lot of sources generate valuable information into text in the medical report, economic analysis, scientific journals, news, blog etc. Maintain and access those documents are very difficult without proper classification. Those problems can be overcome by proper document classification. Only a few documents are classified. All need classification and those are unsupervised. In this context clustering is the only solution. Traditional clustering technique and textual clustering have some difference. Relations between words are very imported to do clustering. Semantic clustering is proven as more appropriate clustering technique for texts. In this review paper, there has valuable information about clustering to semantic document clustering technique. In this paper, there has some information provided about advantage and disadvantage for various clustering methods.

Keywords: (clustering, Semantic clustering, Conceptual Clustering, Cobweb Algorithm)

1. Introduction

A substantial portion of the on the market information is unbroken in Text databases, that accommodates huge collections of documents from varied sources, like news articles, analysis papers, books, digital libraries, e-mail messages, and sites. Text documents area unit growing speedily as a result of the increasing amount of data on the market among the electronic and digitized kind, like electronic publications, varied types of electronic documents, e-mail, and additionally the globe Wide web. Lately, most of the info regarding government, industry, business, and different institutions area unit keep electronically, among the sort of text databases. Most of the text databases area unit semi-structured info that they are neither totally structured. Huge document corpus may afford lots of useful information to parents. However, it's to boot a challenge to hunt out the useful information from an outsized form of documents. notably with the explode of knowledge around the cyber-world, company and organizations demand economical and effective ways in which to rearrange the huge document corpus and build later navigating and browsing become less complicated, friendly and economical. An evident characteristic of enormous document corpus is that the massive volumes of documents. It's nearly impossible for someone to flick through all the documents and establish the relative for a selected topic. The thanks to organizing huge document corpus are that the drawback there has a tendency to tend to concern. As text information area unit inherently unstructured, some researchers applied the varied technique for document management. Researchers have conferred info discovery in text system, which uses the sole information extraction to induce fascinating information and data from unstructured text assortment. Ancient agglomeration methods are not effective on matter agglomeration. When you cluster language you've got have to be compelled to elect Associate in nursing awfully capable agglomeration methodology that will build Associate in nursing economic agglomeration on language.

2. Classification and Clustering

Clustering and classification appear some close processes; there have a difference between clustering and classification based on their meaning.

Table 1
Difference between Clustering and Classification

Definition	Clustering	Clustering is an unattended learning technique accustomed cluster similar instances on the premise of options
	Classification	Classification may be a supervised learning technique accustomed assign predefined tags to instances on the premise of options
Supervision	Clustering	Clustering is AN unattended learning technique
	Classification	Classification may be a supervised learning technique
Training Set	Clustering	A coaching set isn't employed in clump
	Classification	A coaching set is employed to search out similarities in classification
Process	Clustering	Statistical ideas square measure used, and datasets square measure split into subsets with similar options
	Classification	Classification uses the algorithms to categorize the new knowledge in step with the observations of the coaching

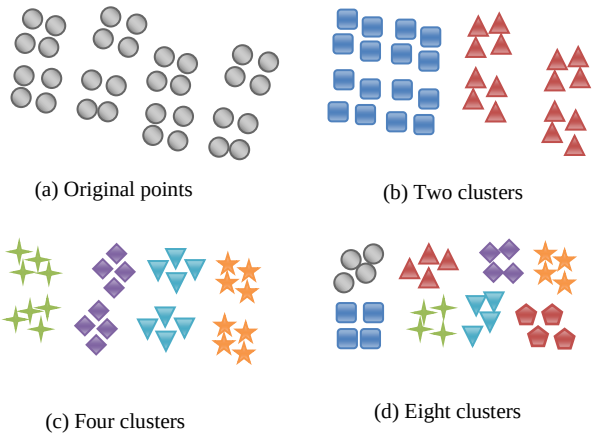
		set
Labels	Clustering	There are not any labels in clump
	Classification	There square measure labels for a few points
Aim	Clustering	The aim of clump is, grouping a group of objects so as to search out whether or not there's any relationship between them
	Classification	The aim of the clump is to search out that category a replacement object belongs to from the set of predefined categories.

In the data processing world, agglomeration and classification square measure 2 sorts of learning ways. Agglomeration and classification seem some shut processes; there encompasses a distinction between agglomeration and classification supported their which means. Within the data processing world, agglomeration and classification square measure 2 sorts of learning ways. Each these ways characterize objects into teams by one or a lot of options. The key distinction between agglomeration and classification is that; agglomeration could be an unsupervised learning technique won't to cluster similar instances on the idea of options whereas classification is a supervised learning technique won't to assign predefined tags to instances on the idea of options.

3. Clustering

In today's extremely competitive business surroundings, clump plays a very important role. The clump could be an important task in knowledge method that's used for the aim to make groups or clusters of the given data set supported the similarity between them. The clump could be an important conception to unite objects in groups (clusters) in line with their similarity. The clump is comparable to classification except that the teams don't seem to be predefined, however rather outlined by the info alone.

Cluster analysis is one amongst the foremost necessary data processing strategies. It's a central downside in information management. Document clump is that the act of grouping similar documents into categories, wherever similarity is a few performs on a document. Document clump wouldn't like separate coaching method or manual tagging cluster earlier. It's the strategy of partitioning or grouping a given set of patterns into disjoint clusters. The documents within the same clusters square measure a lot of similar, whereas the documents in several cluster square measure a lot of dissimilar.



(a) Original points (b) Two clusters

(c) Four clusters (d) Eight clusters

Figure: 1: Data clustering

Data agglomeration may be an information exploration technique that permits objects with similar characteristics to be sorted along so as to facilitate their more process. Most of the initial clump techniques were developed by statistics or pattern recognition communities [1], where the goal was to cluster a modest kind of data instances. In further recent years, clump was referred to as a key technique in processing tasks. This basic operation is also applied to many common tasks like unsupervised classification, segmentation, and dissection. Within the unsupervised technique, the right answers aren't famed or simply not told to the network.

3.1 Type of Clustering

Different approaches to clump data area unit delineate with the help of the hierarchy. At the very best level, there is a distinction between graded and partitional approaches. Graded ways in which manufacture a nested series of partitions, whereas partitional ways in which manufacture only 1. There have various ways in which of taxonometric representations of a clump. Ours depends on the discussion in religious belief and Dubes [2]. At the very best level, there is a distinction between graded and partitional approaches. Graded ways in which manufacture a nested series of partitions, on the other hand, partitional ways in which manufacture only one.

The taxonomy is ought to be supplemented by a discussion of cross-cutting issues which can have a bearing on all of the varied approaches despite their placement inside the taxonomy.

Agglomerative vs. divisive; this facet relates to algorithmic structure and operation. Associate in nursing agglomerate approach begins with each pattern in AN extremely distinct cluster, and successively, merges clusters on until a stopping criterion is happy. A discordant methodology begins with all patterns in a single cluster and performs squawky until a stopping criterion is met.

Monothetic vs. polythetic; this facet relates to the sequent or concurrent use of choices inside the agglomeration technique. Most algorithms are polythetic; that is, all choices enter into the computation of distances between patterns, and selections are supported those distances. A simple monothetic algorithm considers choices consecutive to divide the given assortment of patterns.

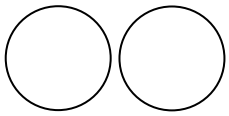
Hard vs. fuzzy; a troublesome agglomeration algorithm allocates each pattern to one cluster throughout its operation and in its output. A fuzzy agglomeration methodology assigns degrees of membership in several clusters to each input pattern. A fuzzy agglomeration is additionally regenerating to a hard agglomeration by distribution every pattern to the cluster with the foremost necessary period of time of membership.

Deterministic vs. stochastic; this issue is most relevant to partitional approaches designed to optimize a SQL Error operates. These improvements are accomplished practice

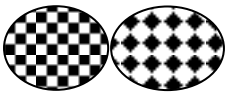
ancient techniques or through a random search of the state space consisting of all double labelling.
Incremental Vs. Non-Incremental; This issue arises, once the pattern set to be clustered is huge, and constraints on execution time or memory house have a bearing on the look of the rule. the first history of clump methodology doesn't contain several samples of clump algorithms designed to figure with huge data sets, however the arrival of knowledge mining has fostered the event of clump algorithms that minimize the number of scans through the pattern set, cut back a number of patterns examined throughout execution, or cut back the dimensions of knowledge structures used among the algorithm's operations.

3.2 Type of Cluster

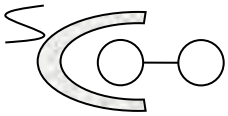
Clustering aims to find useful groups of objects (clusters), where usefulness is defined by the goals of the data analysis. Not surprisingly, there are several different notions of a cluster that prove useful in practice. In order to visually illustrate the differences among these types of clusters, there have two-dimensional points, as shown in Figure 2, as our data objects. Its stress, however, that the types of clusters described here are equally valid for other kinds of data.



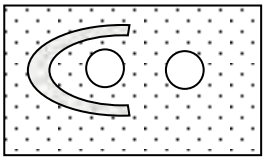
(a) Well-separated clusters. Every purpose is nearer to any or all of the purposes in its cluster than to any point in another cluster.



(b) Center-based clusters. Every purpose is nearer to the middle of its cluster than to the middle of the other cluster.



(c) Contiguity-based clusters. Every purpose is nearer to a minimum of one purpose in its cluster than to any purpose in another cluster.



(d) Density-based clusters. The cluster is regions of high density separated by regions of tenacity.

(e) Conceptual clusters. Points during a cluster share some general property that derives from the whole set of points. Points within the intersection of the circles belong to each.

Figure: 2: Different types of a cluster as illustrated by sets of two-dimensional points.

Well-Separated: A cluster might be a collection of objects among which each object is nearer (or plenty of similar) to every totally different object among the cluster than to any object not among the cluster. Generally a threshold is utilized to specify that everybody the objects throughout a cluster ought to be sufficiently shut (or similar) to a minimum of each other. This idealistic definition of a cluster is glad solely the knowledge contains natural clusters that unit quite faraway from each other. Figure 2(a) offers associate degree associate example of well-separated clusters that consists of two groups of points throughout a two-dimensional space. The gap between any two points in varied groups is larger than the gap between any two points within a gaggle. Well-separated clusters ought to not be circular, however, can have any type.
Prototype-Based: A cluster could also be a collection of objects throughout which each object is nearer (more similar) to the paradigm that defines the cluster than to the paradigm of the opposite cluster. For data with continuous attributes, the paradigm of a cluster is typically a middle of mass, i.e., the everyday (mean) of all the points among the cluster. Once a middle of mass is not meaty like once the information has categorical attributes, the paradigm is typically a medoid, i.e., the foremost representative purpose of a cluster. For many styles of data, the paradigm is thought to be the foremost central purpose, and in such instances, there have a tendency to tend to normally check with prototype- primarily based clusters as centre-based clusters. Not surprisingly, such clusters tend to be the world. Figure 2(b) shows Associate in nursing example of centre-based clusters
Graph-Based: If the knowledge is painted as a graph, where the nodes are objects and thus the links represent connections among objects, then a cluster are typically printed as a connected component; i.e., a bunch of objects that are connected to a minimum of each other, but that do not have any association to things outside the cluster. Necessary samples of graph-based clusters are contiguity-based clusters, where a pair of objects are connected given that they are within such distance of each completely different. Figure 2(c) shows degree example of such clusters for two-dimensional points. This definition of a cluster is useful once clusters are irregular or tangled but can have trouble once the noise is that the gift since, as illustrated by two spherical clusters of figure 2.4(c), a little low bridge of points can merge a pair of distinct clusters.
Density-Based: A cluster may well be a dense region of objects that is swallowed by a section of pertinacity. Figure 2.4(d) shows some density-based clusters for information created by adding noise to the information of Figure 2.4(c). The two circular clusters do not appear to be united, as in Figure 2(c), as a result of the bridge between them fades into the noise. Likewise, the curve that is the gift in Figure 2(c) jointly fades into the noise and does not reasonably the cluster in Figure 2(d). A density- based definition of a cluster is typically used once the clusters unit of measurement irregular or tangled, and once noise and outliers unit of measurement gift. Against this, a contiguity based definition of a cluster would not work well for the

information of Figure 2(d) since the noise would tend to make bridges between clusters.

3.3 Partitional Clustering

Partitional ways that need to be equipped a gaggle of initial seeds (or clusters) that area unit then improved iteratively. Hierarchical ways that, on the other hand, can pop out with the individual data points in single clusters and build the clump. The role of the gap metric is to boot utterly totally different in every of these algorithms. In hierarchical clump, the gap metric is at the beginning applied on the data points at rock bottom level, therefore, additional and additional applied on sub-clusters by choosing absolute representative points for the sub-clusters. However, inside the case of partitional ways that, in general, the representative purposes chosen at utterly totally different iterations may be a virtual point just like the centre of mass of the cluster (which is non-existent inside the data).

In distinction to graded agglomeration, that yields a sequent level of clusters by unvarying fusions or divisions, partitional agglomeration assign a gaggle of data points into K clusters with a better-known organization. This methodology usually accompanies the advance of a criterion performs. plenty of specifically, given a gaggle of data, points $x_i \in R^d, i = 1, \ldots\ldots, N$ partitional clump algorithms aim to rearrange them into K clusters $\{C_1, \ldots\ldots\ldots, C_k\}$ whereas maximizing or minimizing a pre-specified criterion perform f. in theory, the optimum partition, supported the criterion perform f, are found by enumerating all prospects. However, this brute force technique is infeasible in following due to the terribly expensive computation, as given by the formula [3]:

$$P(N,K) = \frac{1}{K!}\sum_{m=1}^{K}(-1)^{K-m}C_K^m m^n$$

Obviously, even for a little - scale bunch drawback, a simple enumeration isn't attainable. As associate in nursing example, thus on cluster thirty objects into three clusters, a number of potential partitions are roughly 2×10^{14}. Therefore, heuristic algorithms explore for approximate solutions.

3.3.1Partitional Clustering Algorithms

The first partitional clustering algorithm that will be discussed in this section is the K-Means clustering algorithm. Some of the widely used variations of K-Means will also be discussed in this section. It is one of the simplest and most efficient clustering algorithms proposed in the literature of data clustering.

K-means clustering [4, 5] is the most widely used partitional clustering algorithm. It starts by choosing K representative points as the initial centroids. Each point is then assigned to the closest centroid based on a particular proximity measure chosen. Once the clusters are formed, the centroids for each cluster are updated. The algorithm then iteratively repeats these two steps until the centroids do not change or any other alternative relaxed convergence criterion is met. K-means clustering is a greedy algorithm which is guaranteed to converge to a local minimum but the minimization of its score function is known to be NP-Hard [6]. Typically, the convergence condition is relaxed and a weaker condition may be used. In practice, it follows the rule that the iterative procedure must be continued until 1% of the points change their cluster memberships.

Algorithm: K-Means Clustering;

1: Select K points as initial centroids.
2: repeat
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: until convergence criterion is met.

The major factors that can impact the performance of the K-means algorithm are the following:

1. Choosing the initial centroids.
2. Estimating the number of clusters K.

The simple framework of the K-means algorithm makes it very flexible to modify and build more efficient algorithms on top of it. Some of the variations proposed to the K-means algorithm are based on;

(i) Choosing different representative prototypes for the clusters (K-medoids, K-medians, K-modes),
(ii) Choosing better initial centroid estimates (Intelligent K-means, Genetic K-means), and
(iii) Applying some kind of feature transformation technique (Weighted K-means, Kernel Kmeans). In this section, there have a discussion, the most prominent variants of K-means clustering that have been proposed in the literature of partitional clustering.

Some of the variations proposed to the K-means algorithm are;

- K-Medoids Clustering
- K-Medians Clustering
- K-Modes Clustering
- Fuzzy K-Means Clustering
- X-Means Clustering
- Intelligent K-Means Clustering
- Bisecting K-Means Clustering
- Kernel K-Means Clustering
- Mean Shift Clustering
- Weighted K-Means Clustering
- Genetic K-Means Clustering

It is believed that the K-means clustering algorithm consumes a lot of time in its later stages when the centroids are close to their final locations but the algorithm is yet to converge. An improvement to the original Lloyd's K-means clustering using a kd-tree data structure to store the data points was proposed in [7]. This algorithm is called the filtering algorithm where for each node a set of candidate centroids is maintained similar to a normal kd-tree. These candidate set centroids are pruned based on a distance comparison which measures the proximity to the midpoint of the cell. This filtering algorithm runs faster when the

separation between the clusters increases. In the K-means clustering algorithm, usually there are several redundant calculations that are performed. For example, when a point is very far from a particular centroid, calculating its distance to that centroid may not be necessary. The same applies for a point which is very close to the centroid as it can be directly assigned to the centroid without computing its exact distance. An optimized K-means clustering method which uses the triangle inequality metric is also proposed to reduce the number of distance metric calculations [8].

3.4 Hierarchical Clustering

Clustering techniques are generally classified as partitional clustering and hierarchical clustering, based on the properties of the generated clusters [9, 10, 11, 12]. Partitional clustering directly divides data points into some pre-specified number of clusters without the hierarchical structure, while hierarchical clustering groups data with a sequence of nested partitions, either from singleton clusters to a cluster including all individuals or vice versa. The former is known as agglomerative hierarchical clustering, and the latter is called divisive hierarchical clustering.

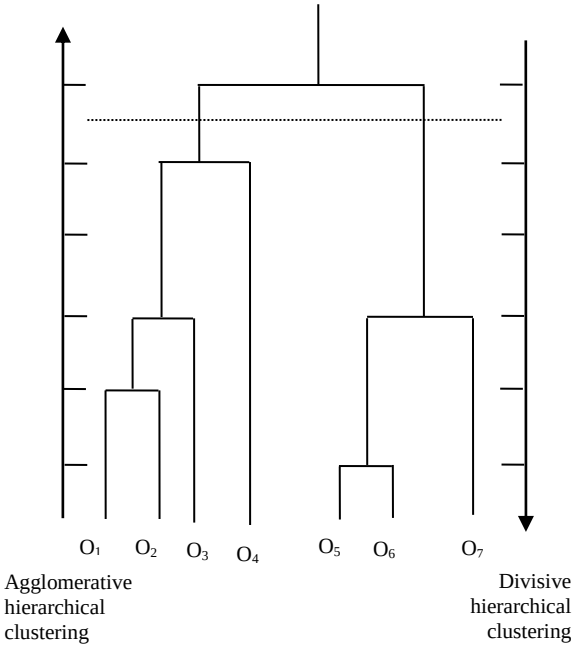


Figure. 3: Example of a dendrogram from hierarchical clustering. The clustering direction for the divisive hierarchical clustering is opposite to that of the agglomerative hierarchical clustering. Two clusters are obtained by cutting the dendrogram at an appropriate level.

Both agglomerative and divisive clustering methods organize data into the hierarchical structure based on the proximity matrix. The results of hierarchical clustering are usually depicted by a binary tree or dendrogram, as depicted in Fig. 3. The root node of the dendrogram represents the whole data set, and each leaf node is regarded as a data point. The intermediate nodes thus describe the extent to which the objects are proximal to each other; and the height of the dendrogram usually expresses the distance between each pair of data points or clusters, or a data point and a cluster. The ultimate clustering results can be obtained by cutting the dendrogram at different levels (the dashed line in Fig. 3).
This representation provides very informative descriptions and a visualization of the potential data clustering structures, especially when real hierarchical relations exist in the data, such as the data from evolutionary research on different species of organisms, or other applications in medicine, biology, and archaeology [13] [14].

3.4.1 Hierarchical Clustering Algorithms

Hierarchical clustering algorithms [15] were developed to overcome some of the disadvantages associated with flat or partitional based clustering methods. Partitional methods generally require a user predefined parameter K to obtain a clustering solution and they are often nondeterministic in nature. Hierarchical algorithms were developed to build a more deterministic and flexible mechanism for clustering the data objects. Hierarchical methods can be categorized into agglomerative and divisive clustering methods. Agglomerative methods start by taking singleton clusters (that contain only one data object per cluster) at the bottom level and continue merging two clusters at a time to build a bottom-up hierarchy of the clusters. Divisive methods, on the other hand, start with all the data objects in a huge macro-cluster and split it continuously into two groups generating a top-down hierarchy of clusters.
A cluster hierarchy here can be interpreted using the standard binary tree terminology as follows. The root represents all the sets of data objects to be clustered and this forms the apex of the hierarchy (level 0). At each level, the child entries (or nodes) which are subsets of the entire dataset correspond to the clusters. The entries in each of these clusters can be determined by traversing the tree from the current cluster node to the base singleton data points. Every level in the hierarchy corresponds to some set of clusters. The base of the hierarchy consists of all the singleton points which are the leaves of the tree. This cluster hierarchy is also called a dendrogram. The basic advantage of having a hierarchical clustering method is that it allows for cutting the hierarchy at any given level and obtaining the clusters correspondingly. This feature makes it significantly different from partitional clustering methods in that it does not require a predefined user specified parameter k (number of clusters).
The basic steps involved in an agglomerative hierarchical clustering algorithm are the following. First, using a particular proximity measure a dissimilarity matrix is constructed and all the data points are visually represented at the bottom of the dendrogram. The closest sets of clusters are merged at each level and then the dissimilarity matrix is updated correspondingly. This process of agglomerative merging is carried on until the final maximal cluster (that contains all the data objects in a single cluster) is obtained. This would represent the apex of our dendrogram and mark the completion of the merging process. In this part there have a discussion about the

different kinds of proximity measures which can be used in agglomerative hierarchical clustering.

Single and Complete Link The most popular agglomerative clustering methods are single link and complete link clustering. In single link clustering [16, 17], the similarity of two clusters is the similarity between their most similar (nearest neighbour) members. This method intuitively gives more importance to the regions where clusters are closest, neglecting the overall structure of the cluster. Hence, this method falls under the category of a local similarity-based clustering method. Because of its local behaviour, single linkage is capable of effectively clustering non elliptical, elongated shaped groups of data objects. However, one of the main drawbacks of this method is its sensitivity to noise and outliers in the data. Complete link clustering [18] measures the similarity of two clusters as the similarity of their most dissimilar members. This is equivalent to choosing the cluster pair whose merge has the smallest diameter. As this method takes the cluster structure into consideration it is nonlocal in behaviour and generally obtains compact shaped clusters. However, similar to single link clustering, this method is also sensitive to outliers. Both single link and complete link clustering have their graph theoretic interpretations [19], where the clusters obtained after single link clustering would correspond to the connected components of a graph and those obtained through complete link would correspond to the maximal cliques of the graph.

Divisive Clustering Divisive hierarchical clustering is a top-down approach where the procedure starts at the root with all the data points and recursively splits it to build the dendrogram. This method has the advantage of being more efficient compared to agglomerative clustering especially when there is no need to generate a complete hierarchy all the way down to the individual leaves. It can be considered as a global approach since it contains the complete information before splitting the data. Now in this section, there have a discussion about the factors that affect the performance of divisive hierarchical clustering.

- Splitting criterion: The Ward's K-means square error criterion is used here. The greater reduction obtained in the difference in the SSE criterion should reflect the goodness of the split. Since the SSE criterion can be applied to numerical data only, Gini index (which is widely used in decision tree construction in classification) can be used for handling the nominal data.
- Splitting method: The splitting method used to obtain the binary split of the parent node is also critical since it can reduce the time taken for evaluating the Ward's criterion. The Bisecting K-means approach can be used here (with K = 2) to obtain good splits since it is based on the same criterion of maximizing the Ward's distance between the splits.
- Choosing the cluster to split: The choice of cluster chosen to split may not be as important as the first two factors, but it can still be useful to choose the most appropriate cluster to further split when the goal is to build a compact dendrogram. A simple method of choosing the cluster to be split further could be done by merely checking the square errors of the clusters and splitting the one with the largest value.
- Handling noise: Since the noise points present in the dataset might result in aberrant clusters, a threshold can be used to determine the termination criteria rather splitting the clusters further.

COBWEB [20]: This is a conceptual clustering algorithm that works incrementally by updating the clusters object by object. Probabilistically described clusters are arranged as a tree to form a hierarchical clustering known as probabilistic categorization tree. It handles uncertainty associated with categorical attributes in clustering through a probabilistic framework that is similar to Naive Bayes. The dendrogram in this algorithm is also called a classification tree and the nodes are referred to as concepts.

4. Textual Document Clustering

Clustering is one of the main data analysis techniques and deals with organizing a set of objects in a multidimensional space into cohesive groups, called clusters for better management and navigation [21].

Clustering is an example of unsupervised learning ,classification refers to a procedure that assigns data objects to a set of classes ,unsupervised means that clustering does not depends on predefined classes and training examples through classifying data objects[22, 23]. Document clustering is useful for many information retrieval tasks such as document browsing, organization and viewing of retrieval results [24].

Many clustering algorithms exist in the literature but difficult to provide a categorization of clustering methods because these categories may overlap, so that a method may have features from several categories, however, the major clustering methods can be classified into the following main categories hierarchical methods, partitioning methods [25].

The partitioning method attempts a flat partitioning of a collection of documents into a predefined number of disjoint clusters [26]. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from group to another, partitioning methods include k-means and k-medoids [27].

Hierarchical methods produce a sequence of nested partitions [28]. The method can be classified as being either agglomerative (bottom-up) or divisive (top-down) [29].

Document clustering is the process of grouping a set of documents into clusters so that the documents within each cluster are similar to each other, in other words, they belong to the same topic or subtopic, while documents in different clusters belong to different topics or subtopics. A document clustering algorithm is typically dependent on the use of a pair-wise distance measure between the individual documents to be clustered.

Most of techniques used in document clustering deal with a document as a bag of words without considering the

semantics of each document. A traditional algorithm mainly uses features like: words, phrases, and sequences from the documents based on counting and frequency of the features to perform clustering independent of the context [30, 31, 32, 33] .They ignores the semantics among words in documents.

4.1 Document Clustering

Text Clustering is to find out the groups information from the text documents and cluster these documents into the most relevant groups. Text clustering groups the document in an unsupervised way and there is not label or class information. Clustering methods have to discover the connections between the document and then based on these connections the documents are clustered [34, 35, 36]. Given huge volumes of documents, a good document clustering method may organize those huge numbers of documents into meaningful groups, which enable further browsing and navigation of this corpus be much easier [37]. A basic idea of text clustering is to find out which documents have many words in common and place these documents with the most words in common into same group.

Current researches efforts in document clustering started to focus on the development of a more efficient clustering with considering the semantics between terms in documents to enhance the clustering results. Text clustering aims to segregate documents into groups where a group represents certain topics that are different from other groups. From a geometrical point of view, a corpus can be seen as a set of samples on multiple manifolds, and clustering aims at grouping documents based on intrinsic structures of the manifold. Grouping of documents into clusters is an elementary step in many applications such as Indexing, Retrieval and Mining of data on the web. With a good text clustering method, a document corpus can be organized into a meaningful cluster hierarchy, which facilitates an efficient browsing and navigation of the corpus or efficient information retrieval by focusing on relevant subsets (clusters) rather than whole collections [38, 39].

All the general purpose clustering algorithms can be applied to document/text clustering. Some algorithms have been developed solely for document/text clustering. All these algorithms can be classified into partitional, hierarchical and others such as probabilistic, graph-based, and frequent term-based.

Partitional clustering attempts to break the given data set into k disjoint classes such that the data objects in a class are nearer to one another than the data objects in other classes. The most well-known and commonly used partitional clustering algorithm is K-Means [40], as well as its variances Bisecting K-Means [41] and K-Medoids [42].

Hierarchical clustering proceeds successively by building a tree of clusters. There are two types of hierarchical clustering methods: agglomerative and divisive. Agglomerative hierarchical clustering is a bottom-up strategy that starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until a user-defined criterion is met. Divisive hierarchical clustering is a top-down strategy that starts with all objects in one cluster. It divides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until certain termination conditions are satisfied. In terms of the distance/similarity measure, a hierarchical clustering could use minimum distance (single-link) [43], maximum distance (complete-link) [44], mean distance, or average distance.

Model-based clustering algorithms try to optimize the fit between the given data and some mathematical model under the assumption that the data are generated by a mixture of underlying probability distributions. SOM [45] is one of the most popular model-based algorithms that use neural network methods for clustering. It represents all points in a high-dimensional space by points in a low-dimensional (2-D or 3-D) target space, such that the distance and proximity relationship are preserved as much as possible. It assumes that there is some topology or ordering among input objects and that the points will eventually take on this structure in the target space.

Graph-based clustering algorithms apply graph theories to clustering. A well-known graph-based divisive clustering algorithm [46] is based on the construction of the minimal spanning tree (MST) of the data, and then deleting the MST edges with the largest lengths to generate clusters. Another popular graph-based clustering algorithm is MCL (Markov Cluster algorithm [47]). It will be discussed with more details later in this section.

4.2 Different Document Clustering

Based on their characteristics, text clustering can be classified into different categories.

The most common classifications are hierarchical clustering and flat clustering. Depending on when to perform clustering or how to update the result when new documents are inserted there are online clustering and offline clustering. And according to if overlap is allowed or not there are soft clustering and hard clustering. Based on the features that are used, clustering algorithms can be grouped to document-based clustering and keywords-based clustering.

4.2.1 Hierarchical and Flat Clustering

Hierarchical and flat clustering methods are two major categories of clustering algorithms. Just like departments in a company may be organized in a hierarchical style or a flat one, clusters of a document corpus may be organized in a hierarchical tree structure or in a pretty flat style.

Hierarchical Clustering: Hierarchical clustering techniques produce a nested sequence of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom [48]. The hierarchical clustering result can be viewed as an upside-down tree: the

root of the tree is the highest level of clusters, the leaves of the tree are the lowest level clusters which are the individual documents, and the branches of the tree are the intermediate level in the clustering result. Seeing from different level might get different overview of clusters. For instance in figure 4:

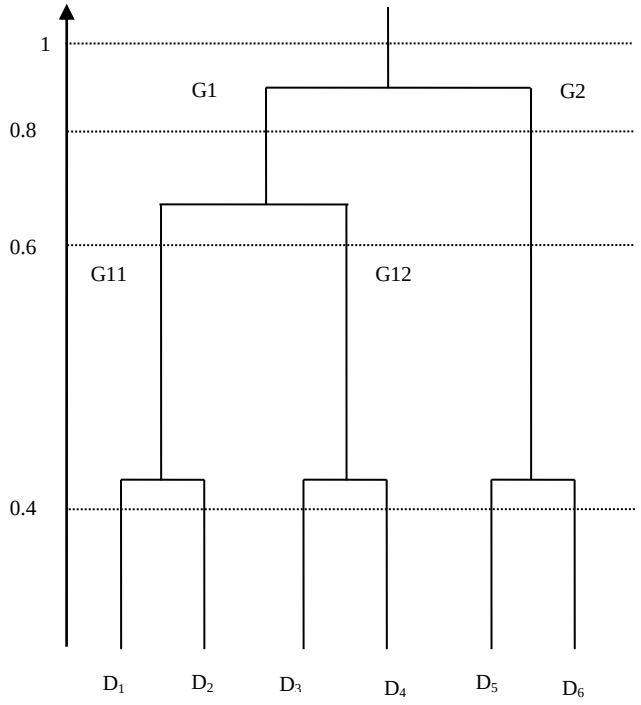


Figure. 4: Hierarchical clustering.

- If seeing from level value 1, all documents are clustered into only one group;
- If from level value *0.8*, documents are clustered into two groups *G1* and *G2*. Where G1 includes documents *D1*, *D2*, *D3* and *D4*; *G2* includes *D5* and *D6*.
- If from level value *0.6*, G1 can be divided into two sub-clusters *G11* and *G12*. And then documents are clustered into three groups *G11*, *G12* and *G2* which respectively contain *D1* and *D2*, *D3* and *D4*, *D5* and *D6*.
- When from a lower level (value *0.4*), each document denotes one cluster.

Basically there are two approaches to generate such a hierarchical clustering [49]:

- Agglomerative: Start from and leaves, and consider each document as an individual cluster at the beginning. Merge a pair of most similar clusters until only one single cluster is left.
- Divisive: Start from the root, and consider the whole document set as a single cluster. At each step divide a cluster into two (or several) sub clusters until each cluster contains exactly one document or until the required number of clusters is archived.

Agglomerative techniques are relatively more common: it is quite straightforward and most common distance calculation and similarity measurement techniques can be applied. Traditional agglomerative hierarchical clustering steps can be summarized as the following [50]:

Given a collection of documents; $D = d_1, d_2, \ldots\ldots, d_n$

1. Consider each document as an individual cluster. Compute the distance between all pairs of clusters and construct the $n \times n$ distance matrix D in which D_{ij} denotes the distance between cluster i and cluster j.
2. Merge the closest two clusters into a new cluster.
3. Update the distance matrix: calculate the distance between the new generated cluster and the rest clusters.
4. Repeat step 2 and 3 until only one single cluster remains, which is the root cluster of the hierarchy.

To generate a flat partition of clusters, a cut is made at the specific level of the hierarchical cluster tree, and on that level each branch represents a cluster and all the leaves (documents) under the same branch belong to one cluster.

Agglomerative techniques need to consider which inter-cluster similarity measures to use;

- Single-link measure: join the two clusters containing the two closest documents.
- Complete-link measure: join the two clusters with the minimum "mostdistant" pair of documents.
- Group average: join the two clusters with the minimum average document distance.

Flat Clustering: Different from hierarchical clustering, flat clustering creates one level (unnested) partitions [51] of documents instead of generating a well organized hierarchical cluster tree. Normally flat clustering techniques demand the expected number of clusters K as an input parameter, start with a random partitioning and then keep refining until algorithms converge. The convergence state is the final state that all clusters are stable and no more documents are switched between clusters.

Similarly flat clustering techniques may also create hierarchical cluster tree. By repeating the flat clustering techniques from the top level (root) of the tree to the lowest level (the leaves), a hierarchical cluster tree can be generated.

Hierarchical and flat clustering have their own advantages and weaknesses: Hierarchical clustering provides more detail about the whole document corpus, in which clusters are well organized in a tree structure. The price is the relatively higher complexity. On the contrary flat clustering techniques are normally simple and easy to implement. They could be applied with more efficiency when comparing with hierarchical clustering techniques.

When dealing with large document corpus, efficiency is the major issue there have concerned. In this thesis project there mainly consider flat clustering techniques. There have alsoevaluate a hierarchical clustering algorithm in this thesis project because hierarchical clustering might give more help in knowing the structure and relation in a large document corpus than flat clustering.

4.2.2 Online and Offline Clustering

According to when clustering is performed, clustering algorithms can be divided into online clustering algorithms and offline clustering algorithms [52].

Online clustering algorithms perform document clustering when receiving the request and return the request within a limited period. It is obvious that online clustering demands very fast operations (low complexity) and make the clustering result up-to-date. Normally online clustering algorithms are applied on small or medium corpus.

Offline clustering, on the contrary, processes the documents and groups them into relevant clusters before receiving the request. When a request is received, offline clustering algorithms perform a few simple operations and then represent the clustering result. Compared with online clustering, offline clustering performs most of the operations before receiving the requests, it is relatively complex (high complexity) and can be applied on large document corpus. The major disadvantage of offline clustering is that the clustering result is not up-to-date. Sometimes it cannot reflect the fact that if a single document or a few documents are added into the corpus before most operations are applied in a long period of time.

Online clustering and offline clustering have their different applications: the former is normally applied to group the search results and the latter is to organize the document corpus.

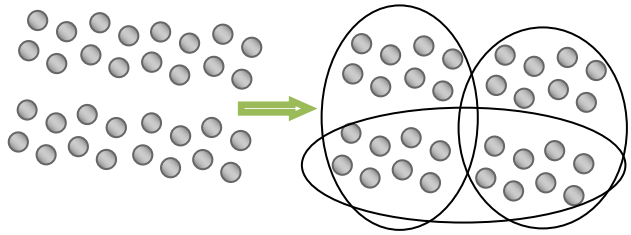
A clustering algorithm is also classified as online clustering if it only updates the necessary documents in the corpus instead of re-clustering all documents when new documents are added into the document corpus. Given an existing document corpus and the clustering result, when new documents are added into the document collection, online clustering algorithms only apply clustering calculation on the new inserted documents and a small part of the original document collection. This relatively less calculation complexity results in fast clustering speed when new documents are inserted into the document corpus occasionally and makes possible that the cluster result is up-to-date.

4.2.3 Hard and Soft Clustering

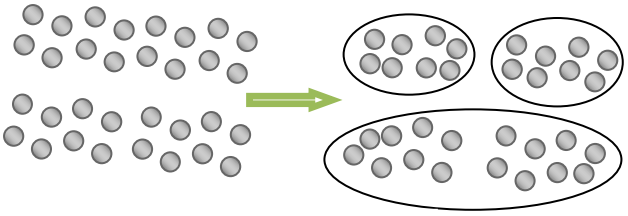
Depending on whether overlapping is allowed in the clustering result, clustering methods may generate hard clustering results or soft ones.

It is very common for one document has multiple topics, it might be tagged with multiple labels and be grouped into more than one clusters. In this scenario overlapping is allowed.

For instance, for a document which describes how scientists discovered the way bats use to “hear” flies and catch them, how this biological technique was applied to create modern radar technique and how the radar benefited to martial engineering, it is quite reasonable to say that this document can be classified into “biology”, “radar”, “martial engineering” and some other relevant classes if there are any others. So, soft clustering includes this kind of clustering algorithms which may cluster documents into different clusters and each document may belong to several clusters and keep the boundaries of the clusters “soft”. In summary with soft clustering each document is probabilistically assigned to clusters [53], just as shown in Figure 5.



(a) Soft Clustering



(b) Hard Clustering

Figure 5: Soft and Hard Clustering

However there are some situations that demand one document should only be organized to the most relevant category. This kind of clustering is called hard clustering because each document belongs to exactly one cluster. It is very important for the hard clustering algorithms to decide which cluster is the most matched one. Given the document above, a very reasonable way is to group it into the “radar” because it is mainly about the invention and the applications of radar. The idea of hard clustering can be illustrated in Figure 5.

4.2.4 Documents-based and Keyword-based Clustering

Keyword-based and document-based clustering are different in the features base on which the documents are grouped.

Document-based clustering algorithms are mainly applied on document vector space model in which every entry presents the term-weighting of term in the corresponding document. Thereby a document is mapped as a data point within an extremely high-dimensional space where each term is an axis. In this space the distance between points can be calculated and compared. Close data points can be merged and clustered into the same group; distant points are isolated into different groups. Thereby the corresponding documents are grouped or separated. As document-based clustering is based on the “document distance”, it is every important to map the documents into the right space and apply appropriate distance calculation methods.

Keyword-based clustering algorithms only choose specific document features and based on these relatively limit number of features the clusters are generated. Those specific features are chosen because they are considered as the core features between the documents and they are

shared by the similar documents and are sparse in unlike documents. Thereby how to pick up the most core feature is a very important step in keyword-based clustering.

5.2 Semantic Document Clustering

First the Clustering is one of the techniques to improve the efficiency in information retrieval for improving search and retrieval efficiency. It is a data mining tool to use for grouping objects into clusters. Clustering divides the objects (Documents) into meaningful groups based on similarity between objects. Documents within one cluster have high similarity with each other, but low similarity with documents in other clusters [55].Document clustering generates clusters from the whole document collection automatically and is used in many fields, including data mining and information retrieval. In the traditional vector space model, the unique words occurring in the document set are used as the features. But because of the synonym problem and the polysemous problem, such a bag of original words cannot represent the content of a document precisely. The growth of the World Wide Web has enticed many re-searchers to attempt to devise various methodologies for or-ganizing such a huge information source. Scalability issues come into play as well as the quality of automatic organization and categorization.

Data clustering partitions a set of unlabeled objects into disjoint/joint groups of clusters. In a good cluster, all the objects within a cluster are very similar while the objects in other clusters are very different. When the data processed is a set of documents, it is called document clustering. Document clustering is very important and useful in the information retrieval area. Document clustering can be applied to a document database so that similar documents are related in the same cluster. During the retrieval process, documents belonging to the same cluster as the retrieved documents can also be returned to the user. This could improve the recall of an information retrieval system. Document clustering can also be applied to the retrieved documents to facilitate finding the useful documents for the user. Generally, the feedback of an information retrieval system is a ranked list ordered by their estimated relevance to the query. When the volume of an information database is small and the query formulated by the user is well defined, this ranked list approach is efficient. But for a tremendous information source, such as the World Wide Web, and poor query conditions (just one or two key words), it is difficult for the retrieval system to identify the interesting items for the user. Sometimes most of the retrieved documents are of no interest to the users. Applying documenting clustering to the retrieved documents could make it easier for the users to browse their results and locate what they want quickly.

Previous methods of clustering mainly uses matching key words of text, However it does not capture the meaning behind the words which is bad side of traditional method to mine the text. In the Semantic document clustering the have option to parse the web documents into two ways, first is syntactically and second is semantically. Syntactical parsing can ignore the less important data from documents so that there have a chance proper data to pass into next step. Then in next step i.e. Semantic parsing can apply on the parsed syntactic data which give can cluster the documents properly and give the needed response to user at the time of data mining which is not accurately in traditional methods.

5. Algorithms for Document Clustering

Document clustering aims to segregate documents into meaningful clusters that reflect the content of each document. For example, in the news wire, manually assigning one or more categories for each document requires exhaustive human labour, especially with the huge amount of text uploaded online daily. Thus, efficient clustering is essential. Another problem associated with document clustering is the huge number of terms. In a matrix representation, each term will be a feature and each document is an instance. In typical cases, the number of features will be close to the number of words in the dictionary. This imposes a great challenge for clustering methods where the efficiency will be greatly degraded. However, a huge number of these words are either stop words, irrelevant to the topic, or redundant. Thus, removing these unnecessary words may help significantly reduce dimensionality.

Feature selection not only reduces computational time but also improves clustering results and provides better data interpretability [88]. In document clustering, the set of selected words that are related to a particular cluster will be more informative than the whole set of words in the documents with respect to that cluster. Different feature selection methods have been used in document clustering recently, for example, term frequency, pruning infrequent terms, pruning highly frequent terms, and entropy-based weighting. Some of these methods and others will be explained in the following subsections.

5.1 Term Frequency

Term Frequency is one of the earliest and most simple yet effective term methods. It is dated back to 1957 in [56]. Thus, it is, indeed, a conventional term selection method. In a text corpus, the documents that belong to the same topic more likely will use similar words. Therefore, these frequent terms will be a good indicator for a certain topic. It can be write that a very frequent term that is normally distributed across different topics is not informative; hence, such term would be unselected. It has to tell this technique pruning highly frequent terms. Similarly, very rare terms should be pruned as well and that is called pruning infrequent terms. Stop words most likely will be pruned due to their high frequency. Furthermore, words such as abecedarian will be ignored since they will not be very frequent.

TF for term f_i with respect to the whole corpus is given by;

$$TF(f_i) = \sum_{j \in Df_i} tf_{ij}$$

5.2 Document Frequency

TF is an effective term selection method. However, it is not effective in terms of term weighting, where all selected terms will be assigned the same weight. Also, there have no chance to link TF value to any document. In other words, it cannot distinguish between frequent words that appear in a small set of documents, which could have discriminative power for this set of documents, and frequent words that appear in all or most of the documents in the corpus. In order to scale the term's weight instead, the inverse document frequency (IDF). IDF measures whether the term is frequent or rare across all documents:

$$idf(f_i) = \log \frac{|D|}{|D_{f_i}|}$$

Where $|D|$the total is number of documents (i.e., sample size) and $|D_{fi}|$ is the number of documents that contain the term f_i. The value of IDF will be high for rare terms and low for highly frequent ones.

5.3 Term Frequency-Inverse Document Frequency

It's time now to combine the above mentioned measures (i.e., TF and IDF) to produce weight for each term f_i in each document d_j. This measure is called TF-IDF. It is given by;

$$tf - idf(f_i, d_i) = tf_{ij} * idf(f_i)$$

$tf - idf$ Assigns greater values to terms that occur frequently in a small set of documents, thus having more discriminative power. This value gets lower when the term occurs in more documents, while the lowest value is given to terms that occur in all documents. In document clustering, terms that have higher $tf - idf$ have a higher ability for better clustering.

5.4 Chi Square Statistic

Chi square (x^2) statistic has been widely used in supervised feature selection [56]. It measures the statistical dependency between the feature and the class. x^2 With r different values and C classes is defined as;

$$x^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}}$$

Where n_{ij} is the number of samples (i.e., documents) with i^{th} feature value in the j^{th} class and $\mu_{ij} = \frac{n_i * n_j}{n}$ and n is the total number of documents. This equation can be interpreted using the probability as;

$$x^2(f, c) = \frac{p(f, c)p(\neg f, \neg c) - p(f, \neg c)p(\neg f, c)}{p(f)p(c)}$$

Where $p(f, c)$ is the probability of class C that contains the term f ,and $p(\neg f, \neg c)$ is the probability of not being in class c and not containing term f and so on. Thus, $\chi 2$ cannot be directly applied in an unsupervised learning such as clustering due to the absence of class label. Y. Li et al in [57] propose a variation of $\chi 2$ called $r\chi 2$ that overcomes some drawbacks of the original $\chi 2$ and is embedded in an Expectation-Maximization (EM) algorithm to be used for text clustering problems. [58] Found out that $\chi 2$ cannot determine whether the dependency between the feature and the class is negative or positive, which leads to ignoring relevant features and selecting irrelevant features sometimes. Therefore, they proposed a relevance measure (R) that can be used in the original $\chi 2$ to overcome this limitation. This new measure R follows.

$$R(f, c) = \frac{p(f, c)p(\neg f, \neg c) - p(f, \neg c)p(\neg f, c)}{p(f)p(c)}$$

R In Equation will be equal to 1 if there is no such dependency between the class and the feature, greater than 2 if there is a positive dependency and less than 1 if the dependency is negative.

From Equations, Hoffman et al. [59] proposed a new variation of $\chi 2$ that is able to distinguish positive and negative relevance:

$$r\chi 2(f) = \sum_{j=1}^{C} p(R(f, cj))\chi 2(f, cj)$$

Where $p(R(f, cj))$ is given by $p(R(f, cj)) = \frac{R(f, cj)}{\sum_{j=1}^{c} R(f, cj)}$

the larger the value of $r\chi 2$ is, the more relevant the feature f will be.

As mentioned earlier, there have not any chance to apply a supervised feature selection in an unsupervised learning directly. Therefore, [60] embedded their proposed method given in Equation in a clustering algorithm using an EM approach. They used k-means as the clustering algorithm and $r\chi 2$ as the feature selection method.

5.5 Frequent Term-Based Text Clustering

Frequent Term-Based Text Clustering (FTC), proposed in [61], provides a natural way to reduce dimensionality in text clustering. It follows the notion of a frequent item set that forms the basis of association rule mining. In FTC, the set of documents that contains the same frequent term set will be a candidate cluster. Therefore, clusters may overlap since the document may contain different item sets. This kind of clustering can be either flat (FTC) or hierarchical (HFTC) clustering since there have different cardinalities of item sets.

First, a dataset D, predetermined minimum support minsup value, and an algorithm that finds frequent item set should be available. The algorithm starts by finding the frequent item set with minimum support minsup. Then, it runs until the number of documents contributing in the selected term set |cov(STS)| is equivalent to the number of documents in D. In each iteration, the algorithm calculates the entropy overlap EO for each set in the remaining term set RTS, where EO is given by;

$$EO_i = \sum_{D_j \in C_i} - \frac{1}{F_j} . \ln(\frac{1}{F_j})$$

Where D_j is the j[th] document, C_i is the i^th cluster and F_j is the number of all frequent term sets supported by document J, with the less overlap assumed to be the better. $E0$ Equals O if all the documents in C_i support only on frequent item set (i.e., F_j = 1). This value increases with the increase of F_j. This method of overlap evaluation was found to produce better clustering quality than the standard one [62,63]. The best candidate set Best Set will be the set with a minimum amount of overlap. Best Set will be selected and added to the STS and excluded from RT S. In addition, the set of documents that supports the Best Set is removed from the dataset since they have been already clustered, which leads to dramatically reducing the number of documents. They are also removed from the documents' list of RTS which leads to reducing the number of remaining term set.

5.6 Frequent Term Sequence

Similar to FTC, a clustering based on frequent term sequence (FTS) was proposed in [63, 64]. Unlike FTC, the sequence of the terms matters in FTS. This means that the order of terms in the document is important. The frequent terms sequence, denoted as f, is a set that contains the frequent terms $< f_1, f_2, \dots, f_k >$. The sequence here means that f_2 must be after f_1, but it is not necessary to be immediately after it. There could be other nonfrequent terms between them. This is true for any f_k and $f_k - 1$ terms. This definition of frequent terms sequence is more adaptable to the variation of human languages [65].

Similar to FTC, FTS starts by finding frequent term sets using an association rule mining algorithm. This frequent term set guarantees to contain the frequent term sequence but not vice versa. Hence, it's not mandatory to search the whole term space for the frequent term sequence. It can be search in only the frequent term set space, which is a dramatic dimension reduction. After that, FTS builds a generalized suffix tree (GST), which is a well-known data structure for sequence pattern matching, using the documents after removing the non-frequent terms. From the suffix nodes in GST, Cluster's obtain the cluster candidates. These cluster candidates may contain subtopics that may be eligible to be merged together to create more general topics. Therefore, a merging step takes place.

The authors of [66] chose to merge cluster candidates into more general topic clusters using k–mismatch instead of the similarity. An example of using the k–mismatch concept is when it have $FS_i = \{feature, selection, clustering\}$ and $FS_j = \{feature, selection, classification\}$, where they have one mismatch. Therefore, it can merge these two clusters if the tolerance parameter $k \geq 1$.

In [67], FTS adopted Landau–Vishkin (LV) to test three types of mismatches: insertion, deletion, substitution. Insertion means that all it needs to insert is k, or fewer, terms into the FS_j in order to match FS_i. Deletion, in contrast, means it should to delete. While substitution means there have some need to substitute terms from FS_j with terms from FS_i.

These merged clusters are prone to overlap. Accordingly, more merging will be performed after measuring the amount of overlap using the ***Jaccard Index***:

$$J(C_i, C_j) = \frac{|C_i \cup C_j|}{|C_i \cap C_j|}$$

6. Conceptual Clustering

When no classification information is known about the data, a clustering algorithm is usually used to cluster the data into groups such that the similarity within each group is larger than that among groups. This is known as learning from observations, as opposed to the classification task which is considered as learning from examples.

A different approach is conceptual clustering. These methods are incremental and build a hierarchy of probabilistic concepts. COBWEB and its successor CLASSIT are the most notable among them. Unlike traditional hierarchical methods (that use similarity measures) they use Category Utility as the cluster quality measure.

Conceptual clustering is based on numerical taxonomy [68] and was originally introduced [69]. Gennari et al. [70] described the problem of conceptual clustering in the following way:

- Given: a sequential presentation of instances and their associated descriptions;
- Find: clustering's that group those instances in categories;
- Find: an intentional definition for each category that summarizes its instances;
- Find: a hierarchical organization for those categories.

As it is stated above, conceptual clustering organizes instances (tuples) into categories. This makes conceptual clustering suitable for categorical data that cannot be ordered and can only is put into categories. Despite differences in representation [71] and quality judgments, all conceptual clustering systems evaluate class quality by looking to a summary or concept description of the class.

There are two problems that must be addressed by a conceptual clustering system:

- The clustering problem involves determining useful subsets of an object set. This consists of identifying a set of object classes, each defined as an extensional set of objects.
- The characterization problem involves determining useful concepts for each (extensionally defined) object class. This is simply the problem of learning from examples.

Fisher and Langley [72] [73] adapt the view of learning as search to fit conceptual clustering. Clustering and characterization dictate a two-tiered search, a search through a space of object clusters and a subordinate search through a space of concepts. 1 In the case of hierarchical techniques this becomes a three-tiered search, with a top-level search through a space of hierarchies.

A successful conceptual clustering algorithm that has been the basis for many other algorithms, for example LABYRINTH [74], ITERATE [75], and COBWEB [76].

6.1 Conceptual Clustering Algorithms

Conceptual clustering is obviously closely related to data clustering; however, in conceptual clustering it is not only the inherent structure of the data that drives cluster formation, but also the Description language which is available to the learner. Thus, a statistically strong grouping in the data may fail to be extracted by the learner if the prevailing concept description language is incapable of describing that particular regularity.

A fair number of algorithms have been proposed for conceptual clustering. Some examples are given below:

CLUSTER/2: Early work on conceptual clustering was done by Mechalski and Stepp [77] who proposed the conceptual clustering algorithm known as CLUSTER/2. The choice for conceptual clustering arises from the interesting property that conceptual clustering is mostly used for nominal-valued data. An extension exists for conceptual clustering that can deal with numeric data [78], but for the purpose of this paper there have only need to be concerned with nominal-valued data as the data set it was dealing with is inherently nominal and symbolic-valued. However, the data set it was dealing with contains large number of attributes, and their values are non-fixed nominal values. Pre-processing of the data is then a very important step to make the data usable.

Conceptual clustering builds a structure out of the data incrementally by trying to subdivide a group of observations into subclasses. The result is a hierarchical structure known as the concept hierarchy. Each node in the hierarchy subsumes all the nodes underneath it, with the whole data set at the root of the hierarchy tree.

LABYRINTH: [79] to incorporate a structured object into a node, Labyrinth performs an additional search to determine the best characterization for the object. Since the values labyrinth uses for structured objects are stored concepts that have been returned by earlier classification, they are hierarchically related to each other. Labyrinth uses an attribute generalization operator analogous to the climbing tree operator to take advantages of these hierarchical relationships and to search for more predictive characterizations of structured method.

Traditionally, Concept formation system have started tabula rasa, i.e., without exploiting knowledge of the domain. However, the incremental nature of system labyrinth and Cobweb means that they can revise and existing memory structure. One can simply hand-encode an initial memory and start operation from there. The initial memory thus serves to prime the learning algorithm, and the normal concept learning operators revise and extend the initial theory.

Since cobweb store information characterizing each class individually, rather than as organization of several component concepts, expressing knowledge of subsets of attributes in not straight forward. In contrast, labyrinth can be primed with class that characterize arbitrary subsets of attributes, provided instances and decomposed in the same way. Thus, labyrinth's use of components in classification enables it to take advantage of a form of background knowledge that is common in many domains: information about correlated sets of attributes.

ITERATE: Research conducted by our group has led to the development of ITERATE, a conceptual clustering algorithm that works with combinations of numeric and non-numeric data. The primary motivations for developing ITERATE were to extend previous conceptual clustering algorithms (e.g., COBWEB) to generate stable and maximally distinct partitions [80] and to produce an efficient algorithm for an interactive data analysis tool. Like other conceptual clustering algorithms, ITERATE builds a concept tree from domain objects or instances represented as a vector of attributes value pairs but tries to mitigate the effect of incremental control structures. The algorithm exploits information on the entire object set in creating the object hierarchy. More specifically, it adopts an ordering operator that preorders the object sequence to exploit the biases of the criterion function, in forming maximally distinct classes in the initial classification tree [88]. The tree is generated in a breadth-first manner; therefore, class probabilities at a parent node are allowed to stabilize before child nodes are created.

COBWEB: Cobweb is a conceptual clustering algorithm developed by Fisher [81] for the analysis of categorical data that cannot be ordered. The algorithm builds a hierarchy of clusters following the divisive approach to clustering. The goal of Cobweb, like all conceptual clustering algorithms, is to build a model that can be used for future predictions [82].

Cobweb is a relatively old algorithm but since it was introduced its relevance to solving data mining problems has remained important. Biswas et al. [83] use Cobweb for predicting missing values. Perkowitz & Etzioni [84] discuss the suitability of Cobweb for data mining on the web, [85] and Paliouras et al. [86] use Cobweb on the web, while Li et al. [87] combine Cobweb with k-means [88] to present an algorithm for large scale clustering. The algorithm is, also, part of a number of popular general purpose data mining tools. Two of these data mining tools are (i) Weka [89] which provides an implementation of Cobweb that is applicable to categorical and numeric data, and (ii) OIDM [90], which provides an implementation of Cobweb based on the original Fisher's paper.

6.2 COBWEB Algorithm

The Cobweb algorithm is an incremental clustering algorithm that clusters one tuple at a time in a top down manner. It starts clustering a tuple by inserting it into the root cluster of the tree (figure 6 is an example of a Cobweb tree). Inserting a new tuple in a cluster involves updating the probabilities the cluster covers.

The algorithm uses four operators to evaluate and improve the quality of the tree. The quality measure in Cobweb is category utility. The four operators are: (i) incorporate, (ii) disjunct, (iii) split, and (iv) merge. The incorporate and disjunct operators are used to build the tree while the merge and split operators are used to correct any data ordering bias in the clusters by reordering the hierarchy.

and merges - and identifies which is the best operator to implement by measuring the category utility of the clustering produced by each operator. Category utility favours the operator that when implemented produces a clustering that maximises the potential for inferring information [92] [93].

If the best operator is incorporate, the algorithm inserts the new tuple in the best cluster and proceeds to the next level.

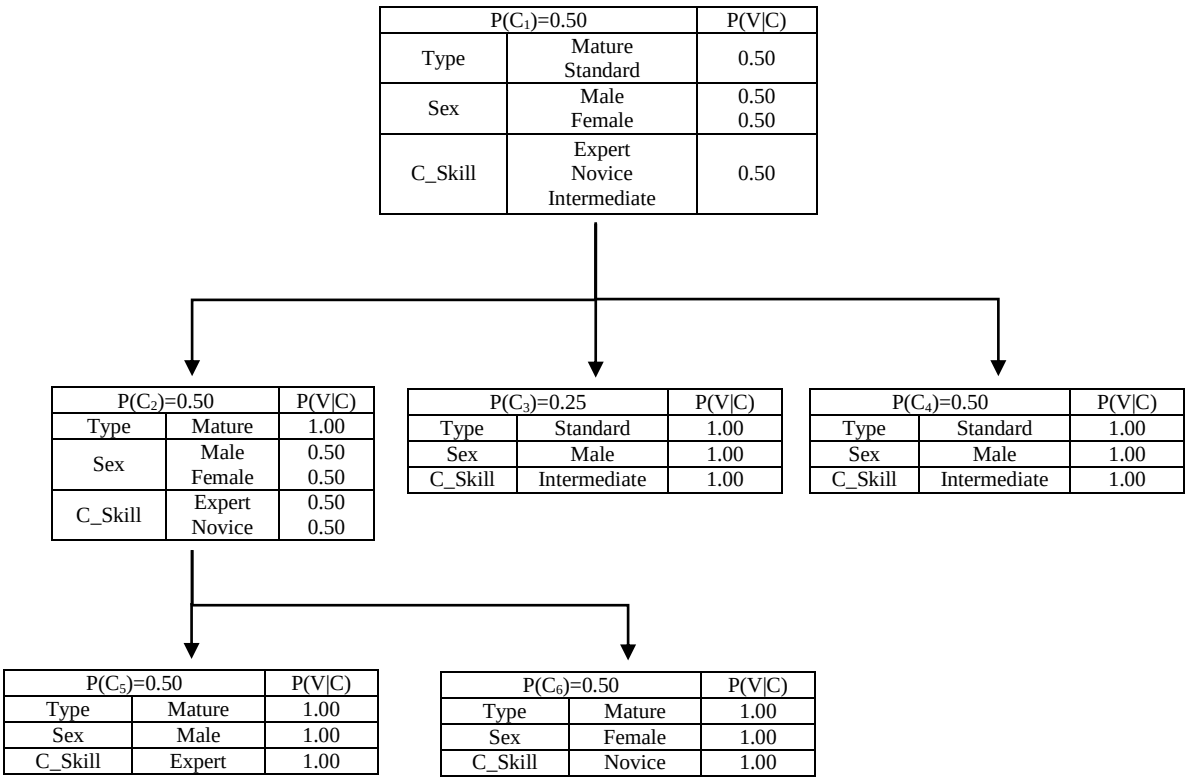


Figure 6: The COBWEB Tree

- Incorporate: Cobweb tries a new tuple in every cluster of the assessed level to identify the best cluster to incorporate the new tuple. It also records the second best cluster as it is needed by other operators.
- Disjunct: Cobweb tries a new tuple in a new cluster that covers only the tuple.
- Split: Cobweb replaces the best cluster, identified by the incorporate operator, with its children and tries the new tuple in every child of the best cluster.
- Merge: Cobweb merges the best and second best clusters, identified by the incorporate operator, and tries the new tuple in the merged cluster.

According to Fisher et al. [91], the incremental property can have an impact on the quality of the clusters as incremental algorithms are sensitive to the order of the data [98]. With the merge and split operators the algorithm corrects the ordering effect by restructuring the tree.

As it descends down the tree, at every level of the tree, Cobweb tries all four operators - incorporate, disjunct, split

If the best operator is disjunct, the algorithm creates a new cluster in the tree. If the best operator is split, the algorithm re-arranges the tree by replacing the best cluster with its children and moves to the next level. If the best operator is merge, the algorithm merges the best and second best cluster (best and second best cluster are indicated by the incorporate operator) and moves to the next level.

```
Function Cobweb (tuple, root)
    Incorporate tuple into the root;
    If root is a leaf node Then
        Expand leaf node;
            Return expanded leaf node with
        the tuple;
    Else
            Get the children of the root;
```

Evaluate operators and select the best:

a) Try incorporate the tuple in every child;

b) Try creating a new cluster with the tuple;

c) Try merging the two best clusters;

d) Try splitting the best cluster into its children;

If (a) or (c) or (d) is best operator Then

call Cobweb (tuple, best cluster);

Cobweb has an additional operator used to predict missing values, the predict operator. The predict operator classifies a tuple down the tree using the incorporate operator but it does not add the tuple to the clusters in the tree.

References

[1] A. Kaur Toor and A. Singh, An Advanced Clustering Algorithm (ACA) for Clustering Large Data Set to Achieve High Dimensionality:. Computer Science Systems Biology, Toor and Singh, J Comput Sci Syst Biol 2014, 7:4. URL: http://dx.doi.org/10.4172/jcsb.1000146

[2] Anil K. Jain, Michigan State University & M.N. MURTY Indian Institute of Science & FLYNN the Ohio State University: Data Clustering: A Review; ACM Computing Surveys, Vol. 31, No. 3. 264-323

[3] Liu , G. Introduction to combinatorial mathematics . New York, NY : McGraw Hill .

[4] S. Lloyd. Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2):129–137.

[5] J. MacQueen. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281–297, Berkeley, CA, USA.

[6] C. D. Manning, P. Raghavan, and H. Schutze. Introduction to Information Retrieval, volume 1. Cambridge University Press, Cambridge, 2008.

[7] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):881–892, 2002.

[8] C. Elkan. Using the triangle inequality to accelerate k-means. In Proceedings of International Conference on Machine Learning (ICML), pages 147–153, 2003.

[9] Everitt , B. , Landau , S. , and Leese , M. Cluster analysis, 4 th edition . London : Arnold .

[10] Anil K. Jain and Richard C. Dubes, Michigan State University; Algorithms for Clustering Data: Prentice Hall, Englewood Cliffs, New Jersey 07632. ISBN: 0-13-0222278-X

[11] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation, Journal of Machine Learning Research, 3:993–1022, 2003.

[12] Hansen , P. and Jaumard , B. Cluster analysis and mathematical programming. Mathematical Programming , 79 : 191 – 215

[13] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS-clustering categorical data using summaries. ACM KDD Conference, 1999.

[14] Theodoridis , S . and Koutroumbas, K. (2006). Pattern recognition, 3r d . San Diego, CA : Academic Press .

[15] S. Fahad & M. Alam, "A modified K-means algorithm for big data clustering", International Journal of Science, Engineering and Computer Technology, vol. 6, no. 4, 2016.

[16] L. L. McQuitty. Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. Educational and Psychological Measurement 17(2):207–229, 1957.

[17] P. H. A. Sneath and R. R. Sokal. Numerical Taxonomy: the Principles and Practice of Numerical Classification. Freeman.

[18] B. King. Step-wise clustering procedures. Journal of the American Statistical Association, 62(317):86–101.

[19] J. C. Gower and G. J. S. Ross. Minimum spanning trees and single linkage cluster analysis. Journal of the Royal Statistical Society. Series C (Applied Statistics), 18(1):54–64.

[20] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2(2):139–172.

[21] Oikonomakou, N. and M. Vazirgiannis, A Review of Web Document Clustering Approaches, in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Editors. 2010, Springer US. p. 931-948.

[22] Sathiyakumari, K., et al., A Survey on Various Approaches in Document Clustering. Int. J. Comp. Tech. Appl., IJCTA, 2011. 2(5): p. 1534-1539.

[23] Wael M.S. Yafooz, Abidin, S. Z., Omar, N., & Halim, R. A. (2014). Shared-Table for Textual Data Clustering in Distributed Relational Databases. In Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013) (pp. 49-57). Springer, Singapore.

[24] Pantel, P. and D. Lin, Document clustering with committees, in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval2002, ACM. p. 199-206.

[25] Han, J., M. Kamber, and J. Pei, Data mining: concepts and techniques. 2006: Morgan kaufmann.

[26] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. IEEE Transactions on Knowledge and Data Engineering, 16(11):1370–1386, 2004.

[27] S. Schaeffer. Graph clustering. Computer Science Review, 1(1):27–64, 2007.

[28] L. R. Rabiner. A tutorial on hiddenMarkov models and selected applications in speech recognition, Proceedings of the IEEE, 77(2):257–285.

[29] Hung Chim, F.X.D.F., Efficient Phrase-Based Document Similarity for Clustering. IEEE Trans. Knowl. Data Eng. IEEE Transactions on Knowledge and Data Engineering. 20(9): p. 1217-1229.

[30] Li, Y., S.M. Chung, and J.D. Holt, Text document clustering based on frequent word meaning sequences. Data & Knowledge Engineering, 2008. 64(1): p. 381-404.

[31] Fung, B.C., K. Wang, and M. Ester, Hierarchical document clustering using frequent itemsets, in Proceedings of SIAM international conference on data mining2003. p. 59-70.

[32] Li, Y., C. Luo, and S.M. Chung, Text clustering with feature selection by using statistical data. Knowledge and Data Engineering, IEEE Transactions on, 2008. 20(5): p. 641-652.

[33] Wael M.S. Yafooz, Abidin, S. Z., Omar, N., & Idrus, Z. (2013, December). Managing unstructured data in relational databases. In Systems, Process & Control (ICSPC), 2013 IEEE Conference on (pp. 198-203). IEEE.

[34] Wael M.S. Yafooz, Abidin, S. Z., Omar, N., & Halim, R. A. (2014). Model for automatic textual data clustering in relational databases schema. In Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013) (pp. 31-40). Springer, Singapore.

[35] D. Gusfield. Algorithms for Strings, Trees and Sequences, Cambridge University Press, 1997.

[36] Wei Xu, Xin Liu, Yihong Gong. Document Clustering Based On Nonnegative Matrix Factorization. In ACM. SIGIR, Toronto, Canada, 2003.

[37] McKeown, K.R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J.L., Nenkova, A., Sable, C., Schiffman, B., Sigelman, S.: Tracking and summarizing news on a daily basis with Columbia's

Newsblaster. Proceedings of the second international conference on Human Language Technology Research, pp. 280-285. Morgan Kaufmann Publishers Inc., San Diego, California (2002)
[38] Liu, X., Croft, W.B.: Cluster-based retrieval using language models. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 186-193. ACM, Sheffield, United Kingdom (2004)
[39] Hartigan, J. Clustering algorithms. New York, NY: John Wiley & Sons.
[40] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs, Bell System Tech. Journal, 49:291–307.
[40] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: A partition-and-group framework. SIGMOD Conference, 593–604, 2007.
[41] S. Fortunato. Community detection in graphs, Physics Reports, 486(3–5):75–174, February 2010.
[42] Kaufman, L., and Rousseeuw, P. J. Finding Groups in Data. An Introduction to Cluster Analysis. John Wiley & Sons.
[42] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences, 95(25):14863–14868, http://rana.lbl.gov/EisenSoftware.htm
[43] Zahn, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transaction on Computers C-20, 1, 68–86.
[44] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing, 48(1):96–129.
[45] Michael Steinbach, George Karypis, Vipin Kumar. A Comparison of Document Clustering Techniques. KDD. Workshop on Text Mining, 2000. http://www-users.cs.umn.edu/karypis/publications/Papers/PDF/doccluster.pdf
[46] G. Qi, C. Aggarwal, and T. Huang. Community detection with edge content in social media networks, ICDE Coference, 2013.
[47] G. Qi, C. Aggarwal, and T. Huang. Online community detection in social sensing. WSDM Conference, 2013.
[48] Y. Sun, C. Aggarwal, and J. Han. Relation-strength aware clustering of heterogeneous information networks with incomplete attributes, Journal of Proceedings of the VLDB Endowment, 5(5):394–405, 2012.
[49] Kenneth Lolk Vester, Moses Claus Martiny. Information retrieval In Document Spaces Using Clustering. in Informatics and Mathematical Modelling, Technical University of Denmark, DTU. 2005
[50] Inderjit S. Dhillon, University of Texas, Austin Information Theoretic Clustering, Co-clustering and Matrix Approximations. MA Workshop on Data Analysis and Optimization. 2003.
[51] M.E.S. Mendes Rodrigues and L. Sacks, ‚A Scalable Hie-rarchical Fuzzy Clustering Algorithm for Text Mining', Department of Electronic and Electrical Engineering University College London Torrington Place, London, WC1E 7JE, United Kingdom, 2004.
[52] M. Mugunthadevi,M. Punitha, and M. Punithavalli. Survey on feature selection in document clustering. International Journal, 3, 2011.
[53] H.P. Luhn. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of Research and Development, 1(4):309–317, 1957.
[54] I.H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.
[55] Y. Li, C. Luo, and S.M. Chung. Text clustering with feature selection by using statistical data. IEEE Transactions on Knowledge and Data Engineering, 20(5):641–652, 2008.
[56] S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1(1):24–45, 2004.
[57] J. Yang and W. Wang. CLUSEQ: Efficient and effective sequence clustering, ICDE Conference, 2003.
[58]Langley, P., Order effects in incremental learning, in P. Reimann & H. Spada, eds, ‘Learning in Humans and Machines: Towards an Interdisciplinary Learning Science’, Pergamon, pp. 154–167.
[59] F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 436–442. ACM, 2002.
[60] George A. Miller, ‘Nouns in WordNet: A Lexical Inheritance System ’
[61] T. Hofmann. Probabilistic latent semantic indexing. ACM SIGIR Conference, 1999.
[62] Wael.M.S. Yafooz, Abidin, S. Z., Omar, N., & Halim, R. A. (2013, August). Dynamic semantic textual document clustering using frequent terms and named entity. In System Engineering and Technology (ICSET), 2013 IEEE 3rd International Conference on (pp. 336-340). IEEE.
[63] Wael M.S. Yafooz, Abidin, S. Z., & Omar, N. (2011, November). Towards automatic column-based data object clustering for multilingual databases. In Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on (pp. 415-420). IEEE.
[64] D. R. Karger. Random sampling in cut, flow, and network design problems, STOC, pp. 648–657.
[65] T. Liao. Clustering of time series data—A survey. Pattern Recognition, 38(11):1857–1874, 2005.
[66] Forgy, E. Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. Biometrics 21, 768–780.
[67] G. Das and H. Mannila. Context-based similarity measures for categorical databases. PKDD Conference, pages 201–210, 2000.
[68] Fisher, D. H. & Langley, P., Conceptual clustering and its relation to numerical taxonomy, in W. A. Gale, ed., ‘Artificial Intelligence and Statistics’, Boston, MA: Addison-Wesley, pp. 77–116.
[69] Michalski, R. S. & Stepp, R. E., Learning from observation: conceptual clustering, in R. S. Michalski, J. G. Carbonell & T. M. Mitchell, eds, ‘Machine Learning: An Artificial Intelligence Approach’, San Mateo, CA: Morgan Kaufmann, pp. 331–364.
[70] Fisher, D. H., & Langley, Approaches to conceptual clustering. Proceedings of the Ninth International Conference on Artificial Intelligence (pp. 691 697). Los Angeles, CA: Morgan Kaufmann.
[71] Y. Zhou, H. Cheng, and J. X. Yu, Graph clustering based on structural/attribute similarities, Proc. VLDB Endow., 2(1):718–729, 2009.
[72] Thompson, K. & Langley, Concept formation in structured domains, in D. H. Fisher, M. J. Pazzani & P. Langley, eds, ‘Concept Formation: Knowledge and Experience in Unsupervised Learning’, Morgan Kaufmann, pp. 127–161.
[73] Biswas, G.,Weinberg, J., Yang, Q. & Koller, Conceptual clustering and exploratory data analysis, in L. A. Birnbaum & G. C. Collins, eds, ‘Proceedings of the Eighth International Workshop on Machine Learning’, San Francisco, CA: Morgan Kaufmann, pp. 591–595.
[74] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering, ACM KDD Conference, 2003.
[75] B.-K. Yi, N. D. Sidiropoulos, T. Johnson, H. Jagadish, C. Faloutsos, and A. Biliris. Online data mining for co-evolving time sequences. ICDE Conference, 2000.
[76] C. Li and G. Biswas, “Conceptual clustering with numeric-and-nominal mixed data – A new similarity based system,” IEEE Trans. Knowl. Data Engineering.
[77] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. ACM SIGIR Conference, pages 318–329.
[78] G. Biswas, et al. ITERATE: A conceptual clustering algorithm that produces stable clusters," in review, IEEE Trans. on Pattern Analysis and Machine Intelligence.
[79] I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning, ACM KDD Conference, 2001.
[80] C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. SDM Conference, 2005.
[81] Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery, 10(2):141–168, 2005.
[82] Y. Zhu and D. Shasha. StatStream: Statistical monitoring of thousands of data streams in real time. VLDB Conference, pages 358–369, 2002.
[83] Perkowitz, M. & Etzioni, O. (2000), ‘Towards adaptive web sites: Conceptual framework and case study’, Artificial Intelligence 118(1), 245–275.
[84] Hurst, N., Marriott, K. & Moulder, P. (2003), Cobweb: a constraint-based web browser, in M. J. Oudshoorn, ed., ‘Twenty-sixth Australian computer science conference (ACSC 2003)’, Vol. 16, Adelaide, South Australia, Australian Computer Society, pp. 247–254.
[85] Paliouras, G., Papatheodorou, C., Karkaletsis, V., Spyropoulos, C. & P.Tzitziras, From web usage statistics to web usage analysis, in ‘Proceedings of the IEEE International Conference on Systems, Man and Cybernetics’, Vol. 2, Tokyo, Japan, pp. 159–164.

[86] Li, T. (2005), A general model for clustering binary data, in R. Grossman, R. J. Bayardo & K. P. Bennett, eds, 'Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Chicago, Illinois, USA, ACM, pp. 188–197.
[87] P. Andritsos et al. LIMBO: Scalable clustering of categorical data. EDBT Conference, 2004.
[88] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2000.
[89] Chen, Q., Wu, X. & Zhu, X. Oidm: Online interactive data mining, in R. Orchard, C. Yang & M. Ali, eds, 'Proceedings of the 17th International Conference on Innovations in Applied Artificial Intelligence', Ottawa, Canada, Springer, pp. 66–76.
[90] C. Aggarwal and C. Zhai. A survey of text clustering algorithms, Mining Text Data, Springer, 2012.
[91] Anderberg ,M. cluster analysis for applications.New York, NY : Academic Press .
[92] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. Information Systems, 25(5):345–366, 2000.
[93] D. Gibson, J. Kleiberg, and P. Raghavan. Clustering categorical data: An approach based on Dynamical Systems. VLDB Conference, 1998.