NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
FACULTY OF INFORMATICS AND TELECOMMUNICATIONS

BACHELOR THESIS

Clustering and Classification in High Dimensional Sparse Data

Ioannis M. Borektsioglou
Konstantinos N. Patsourakos

Supervisor: **Ioannis Z. Emiris,** Professor

ATHENS

MARCH 2016

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ομαδοποίηση και Κατηγοριοποίηση σε Πολυδιάστατα Αραιά Δεδομένα

Ιωάννης Μ. Μπορεκτσίογλου
Κωνσταντίνος Ν. Πατσουράκος

Επιβλέπων: **Ιωάννης Ζ. Εμίρης** Καθηγητής

ΑΘΗΝΑ
ΜΑΡΤΙΟΣ 2016

BACHELOR THESIS

Clustering and Classification in High Dimensional Sparse Data

Ioannis M. Borektsioglou

A.M.: 1115201000111

Konstantinos N. Patsourakos
A.M.: 11152009000129

SUPERVISOR : Ioannis Z. Emiris, Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ομαδοποίηση και Κατηγοριοποίηση σε Πολυδιάστατα Αραιά Δεδομένα

Ιωάννης Μ. Μπορεκτσίογλου
Α.Μ.: 1115201000111

Κωνσταντίνος Ν. Πατσουράκος
Α.Μ.: 11152009000129

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Ιωάννης Ζ. Εμίρης** Καθηγητής

ABSTRACT

The main goal of this dissertation can be accumulated as the effort of classification of real high-dimensional sparse data in the area of homeopathy. In order to achieve these goals there have been gathered various methodologies from data mining area. Some suitable clustering algorithms were implemented until there was a good and useful result according to field experts. The biggest challenge was the absence of ground truth that would help lead the attempts to better understand the problem. For that reason, we had to rely on internal evaluation and experiment with different scoring functions.

Specifically in order to attain the above mentioned goals, a partitional clustering algorithm was implemented. We started with k-medoids approach with k-medoids++ initialization, PAM assignment (Leonard Kaufman and Peter J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis)* and CLARANS update (Raymond T. Ng and Jiawei Han, "Efficient and Effective Clustering Methods for Spatial Data Mining"). Because of the hierarchical structure of data the above methods did not give useful results, according to internal evaluation, so a hierarchical algorithm known as connected compenents was implemented.

Lastly, in order to make some conclusions about the words that appeared in data, we implemented hitting set algorithm. It was important to find the words that appeared the most independently of the others and we saw the problem as the known set covering problem.

SUBJECT AREA: Data mining.

KEY WORDS: clustering, classification, big data, sparse, high dimensional.

ΠΕΡΙΛΗΨΗ

Ο βασικός στόχος της συγκεκριμένης πτυχιακής είναι η κατηγοριοποίηση πραγματικών πολυδιάστατων δεδομένων στο τομέα της ομοιοπαθητηκής. Για να το πετύχουμε αυτό συγκεντρώσαμε αρκετές μεθοδολογίες από το χώρο της εξόρυξης δεδομένων. Υλοποιήθηκαν μερικοί ιδανικοί αλγόριθμοι ομαδοποιήσης μέχρι να υπάρξει ένα καλό και χρήσιμο αποτέλεσμα σύμφωνα με τους ειδικούς του τομέα.

Πιο συγκεκριμένα, για να πετύχουμε αυτό το αποτέλεσμα, υλοποιήσαμε έναν διαιρετικό αλγόριθμο ομαδοποιησης. Ξεκινήσαμε με τη μέθοδο του k-medoids με αρχικοποιήση k-medoids++, ανάθεση PAM και ανανέωση CLARANS. Επειδη τα δεδομένα ακολουθούσαν μμια ιεραρχική δομή οι παραπάνω μεθόδοι δεν έδωσαν ένα χρήσιμο αποτέλεσμα σύμφωνα με τις μεθοδους αξιολόγησης που χρησιμοποιήσαμε, έτσι υλοποιήσαμε ιεραρχικούς αλγορίθμους, ενας εκ των οποίων ο αλγόριθμος Connected components.

Τελος, για να βγάλουμε κάποια συμπεράσματα για τις λέξεις που είχαμε στα δεδομένα, υλοποίησαμε τον αλγόριθμο hitting set. Ήταν σημαντικό να βούμε τις λέξεις που ήταν ανεξάρτητες από τις υπόλοιπες και για αυτό είδαμε το πρόβλημα σαν το γνωστό πρόβλημα set covering

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εξόρυξη Δεδομένων.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ομαδοποίηση, κατηγοριοποίηση, μεγάλα δεδομένα, αραιά, πολυδιάστατα.

ACKNOWLEDGEMENT

We would first like to thank our thesis supervisor Prof. Ioannis Z. Emiris. The door to his office was always open whenever we ran into a trouble spot or had a question about our research or writing. He consistently allowed this paper to be our own work, but steered us in the right direction whenever he thought we needed it.

We would also like to thank Prof. Theodoros Lilas for sharing his experience in the application part of the thesis and giving us a lot of insight in the area of homeopathy as viewed from computer science.

Finally, we would like to thank the researchers who were involved in the conduction of this thesis: Reasearcher Evangelos Anagnostopoulos and Senior Researcher Ioannis Avrithis. Without their passionate participation and input, the project could not have been successfully carried out.

CONTENTS

LIST OF EQUATIONS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In recent years, data generation and data collection has seen a dramatic rise by a variety of sources, from physics to ads. It seems straightforward that in the future the amount of data available to us will increase, not only in volume from the current sources but also from new sources.

Data mining is the procedure of naturally discovering valuable data in expansive information archives. The reason for applying data mining procedures is finding critical designs from datasets and furthermore giving capacities to anticipate the result of a future perception. For example, market basket analysis, implies that by utilizing "Association Rules" learning, the supermarket can figure out which items are as often as possible purchased together or to predict if the new client will spend more than 100 $ for shopping today at the store. With respect to the Wikipedia definition, data mining consists of six normal assignments: Anomaly Detection, Association rules learning, Classification, Clustering and Regression. In this report, we talked about for the most part on Clustering.

Clustering is the most critical unsupervised-learning problem. The primary reason for existing is discovering a structure in a collection of unlabeled data. Absolutely, the clustering includes partitioning a given dataset into a few groups of data whose individuals are comparable somehow. The ease of use of cluster analysis has been used broadly as a part of data recovery, text and web mining, pattern recognition, image segmentation and software reverse engineering.

1.1. Outline

The content of this dissertation is the clustering of high-dimensional sparse data. It consists of five chapters, which we describe below.

In the second chapter, clustering analysis and methods are presented. Particularly methods of clustering are described. It is also described a comparison between each method of cluster analysis and it is stated which is better for what purpose. Then, there is a description of the clusters that are formed from each type of algorithm (e.g. center-

based, density-based etc.) In addition, they are described algorithms that were used on our data. The biggest challenge was the absence of ground truth; for that reason, we had to rely on internal evaluation. Functions of internal evaluations are described in this chapter.

In the third chapter, we described a number of scoring functions, which were used in order to measure vectors with weighted dimensions. There is also a comparison between each scoring function.

In the fourth chapter, it is described a technique named set covering. We used it to search over phrases with large but strict vocabulary. We had to see phrases as vector and words as its dimensions.

In the final chapter, we present where we applied all these techniques. We present the medical problem that we had to deal with and the data set on which we worked.

1.2. Trends

Because it is an emerging discipline, many challenges remain in data mining. Due to the enormous volume of data acquired on an everyday basis, it becomes imperative to find an algorithm that determines which technique to select and what type of mining to do. Data sets are often inaccurate, incomplete, and have redundant or insufficient information. It would be desirable to have mining tools that can switch to multiple techniques and support multiple outcomes. Current data-mining tools operate on structured data, but most data are unstructured. For example, enormous quantities of data exist on the World Wide Web. This necessitates the development of tools to manage and mine data from the World Wide Web to extract only the useful information. There has not yet been a good tool developed to handle dynamic data, sparse data, incomplete or uncertain data, or to determine the best algorithm to use and on what data to operate

2. CLUSTER ANALYSIS

Dividing objects in meaningful groups of objects or classes (cluster) taking into account basic characteristic, play an important role in how individuals analyze and depict the world. For example, every human being can rapidly label objects in a photo, such as structures, trees, and so on.

In the field of understanding data, we can say that clusters are potential classes and cluster analysis is a technique to identify classes [1]. Before discussing about cluster analysis we need to provide a necessary description as a background for understanding the topic. First, we need to clarify the cluster analysis and the reasons behind its difficulties, and explain its relationship to other techniques of grouping data objects. In addition, we have to explain two subjects, different ways of grouping a set of objects into a set of clusters and cluster types

Cluster analysis or **Clustering** is the process of partitioning data objects into subsets, called **clusters**, based on the information from which are described. The goal is to make the data of each cluster similar to one another and dissimilar to data objects of other clusters. This separation might be useful in order to understand the data and their physical structure. It is being connected with plenty of science disciplines and has been studied in numerous of research communities, for example, machine learning, statistic, optimization and computational geometry [2]

Here are some examples:

- **Biology.** Biologists when they quite a while prior made a scientific categorization (hierarchical classification) made a type of clustering as indicated by genus, family, species etc. But also, recently they have connected clustering to examine the heap measure of hereditary data, for example, a group of genes that has similar functions.

- **Information Retrieval.** In Web Search, for example, a keyword may find a huge number of hits. Clustering helps the user to organize all this information into groups and present it to the user in an accessible way.

- **Psychology and Medicine.** Clustering techniques are utilized to break down continuous states of an ailment and distinguishing distinctive subcategories. Case in point, clustering is used to distinguish diverse sorts of depression, and cluster analysis is used to identify examples in the spread of an ailment.

- **Business.** In this field there exists a lot of data on current and potential clients. Clustering aids to gathering client activities, as previously specified in detail.

Because, clustering can organize data objects into clusters where all objects are similar to one another and dissimilar to objects in other clusters, we can treat a cluster as a class. In this sense, clustering can be called **automatic classification.** A substantial difference here is that clustering can automatically find the clusters. This is a unique advantage of cluster analysis.

It is important to notice the difference between classification and clustering. Classification is a type of **supervised learning** algorithm, because class label is given to the algorithm. [1] Clustering is a type of **unsupervised learning** algorithm, because class label is not given. In data mining, efforts have been made to find effective and efficient algorithms for data analysis, in large databases and research has focused on the *scalability* of clustering methods, the *effectiveness* of methods for clustering, *high-dimensional* techniques and much more.

In this report, we describe various methods of clustering that we have implemented and some evaluation techniques which are very useful when an algorithm is used. We apply these methods on sparse and high-dimensional[1] data.

2.1. Methods of Cluster Analysis

A common way to differentiate between clustering techniques is whether we want the result to contain nested clusters giving a hierarchical clustering or a partitional. The choice has to be taken depending on the underlying structure that the data might have. Another way is whether we want some object to be placed into two or more clusters and

[1] Vectors with over 90% zero values and over 700 total values as dimensions

with a weight. We distinguish these methods into exclusive, overlapping and fuzzy clustering. In this report we talked about mostly for partitional and hierarchical algorithms.

2.1.1. Partitional

Partitional method is the most common method of separating *N*data objects into *K* non-overlapping clusters such that each object belongs to a single cluster. Most partitional algorithms start using an initial partition, following iterative steps in order to minimize or maximize an objective function. There are two approaches, **k-means**, where the center of the cluster is the mean of its points, based on a distance function and **k-medoids**, where the center of the cluster is one of the points from the dataset.

PAM [3] is a partitional algorithm which uses the k-medoids approach. PAM selects *K* objects as medoids to represent clusters and swaps with other objects until an objective function is optimized. PAM has slow processing time, $O(K(N-K)^2)$ because it compares each medoid with all objects of the data set.

CLARA(Clustering LARge Applications) [3] applies PAM on a random, uniform sample of the data set, and finds the medoids of this sample. CLARANS (Clustering Large Applications based on RANdomizedSearch) was an improvement of CLARA [4]. This is the first method which applies clustering on large high-dimensional dataset and it has overcome most of the disadvantages of clustering techniques [5], but there are no guarantees for the quality of the results for very large data sets, because of its randomized approach.

2.1.2. Hierarchical

With this type of clustering method, we create a hierarchical structure which is represented as a *dendrogram* so it is used when we need a hierarchy or a classification. The set of objects is separated iteratively into nested groups based on the distances between each object. Every inner node that is created in *dendrogram* is a union between the groups and the leaves are the data. The advantage with this method is that

we can stop the procedure at the desired level in order to have a balanced structure. Also hierarchical algorithms do not need K as an input, which is a big advantage over partitional algorithms. On the other hand, hierarchical method has a slow processing time, $O(N^3)$ and needs too much space to store the structure. Moreover, during the connection of nodes, high-dimensional or noisy data may cause problems at the result. Finally, hierarchical clustering can be seen as a sequence of partitional clustering and a partiotional clustering can be obtained by taking any level of hierarchical clustering.

There are two approaches of hierarchical method. The first is *agglomerative*, where we start with clusters with one object and finding the most similar object we merge clusters. The second one which is the opposite procedure, is called *divisive*, where following some iterative steps we separate the initial cluster in smaller clusters with one object.

2.1.3. Exclusive

Exclusive clustering is as the name suggests and stipulates that each data object can only exist in one cluster. This method of clustering although may be problematic, it is the most efficient.

2.1.4. Overlapping

In overlapping clustering, data objects are simultaneously assigned to more than one cluster, usually adjacent. This method of clustering is used in order to avoid arbitrary assignments to clusters by placing objects to all of the equally good clusters. For example, a person at a university might be enrolled as a student and as an employee at the university. Overlapping clustering might be also used when an object is between two clusters, and we want to place it in both clusters.

2.1.5. Fuzzy

In this method of clustering we treat clusters as **fuzzy sets[2].** We assign to every object a vector with weights between 0 and 1, which sum is equal to 1. This vector shows how much an object belongs to the specific cluster (0 means that absolutely does not belong, and 1 means that absolutely belongs). Similarly, probabilistic clustering techniques compute the probability with which each points belongs to each cluster and these probabilities must sum to 1. In practice fuzzy clustering can be converted to exclusive by assigning an object to a cluster with the maximum weight.

The Fuzzy c-Means algorithm known as FCM [6] is the most known algorithm which performs fuzzy clustering. The FCM algorithms is applicable to a wide variety of statistical data analysis problems and pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\left\|x_i - c_j\right\|^2$$

Equation 1:Objective function FCM

where *m* is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster *j*, x_i is the *i*-th of d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function described above with the update of membership u_{ij} and the cluster c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\frac{1}{\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|}\right)^{\frac{2}{m-1}}}} \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m}.\ x_j}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Equation 2: Update uij and center cj FCM

[2] Mathematically fuzzy set is one whose objects belong to it with a weight between 0 and 1

This iteration will stop when $\max_{ij}\left\{\left\|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right\|\right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas *k* are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m. [7]

2.2. Types of Cluster

Cluster analysis aims to find useful groups of data objects for each application of data analysis. There are several notions for types of clusters. They are separated based on their structure and how they are separated. In order to visualize each type of cluster, we use two-dimensional points, as shown in Figure 1.

2.2.1. Well-Separated

Well-Separated cluster is a group of points where each point is closer to any other point in its cluster than any point that does not lie in the cluster. Sometimes we can specify a threshold in order to assure that all objects are sufficiently close in the group. Figure 1(a) gives an example of three well-separated clusters with two-dimensional points where every two point in different clusters have more distance than every two points in the same clusters. Well-Separated clusters can have any shape

2.2.2. Center-Based

Center-Based Cluster is a group of objects that are closer (or similar) to the center their cluster than to a center of any other cluster. For data with continuous attributes the center of the cluster is the centroid i.e. the average of all the points in the cluster. When the data has categorical attributes, centroid is not so meaningful and the center is often the medoid i.e. the most representative point of the cluster. The center of the cluster is commonly referred, as prototype and in such instances, center-based clusters can be

called **prototype-based clusters.** Center-Based clusters tend to be spherical. In Figure 1(b) it is shown an example of center-based clustering.

K-Means is a center-based, partional algorithm that attempts to find a user specified number of cluster (K), which are represented by their centroid. When we want to handle any distance metric we use k-medoids which uses centroids that belongs to the dataset.

An acceleration of k-means was proposed by Elkan [8]. It is referred that although k-mean is a fast algorithm, it is still slow because it requires kne distance computations, where k is the number of clusters, n the number of data points and e the number of iterations. It was proposed that if a point is far away from the center it should not be assigned to that and the distance should not be calculated. In addition if a point is much close to a center, it should be assigned to that. In order to these conclusions be excluded triangular inequality is used to obtain these upper and lower bound between the points and the centroids.

2.2.3. Graph-Based

Let's consider that the data is represented as a graph where its nodes are the objects and the links are the relationship among them. In that case we can define the cluster as a **connected component**[3] [1]. An example of graph-based cluster is **contiguity-based clusters**. In this type of clusters, objects are connected only if they are within a specified distance with each other. We make the conclusion that object in this clusters is closer to some object in the cluster than to any point outside the cluster. Figure 1(iii) shows an example of such clusters in two dimensions. This definition of clusters may be useful when clusters are irregular, but can have problems when noise is present, as shown between the two circular clusters in Figure 1(iii) a small bridge merges two distinct clusters.

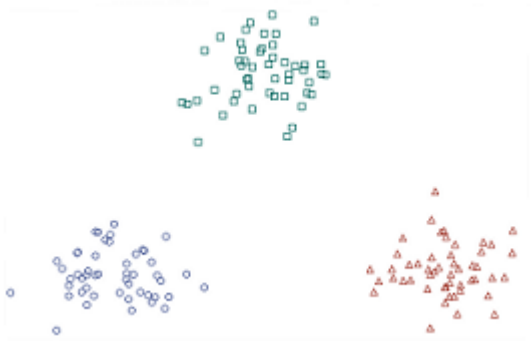
[3] A group of objects that are connected to one another, but they have no connections to objects outside the group

2.2.4. Density-Based

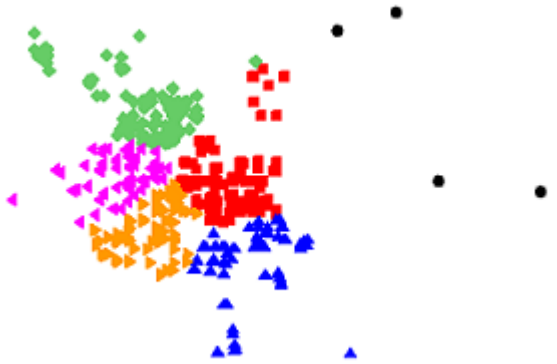
A cluster is a dense region of objects that is surrounded by regions of low density. This definition is at most used when the clusters are irregular and when noise and outliers are present. For example, Figure 1(iv) shows density-based clusters created by data of Figure 1(iii). The two circular clusters are not merged because the bridge between them is considered as noise. Likewise, the curve in Figure 1(iii) does not form a cluster because it is also considered as noise.

2.2.5. Shared-Property (Conceptual Clusters)

In general, we can characterize a cluster as a set of objects that share some property. This definition envelops all the past meanings of a cluster; for example, objects in a center-based cluster share the property that they are all closest to the centroid or medoid. In any case, the shared-property approach incorporates new sorts of groups. Think about the clusters shown in Figure 1(v). Points with different shape or color are adjacent. In this case a clustering algorithm needs a specific concept of properties in order to successfully detect these clusters. This process is called **conceptual clustering.** However, too complex a notion of a cluster would take us into the area of pattern recognition.



(i) Well-Separated Clusters: Each point is closer to every point in its cluster than to every point in other clusters



(ii) Center-Based Clusters: Each point is closer to the center of its cluster(centroid)

(iii) Graph-Based Clusters: Each point is closer to at least one point in its cluster than to any point in other cluster

(iv) Density-Based Clusters: Clusters are regions of high density separated by regions of low density

(v) Conceptual Clusters: Points in a cluster share some general properties

Figure 1. Different types of cluster illustrated by two-dimensional points

2.3. Algorithms

- **K-means**

K-means is a simple clustering algorithm [1].
The algorithm is as follows:

1. Select k random points as centroids
2. Put each point to the cluster corresponding to the nearest centroid
3. Reassign the centroid to the mean of the cluster
4. Go to step 2 until all the centroids remain unchanged

Usually with k-means we use a metric like the Euclidian distance or the cosine variance in order for the mean to make sense.

The complexity of k-means is $O(knI)$ where k is the number of clusters, n the number of points in the dataset and *I* the number of iterations. In practice we usually have a small k and we see that the algorithm converges in the first few repetitions therefor, we can consider *ki* as constant and we can think of k-means as being linear to the number of points in our dataset.

Also it is common to relax the condition that halts the program no points changing cluster (which is equivalent to the condition in step 4) to only 1% of the points changing cluster.

K-means is a fast and easy to understand algorithm that helps us gain some intuitions on the data that we are dealing with. It is important to note that k-means and all adjacent version of k-means, create convex clusters that may or may not, reflect the nature of our data.

- **K-medoid**

K-medoid is a variation of K-means with the intent to fix some of its drawbacks.

This variation improves k-medoid from the point of view of statistics since the median is not as sensitive to noise as the mean. For example, if we add to our dataset a single point that is far away from all the rest (an outlier), k-means is going to change the clusters drastically while k-medoid is going to ignore it.

Another reason for choosing k-medoid is the ability to handle any distance matrix. Therefor it is the only one that can work in any space even if for our data it doesn't make sense to talk about the mean of a set (e.g. molecules, the human face, etc)

- **PAM**

PAM or Partition Around Medoids is a K-medoid algorithm that is very closely related to k-means.
The difference between PAM and k-means is that in the initialization face (step 1 of k-means) we pick k random points from the dataset and not arbitrary points and at the reassignment (step 3 of k-means) we set the centroid, of the new cluster, to the median of the points in the cluster [9].

- **Clarans**

Clarans or Clustering LARge Application based on RANdomized Search, is an algorithm that was designed with the indent to make a more efficient algorithm for high-dimensional data. It produces a k-medoid clustering.

There are a lot of variations but they are all based on the same idea of using random elements and samples in order to find a clustering around medoids.

One of them is this algorithm that was in the original paper where tha algorithm was introduced [10]:

1. Input parameters: numlocal and maxneighbor. Initialize i to 1, and mincost to a large number.
2. Set current to an arbitrary node in the dataset.

3. Set j to 1.
4. Consider a random neighbor S of current and based on objective function, calculate the cost differential of the two nodes.
5. If S has a lower cost, set current to S, and go to Step 3.
6. Otherwise, increment j by 1. If j < maxneighbor, go to Step 4.
7. Otherwise, when j > maxneighbor, compare the cost of current with mincost. If the former is less than mincost, set mincost to the cost of current and set bestnode to current.
8. Increment i by 1. If i > numlocal, output bestnode and halt. Otherwise, go to Step 2.

- **Hierarchical**

For the hierarchical clustering algorithm there are two methods for the production of the clustering, agglomerative and divisive.

- **Agglomerative**

  This is a bottom-up approach. We begin with each point as its own cluster and we combine clusters to make the tree structure.

- **Divisive**

  This is a top-down approach. We begin with a single cluster that contains the entire dataset and we split it until all clusters contain a single point.

When compared to the k-means family of algorithm the hierarchical algorithm is of a higher complexity both in space and time, but we have a lot of advantages over them. For instance, the tree structure allows us to decide how good a “resolution” of the clustering is demanded for our application, meaning that we can decide, based on the nature of our data, when we should prune the tree. Also the hierarchical doesn’t constrain itself to convex clusters but allows for arbitrary shapes of each cluster.

- **Connected components**

Connected components are a way to find a agglomerative hierarchical clustering without the space and time complexity it demands. The algorithm works as follow: we

connect two clusters if their distance is smaller than a threshold. The distance between two clusters is the distance of the closest points in these clusters.

It is an easy but powerful idea. The clusters it produces are related by transitivity. The resulting cluster doesn't have the tree structure the hierarchical demands but it is a cut of the tree to the threshold we set.

We should note that there is a risk our clusters are thin conic section that span a wide area. This might be a problem and we have to be careful about the threshold under which we consider two points connected.

2.4. Evaluation

Using methods for evaluating a clustering of our data is useful for many reasons. First due to the fact that all clustering algorithms will give a result even if there is no structure in the data set (e.g. even if we have a uniformly distributed data-set) we want to know if there is a structure that has been capture by the specific algorithm we used. Many clustering algorithms take as input the number of clusters that exist in the data but in a real-life example we may don't know the correct number of clusters so there are methods and techniques for assisting in our endeavor. Another useful result would be to compare to clusters and find either how similar they are or which is better depending on our application. In our attempts to use an evaluation technique it is very important to disambiguate between the use of external knowledge meaning some ground truth not given to the algorithm a priori so that we could test the output or evaluating the result internally by looking at the clusters structure.

2.4.1. External Evaluation

In External evaluation we have access to some ground truth which we can use to validate our clustering and verify that the algorithm we used gave the desired result. Some common methods for measuring how close our result is to the ground truth are the following:

- **Entropy**

- **Purity**
- **Precision**
- **Recall**
- **F-mesure**
- **Roc**
- **K-fold Cross Validation**

2.4.2. Internal Evaluation

- **Root Mean Square Standard deviation**

RMSSTD [11] treats the points of our dataset as observed values of a random variable and for each cluster that represent the random variable we take the square root of the variance. It is calculated by

$$RMSSTD = \sqrt{\frac{\sum_i \sum_{x \in C_i} \|x - c_i\|^2}{|D| \sum_i (n_i - 1)}}$$

Equation 3: RMSSTD

RMSSTD quantifies the homogeneity of the clustering by resulting in a smaller value the smaller the variance of each cluster (meaning the closer the points are to the center of the cluster). Therefore the smaller the RMSSTD value the better cluster we have (assuming the desired cluster is structured by convex shapes)\
RMSSTD measures how compact the clusters are.

- **R-Squared**

RS [11] is closely related to RMSSTD. It was developed hand-in-hand with RMSSTD and instead of measuring the homogeneity of each cluster it measures the homogeneity between clusters. This means that RS can be consider as a measurement for dissimilarity between clusters. It is calculated by

$$RS = \frac{SS_b}{SS_t} = \frac{SS_t - SS_w}{SS_t}$$

Equation 4: RS

Where SS (Sum of Squares)

$$SS = \sum_{i=1}^{n} (X_i - \bar{X})^2$$

Equation 5: SS

And the subscript describes the set of points to be consider

- SS_w is the SS within groups

$$SS_w = \sum_C \sum_{x \in C} (x - \bar{x})^2$$

- SS_b is the SS between groups

$$SS_b = \sum_{x \in D} (x - \bar{x})^2 - \sum_C \sum_{x \in C} (x - \bar{x})^2$$

- SS_t is the SS of the whole dataset

$$SS_t = \sum_{x \in D} (x - \bar{x})^2$$

Where C is the set of clusters, D the set of all points in the dataset and $\bar{x}$ the mean of each dimension.

As we can see RS is the ratio of difference between groups relative to the difference of the whole dataset this explains our claim about RS measuring the homogeneity between clusters. RS ranges between 0 and 1 the higher the value of RS we have more distinct groups in our clustering thus having a better, more robust clustering.
RS index measures how well separated the clusters are.

- **Modified Hubert Γ statistic** [12]

Hulbert Γ statistic was developed by Lawrence Hulbert and Phipps Arabie in [12] and it was described as a generalization of the rand index which is a natural way of statistically comparing two clustering. Rand index was developed by many researchers

independently with slight variations which as Hulbert et all show they are different ways of measuring the same thing (i.e. instead of taking the number of indices where the two clusters agree some researchers chose the number of indices where the two clusters disagree and some chose a linear combination of the two). The Hubert statistic measures the correlation of two matrices drawn independently. It is defined as

$$\Gamma = \left(\frac{1}{M}\right)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} X(i,j)Y(i,j)$$

Equation 6: Hulbert Γ statistic

Where X and Y are $N \times N$ matrices and $M = \binom{N}{2}$ (the possible pairs)
Using this index we can now define the modified Hubert statistic index. We set each point x_{ij} of the matrix X as the distance between the cluster of i and the cluster of j. We will denote matrix X as Q from now on. The matrix Y will be the proximity matrix (distance matrix) and will be denoted as P. From that we can see that when our clustering is a compact one the value of Γ will be high and we expect that high values of Γ is a strong indication for compact clustering. [13] It is useful to notice that Γ is a monotonic function over the number of clusters since it counts the only pairs in different clusters so it will count $\frac{k-1}{k}n^2$ distances where k is the number of clusters
Γ index measures how well separated the clusters are.

In all internal evaluation indexes we have seen so far the function are monotonic as the number of clusters. So if we want to use this indexes to find the number of clusters that exist in a dataset we will have to look at the plot of their values relative to the number of clusters and find the knee that appears in the graph to find the number of clusters that exist in the dataset.

- **Calinski-Harabasz index**

Calinski and Harabazs crated an index [14] based on the idea that within a cluster the dispersion must be low – meaning that the points are close – and between clusters the

dispersion must be high - meaning that the clusters should be distinct and clearly separated.

In order to compute such a metric we need a way to calculate the within, as well as, the between cluster dispersion.

For WGSS (within cluster dispersion) we will add the squares of the distance from its point to its centroid:

$$WGSS = \sum_{k \in Clusters} \sum_{i \in k} (i - c(i))^2$$

Equation 7: WGSS

For BGSS (between cluster dispersion), for a cluster G, we will add the weighted distance of the clusters with weight the number of points in the cluster, i.e.:

$$BGSS = \sum_{C \in \{Clusters - G\}} N_k \|C - G\|$$

Equation 8: BGSS

Using this we can easily define Calinski-Harabasz index as follows:

$$CH = \frac{BGSS/(K-1)}{WGSS/(N-K)} = \frac{BGSS(N-K)}{WGSS(K-1)}$$

Equation 9: Calinski-Harabasz

- **Dunn's Index**

Dunn in his paper [15] suggested that a partitioning, in the case of convex clusters, that the distance between a point in a cluster and one in the clusters convex hall should be smaller than the distance of any point and a point in a convex hall of a different cluster. To simplify this, we can say that the minimum distance of two clusters divided by the maximum diameter of all of the clusters should be greater than one.

We can use this idea and create an index with this definition:

$$DI = \frac{\min\limits_{1 \le i < j \le k} \|C_i - C_j\|}{\max\limits_{1 \le m \le k} diam_k}$$

Equation 10: Dunn's Index

where **k** is the number of clusters and diam the diameter of kth cluster.

- **Silhouette index**

The silhouette coefficient [16] was designed as a graphical aid for representing the clusters quality. Silhouette combines the notions of cohesion and separation in an attempt to show which points lye well inside a cluster, which cluster are well formed, and also the overall quality of the particular clustering.

It is most commonly used as an aid for determining the number of clusters in a data-set we want to partition.
Silhouette is calculated by the following formula:

$$s(i) = \begin{cases} 1 - a\,(i)/b\,(i) & if a(i) < b(i) \\ 0 & if a(i) = b(i) \rangle \\ b\,(i)/a\,(i) - 1 & if a(i) > b(i) \end{cases}$$

Equation 11: Silhouette Index part

or equivalently:

$$s(i) = \frac{b(i) - a(i)}{max[a(i), b(i)]}$$

Equation 12: Silhouette Index

where a(i) is the average “dissimilarity” or distance to the point in its cluster and b(i) is the distance to the closest cluster after the one it belongs to.

Note that $1 \leq s(i) \leq 1$, also when s(i) is close to 1 we have a “perfect” clustering since we have the a(i) much smaller than b(i) and therefor every point clearly belongs to its cluster. When s(i) is close -1 we have the worst situation since every point is very close to some cluster different than the one they belong to.

- **David-Bouldin index**

David and Bouldin [17] argued that in order to have a good index for a clustering we need a similarity function for clusters. They proposed that this similarity function could depend on a clusters dispersion and the a distance between clusters making it of the form $R_{ij}(S_i, S_j, M_{ij})$, where $S_i$is the dispersion of Cluster i and $M_{ij}$is the distance between clusters i and j. Based on experience and frequently used heuristics in the field, they added some restriction to R - namely:

1. R is non-negative
2. R is symmetric
3. R is zero if both clusters are perfectly formed, meaning their dispersion is zero for both clusters
4. For the function $R_{ij}(Const, Const, x)$, as x increases, R decreases.

5. For the function $R_{ij}(Const, x, Const)$, as x increases, R increases.

The idea behind number 4 is that if we move two clusters further away they are less similar, and for number 5 the idea is the less coherent (or the worse formed) a cluster is the more similar it should be to other clusters.

David and Bouldin suggested an R function that satisfied all these restrictions as:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}$$

Equation 13: Example R

if we set $D_i = \underset{i \neq j}{max} R_{ij}$

then David-Bouldin index is:

$$DB = \frac{1}{N} \sum_{i=1}^{N} R_i$$

Equation 14: David-Bouldin index

for the dispersion and the distance between clusters we must use measures that make sense for the given dataset and desired clustering i.e. for data in Euclidean space with convex clusters, we can use the euclidean distance between center and for the dispersion the standard deviation

Comparison of Internal evaluation techniques

We will now show a comparison of internal evaluation techniques. We will leave out of the conversation the monotonic indexes and focus on the ones that their results depend exclusively their output (rather than the elbow that a researcher is responsible for identifying). The comparison for these indexes and a few more are done in [18] and we present a short summarize of their result.

	Noise	Variable Density	Existence of sub-Clusters	complexity
Calinski-Harabasz index	NO	YES	YES	$O(N)$
Dunn's Index	NO	YES	NO	$O(N)$
Silhouette index	YES	YES	NO	$O(N^2)$
David-Bouldin index	YES	YES	NO	$O(N)$

Table 1: Table with comparison of internal evaluation indexes

As far as the rest of the indexes (Root Mean Square Standard deviation, Modified Hubert Γ statistic and R-Squared) rely on the researcher to choose the optimal number of clusters as well as what each output means (why does such a curve occur? Is it due to actual clusters in the dataset or some degenerate case has occurred?). It is hard to define all the possible routes and give the intuition of what we should do but there is a very good and analysis on the heuristics in [1].

3. SCORING FUNCTIONS

Measurement is an essential precursor of all attempts to improve, information retrieval system, effectiveness. Comparative evaluations of a system are based on a number of key premises, including that systems can be sensibly compared depending on aggregate performance over a selected topic. Scoring functions are deployed to satisfy the needs of comparison. We can see each problem from a geometric perspective and according to the set of observations we use the particular function. The most common use of scoring functions is to compare high dimensional vectors which have weights in each dimension. We will now describe some scoring techniques, and present a comparison between them on some test data.

3.1. Weighted Sum

If a set of observations describes a phenomenon, it is natural to seek for an aggregate statistic that summarizes those observations. The simplest of these tendencies is arithmetic sum. It is a change of arithmetic mean without the division. We use this method of scoring because we have sparse vectors and when we have the same sum between two vectors, we need bigger score for the vector with more non-zero values. For a set of observation, $\{x_i \mid \mathrm{i} \in 1 \dots \mathrm{n}\}$ it is computed as:

$$AS = \sum_{i=0}^{n} x_i$$

Equation 15:Arithmetic Sum

Some points count stronger than others, given weights for each vector. In this case weighted sum is needed. For a set of observation as mentioned above and for a set of weights $\{w_i \mid \mathrm{i} \in 1 \dots \mathrm{n}\}$ weighted sum is computed as:

$$WS = \sum_{i=0}^{n} w_i * x_i$$

Equation 16:Weighted Sum

As example consider the four vectors with five observations in Table 1. The arithmetic sum of vectors V_1 and V_3 is equal, while arithmetic mean of V_3 is largest even if it has less non- zero values.

3.2. Geometric Sum

One point worth noting in connection with arithmetic sum is that all of the values should be on the same scale. It is not possible to sum over inches and centimeters without converting them in the same framework. An scoring function to avoid these problems is *geometric sum.* It is a change of geometric mean used for the same reasons as mentioned for the arithmetic sum.

Geometric sum is more stable than the arithmetic or weighted sum, in the sense of being less affected by outlying values. [19] However when any of values is set to zero geometric sum is also equals to zero. Because we have sparse data, we need to overcome this problem. So for observations $\{x_i \neq 0 \mid \mathrm{i} \in 1 \dots \mathrm{n}\}$ and for a set of weights $\{w_i \mid \mathrm{i} \in 1 \dots \mathrm{n}\}$, geometric sum is defined as:

$$GS = \prod_{i=0}^{n} x_i$$

Equation 17: Geometric Sum

and when we have some weights for each vectors as mentioned above it is computed as:

$$GS = \prod_{i=0}^{n} x_i * w_i$$

Equation 18:Weighted Geometric Sum

Further examples for geometric mean are shown in Table 1

3.3. Harmonic Mean

The harmonic mean is another central tendency that is typically used to combine rates, and can also be used as a method of score aggregation. The harmonic mean is

undefined if any of the set of values are zero. For that reason for a set of observations $\{x_i \neq 0 \mid \mathrm{i} \in 1 \ldots \mathrm{n}\}$ harmonic men is defined as the reciprocal of the average of the reciprocals,

$$HM = \frac{n}{\sum_{i=1}^{n} 1/x_i}$$

Equation 19:Harmonic Mean

Harmonic mean is closely related to arithmetic mean. It takes into account the largest sum of the values, the number of the observations and how uniform are the values allocated. For example, V_4 has sum close to the largest one, but it has more observation than V_1which has the largest sum, and it has uniform values. It is shown that V_4has the best score.

Sometimes we need to take into consideration some weights that are given for each vector. In these cases we need to compute another numeric aggregation, called **weighted harmonic mean**. For a set of weights $\{w_i \mid \mathrm{i} \in 1 \ldots \mathrm{n}\}$, and the above set of observations it is defined as

$$WHM = \frac{\sum_{i=1}^{n} w_i}{\sum_{i=1}^{n} \frac{w_i}{x_i}}$$

Equation 20:Weighted Harmonic Mean

Vectors	Values							AS	AM	GS	HM	SAW
V_1	4	4	2	1	0	4	4	**19**	**3.17**	512	2.4	10
V_2	1	3	3	2	2	2	3	16	2.28	216	2	9.99
V_3	4	2	2	4	2	4	1	**19**	2.71	512	2.15385	9.25
V_4	2	2	3	2	3	3	3	18	2.57	**648**	**2.47059**	**11.99**
Weights	3	1	3	2	1	3	1	-	-	-	-	-

Table 2: Example sets of values corresponding to different vectors applied to n=5 topics, and their calculated scores. The values in bold are the largest in each column.

3.4. Simple Additive Weighting

Simple Additive Weighting (SAW) which is also known as weighted linear combination is a simple and most often used multi criteria decision making technique. In this method, which is based on the weighted average, an evaluation score is calculated for each possible selection by multiplying the scaled value of each alternative with the weights of importance, followed by summing the products for all criteria. The advantage of this method is that it is a proportional linear transformation of the row data. [20] This means that the relative order of magnitude of the scores remains equal. Its process consists of the following steps:

Step 1:

- The first two stages of step 1 is to construct a comparison matrix for criteria and decide which of the two criteria is more important and assign a value. In our case is not obligatory because all the criteria have equal importance. Therefore they could have all 1 in the comparison matrix
- Next we have to compute the Weighted Sum Matrix by multiplying the comparison matrix and the vector with weights
- Divide all the elements of the weighted sum matrix by the respective weight vector element
- Compute the average of this value to obtain λ
- Find the Consistency Index, CI as follows

$$CI = \frac{\lambda - n}{n - 1}, where\ n\ is\ the\ matrix\ size$$

Equation 21:Consistency Index (SAW)

- Calculate the consistency ratio, CR as follows

$$CR = \frac{CI}{RI}$$

Equation 22:Consistency Ratio (SAW)

- RI can be obtained by the Table 2. CR is acceptable if it lower than 0.10. If it exceeds the pair-wise matrix is inconsistent and it should be improved.

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.85	0.9	1.12	1.24	1.32	1.41	1.45	1.51

Table 3: AVERAGE RANDOM CONSISTENCY (RI)

Step 2:

Construct a decision matrix (m x n) that includes m alternatives and n criteria. Calculate the normalized decision matrix for positive criteria:

$$n_{ij} = \frac{r_{ij}}{r_{jmax}} \; i = 1, \dots m \quad j = 1, \dots n$$

And for negative criteria:

$$n_{ij} = \frac{r_{jmax}}{r_{ij}} \; i = 1, \dots m \quad j = 1, \dots n$$

r_{jmax} is the maximum number and r_{jmin} is the minimum number of r in the column of j

Step 3:

Evaluate each alternative, A_i by the following folmula:

$$A_i = \sum_{j=1}^{n} w_j * n_{ij}$$

Equation 23:Final Score (SAW)

Where n_{ij} is the normalized score of the i_{th} alternative with respect to the j_{th} criteria. w_j is the weighted criteria. [20]

4. SET COVERING

4.1. The problem

We were tasked with providing search functionality over phrases with a large but strict vocabulary. We decided to see the problem from a geometric perspective, meaning that every phrase is a high-dimensional vector in a dimension of words.

To simplify this, we can say that, if our vocabulary was {A, B, C, D, E, F} then the phrase “A E F” is represented by the vector <1, 0, 0, 0, 1, 1>

From this we decided that in order, for the user, to identify the desired phrase we had to present to him, as few words as possibly that represented the entire set of phrases. Therefor the problem we had to solve was translated to this: find the minimum number of words that if combined represent the dataset. And after each iteration (after the user chooses a word), we would take the Intersection of the current set with the set that contain the chosen word.

The set cover Problem is defined as follows:

Given

Ground elements $U = \{u_1, u_2, u_3, \dots, u_n\}$,

subsets $S = S_1, S_2, \dots, S_k \subset U$ and

Costs $c: S \to Q^+$

Find a set $I \subseteq \{1,2,3, \dots, m\}$ to minimize $\sum_{i \in I} c(S_i)$ with the property $\cup_{i \in I} S_i = U$

We cat set the $u_i$with the ith phrase, the $S_i$with the set of phrases that contain the ith word and We can leave $c = 1$for all S.

Set covering is a hard problem, it was one the first problems to be shown to be NP-complete from Karp[ref Reducibility Among Combinatorial Problems 1972] it is very well studied and has many approximation algorithms that we can use to solve our version of it.

4.2. Approximation/Greedy Algorithm

There many approaches to approximating the set cover problem [21]

The most straightforward algorithm was presented by David Johnson [22]. At each iteration we choose the element that maximizes the number of elements we cover.

To be more precise:

1. SUB $= \emptyset, UNCONV = U$ and $SET(i) = S_i, 1 \leq i \leq N$
2. if $UNCONV = \emptyset$ exit
3. choose $j \leq N$ such that $|SET(i)|$ is maximized
4. set $\text{SUB} = \text{SUB} \cup SET(j)$, $UNCONV = UNCONV - SET(j)$ and $SET(i) = SET(i) - SET(j), 1 \leq i \leq N$
5. Go to 2

The complexity of this algorithm is $O(|U|N)$

Another approach is to use linear programming to solve this problem [23]. In order to do that we have to turn our problem to the canonical form.

Minimize $\sum_{S_i \in S} c(S_i)x_i$

subject to $\sum_{S_i : e \in S_i} c(S_i)x_i \geq 1$ for all $e \in U$ and

$x_i \in \{0,1\}$ for all i

in order to solve this efficiently we will solve the LP-relaxation which is

Minimize $\sum_{S_i \in S} c(S_i)x_i$

subject to $\sum_{S_i : e \in S_i} c(S_i)x_i \geq 1$ for all $e \in U$ and

$x_i \geq 0$ for all I

After that we can solve it by solving the dual and using standard linear programming techniques find a solution for the original.

5. APPLICATIONS

We worked for a company (vithoulkascompass.com) that worked on creating an expert system for suggesting homeopathy remedies to patients with a particular set of symptoms. The machine learning problem arose from this endeavor. We had available a matrix of remedies and symptoms with the values in each cell being a number from 0 to 4 measuring the effectiveness of each remedy to each symptom. It is important to note that the given matrix was sparse and there were many more symptoms than remedies.

$$|symptoms| \gg |remedies|$$

We began by attempting to find some structure in the dataset with an unsupervised clustering algorithm. To do this we had two choices, we could see the data as many, low dimensional, vectors (by viewing each symptom as a vector in the space of remedies) or a few, high dimensional, vectors (by viewing each remedy as a vector in the space of symptoms).

We analyzed the data in order to find a distance function that made sense for our data and then we analyzed the resulting space (from the combination of vectors and a distance function) to figure out which evaluation technique was the best fit to solve our instance of this problem. In order to assert our decisions, we talked with the experts (doctors) in order to clarify and verify our results.

Another task we were asked to undertake was a search problem that was needed in order for a doctor or a patient to find the symptoms that correspond to each case. We developed a prototype that attempted to solve this for people that weren't necessarily aware of the field's terminology.

REFERENCES

[1] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Boston: Pearson Addison Wesley, 2005.

[2] J. Han, M. Kamber and J. Pei, Data mining: concepts and techniques, Morgan Kaufmann Publishers, Inc., 2001.

[3] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, New York: Wiley, 1990.

[4] R. T. Ng and J. Han, Efficient and Effective Clustering Methods for Spatial Data Mining, Vancouver: University of British Columbia. Department of Computer Science, 1994.

[5] G. Sheikholeslami , S. Chatterjee and A. Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases," *The VLDB Journal The International Journal on Very Large Data Bases,* vol. 8, no. 3-4, pp. 289-304, 2000.

[6] J. C. Bezdek, R. Ehrlich and W. Full, "FCM: The Fuzzy c-means Clustering Algorithm," *Computers & Geosciences,* vol. 10, no. 2-3, 1984.

[7] S. Ghosh and S. K. Dubey, "Comparative Analysis of K-Means and Fuzzy CMeans Algorithms," *International Journal of Advanced Computer Science and Applications IJACSA,* vol. 4, no. 4, 2013.

[8] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proc. ICML*, 2003.

[9] L. Kaufman and P. J. Rousseeuw, Clustering by means of medoids, Delft: Faculty of Mathematics and Informatics, 1987.

[10] N. T. Raymond and J. Han, "Efficient and effective clustering method for spatial data mining," *VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases,* pp. 144-155 , 1994.

[11] S. Sharma, Applied Multivariate Techniques, New York: J. Wiley, 1996.

[12] L. Hubert and P. Arabic , "Comparing," *Journal of Classification ,* 1985.

[13] S. Theodoridis and K. Koutroumbas, Pattern Recognition, San Diego: Academic Press, 1999.

[14] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Comm. in Stats. - Theory & Methods LSTA Communications in Statistics - Theory and Methods,* vol. 3, no. 1, 1974.

[15] J. Dunn, "Well separated clusters and optimal fuzzy partitions," *Journal of Cybernetics,* vol. 4, no. 1,

pp. 95-104, 1974.

[16] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Delft: Dep. of Mathematics and Informatics, Univ. of Technology, 1984.

[17] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence IEEE Trans. Pattern Anal. Mach. Intell.,* Vols. PAMI-1, no. 2, pp. 224-227, 1979.

[18] Y. Liu, Z. Li, H. Xiong, X. Gao and J. Wu, "Understanding of Internal Clustering Validation Measures," *Data Mining (ICDM), 2010 IEEE 10th International Conference on,* pp. 911 - 916, 2010.

[19] S. D. Ravana and A. Moffat, "Score Aggregation Techniques in Retrieval Experimentation," *ADC '09 Proceedings of the Twentieth Australasian Conference on Australasian Database,* vol. 92, pp. 57-66 , 2009.

[20] A. Afshari, M. Mojahed and R. M. Yusuff, "Simple Additive Weighting approach to Personnel Selection problem," *International Journal of Innovation, Management and Technology,* vol. 1, December 2010.

[21] V. V. Vazirani, Approximation Algorithms, Berlin: Springer, 2003.

[22] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Proceedings of the fifth annual ACM symposium on Theory of computing - STOC '73,* 1973.

[23] . D. . S. Hochbaum, "Approximation algorithms for the set covering and vertex cover problems," *SIAM J. Comput. SIAM Journal on Computing,* 1982.

[24] T. Kärkkäinen and S. Äyrämö, "Introduction to partitioning-based clustering," 2006.

[25] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *The Computer Journal,* vol. 16, no. 1, pp. 30-34, 1973.

[26] M. Halkidi, M. Vazirgiannis and I. Batistakis, "Quality scheme assessment in the clustering process," *Principles of Data Mining and Knowledge Discovery Lecture Notes in Computer Science,* pp. 265-276, 2000.