

Development of a Hierarchical K-Selecting Clustering Algorithm – Application to Allergy

K.N. Sudeepthi¹, M. Suresh Babu², D. Asha Devi³

¹Asst Professor, Intel Institute of Science, Anantapur, India,

²Principal, Intel Institute of Science, Anantapur, India

³Associate Professor, Department of ECE, Intell Engineering College, Anantapur, India

Abstract: A major type of allergy is mediated by immunoglobulin E (IgE) antibodies and is known as IgE-mediated allergy, immediate- or type I hypersensitivity. An allergic reaction can occur when a sensitized individual encounters the allergen to which he/she has specific IgE antibodies directed. Cross reactivity is a phenomenon that occurs when IgE antibodies originally raised against one allergen binds to another source. Under certain and quite commonly fulfilled conditions this leads to an allergic response against one or several proteins that the individual has no prior contact with. In vitro test assays that quantitatively measure serum levels of specific IgE antibodies are commonly used in the allergy diagnostic setting. Recently, multivariate analysis of a data set holding IgE measurements in over 1000 multireactive individuals to a panel of 89 allergen extracts was reported. Several aggregations of potential cross-reactive allergen extracts were obtained with K-means clustering.

1. Introduction

The need for appropriate computer based tools for multivariate data analysis has increased together with boosted quantity of information that e.g. researchers, developers and medical personnel are faced with. Because of this general trend it has become increasingly difficult, not to mention time consuming, to manually discern meaningful patterns in data sets. Seemingly, there is much to gain with the use of good, robust and easy to use pattern recognition algorithms. Increasing need for adequate data processing, however, entails tougher demands on software reliability, stability and functionality. An easy-to-grasp and reliable representation from a large and complex body of data, which can be used as a foundation upon which decisions are later made, is currently possible to achieve in many cases. Clustering is a type of analysis especially suitable for large datasets of multivariate data. The purpose of a clustering algorithm is to find partitions in data through the use of some kind of similarity measure. Two of the most common clustering methods are called K-means and Hierarchical clustering. They represent two separate approaches towards the partitioning of data. K-means is a partitioning method, which produces a grouping of data into a redefined number of clusters using an error criterion, whereas hierarchical clustering uses a similarity measure to merge pairs of clusters together. The result from K-means is a list of cluster affiliation for each data point, whereas hierarchical clustering returns a tree structure showing nested groupings. Both methods have strengths and weaknesses. K-means is an acknowledged tool when a data set is to be grouped into a specific number of clusters. In some instances, however, it is less useful because of its flat representation of data. Hierarchical clustering is very useful because it produces a stepwise visualization of the clustering process, giving the user the option to observe groups formed at different levels. It is, however, limited to joining two clusters at a time, a constraint that often leads to messy and unclear representations of data.

In a recent publication, cluster analysis (K-means) was used to find cross-reactive allergens. Within allergy, cross-reactivity of B-cell type is a phenomenon that relates to binding of immunoglobulin class E (IgE) antibodies, raised against a certain antigen, to another source due to structural similarities to the sensitizing antigen. The IgE antibody is a key molecule in the cascade of events that ultimately triggers an allergic response and is thereby also important to allergy diagnostics. In the study, a data set that contained quantitative measurements of IgE antibodies from a number of multi-sensitized individuals to a wide range of allergen preparations was studied by means of K-means to identify aggregates of allergen preparations. Although the article presents some very interesting results no hierarchical information on data cluster were obtained clearly. A new clustering tool that in a reliable way could generate hierarchical information is needed, which became a major incentive for this project.

2. Clustering

Clustering is defined as the unsupervised and exploratory classification of samples (observations, data items, or feature vectors) into groups (clusters). Application areas range from data mining, document retrieval, image segmentation to grouping of biological data. The clustering procedure can be divided into several steps.

1. **Specification of sample representation:** The first step involves specification of the sample representation, i.e. how the data to become subjected to treatment should be described. Perhaps there is a subset of the data that better describes the problem or, alternatively, must be excluded from the analysis due to missing or insufficient data annotation. Additionally, there may be a need to homogenize the data e.g. through the use of logarithm transformation or by means of other operations.
2. **Defining approximate Sample proximity measure:** The second step (after sample representation) in the clustering process is to define an appropriate sample proximity measure, i.e. a method to determine distance (or similarity) between two samples. There is a wide variety of methodological directions to choose among, a major measure being the Euclidean distance.
3. **Clustering:** The third step in the clustering process involves the actual clustering. Presentation of output from the clustering step can be adapted to the investigators need. It can e.g. be hard (with all data partitioned into groups), fuzzy (where each pattern has a variable degree of membership) or hierarchical (a nested series of partitions based on a criterion for merging or splitting clusters). Other techniques include probabilistic and graph theoretic approaches.
4. **Data abstraction:** In some cases, additional analysis of the clusters is needed. After completion of the clustering procedure data abstraction may be useful. This can include a number of adaptations ranging from restructuring the output data to extended data analysis to process the result in order to make it easier to interpret by visual inspection.
5. **Cluster validation:** Moreover, it is often desirable to perform a cluster validation, in which analysis is made to confirm the result of the clustering step. In most cases no “gold standard” is at hand so this is often a subjective phase based on how much is known about the data and how reasonable the result seems. There are, however, objective cluster validation methods and some of these have been tested in this thesis. Some of the points mentioned above pose great challenges to users of clustering methods. The questions how a data set should be pre-processed, which similarity measure is appropriate or which of the numerous clustering algorithms that best fits the situation are not easily answered, especially since a single correct answer is rarely available. Thus, no single clustering technique is universally applicable to uncovering the variety of structures present in multidimensional data sets. Each new clustering method typically performs slightly better than those already established on a specific distribution of samples. Therefore, it is of great importance that users of clustering algorithms are well versed in the technique that the particular algorithm utilizes. However, a sound understanding of the data gathering process and a certain degree of domain expertise is just as important as the technical knowledge.

2.1. Initiation techniques

Many clustering techniques, some of which will be described in chapter 2.2.3, require an initial partition of data. The problem at hand is to find k initial groups in a set of n data samples. There are several ways to approach this problem.

1. **Sample or random k-selection:** One of the most straightforward ways to create an initial partition is through the *sample*, or *random k-selection*, method. This method randomly selects the first among k desired initial vectors from the n data samples. Then, it randomly selects the next vector out of the $n-1$ remaining samples and so on.
2. **Uniform distribution method:** The *uniform distribution* method is a second alternative to achieve initial partitioning. This method uses a random selection taken from a range defined by minimum and maximum values of all features in the dataset. This results in initiation points that are uniformly distributed over the space spanned by the dataset.
3. **Subset clustering:** *Subset clustering* is another widely used initiation method. This algorithm works by first randomly selecting a subset of the data, e.g. 10 percent, followed by a clustering iteration on the subset using a simple method, such as the sample algorithm, for initiation. The resulting cluster centers are then used as starting points for global data clustering.
4. **Leader Cluster algorithm:** Finally, the *leader* cluster algorithm is technique that partitions a data set into groups by virtue of a radius distance, T . A leader object is associated with each group and all other objects in the group lie within the distance T from that object. This method is illustrated in figure X. The algorithm starts by selecting the first data point and assigning it as the first leader object, A. Subsequently, the remaining samples are examined and those that are within the distance T are assigned to group one. The first data sample examined that fall outside the radius T is assigned as the next leader object, B. This procedure is iterated to identify cluster centre C as well as the remaining centers. This algorithm has the advantage of being fast, since it only requires processing the data once. All cluster centers are at least a distance T from each other, which means that it is dependent on the ordering of the data set and that a distance (T) is specified rather than the number of clusters.

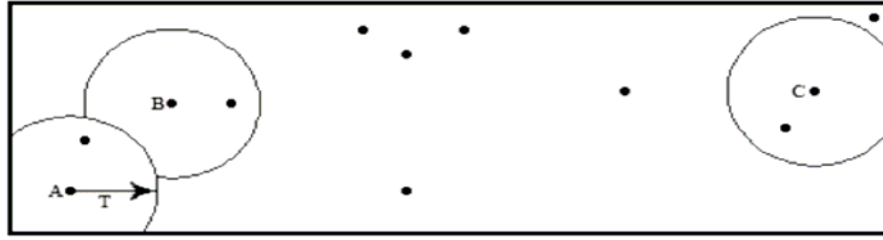


Figure 1: Leader Clustering

2.2. Sample proximity measures

As mentioned above the choice of proximity measure can yield great variations in the resulting clusters. A few of these measures are

1. **Euclidean distance:** The *Euclidean distance* is one of the most common proximity measures. The Euclidean distance between points $P = (p_1, p_2, \dots, p_n)$ and $Q = (q_1, q_2, \dots, q_n)$, in Euclidean n -space, is defined as

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}. \quad (1)$$

A commonly used alternative to this distance measure is the *squared Euclidean distance*.

2. **Mahalanobis distance:** Another well known measure is the *Mahalanobis distance*. Unlike the Euclidean distance, that of Mahalanobis takes the correlations of the data set into account. Moreover, it is scale invariant, i.e. not dependent on the scale of measurements. The Mahalanobis measure for a group of values with mean $m = (m_1, m_2, \dots, m_n)^T$ and the covariance matrix C for a multivariate vector $x = (x_1, x_2, \dots, x_n)^T$ is defined as

$$D_M(x) = \sqrt{(x - m)^T C^{-1} (x - m)}. \quad (2)$$

If the covariance matrix is the identity matrix the Mahalanobis distance is reduced to the Euclidean distance

3. **Jeffrey- Kullback-Liebler divergence:** A dissimilarity measure found in information theory and probability theory is the *Jeffrey- Kullback-Liebler divergence*. This measure is also known as the information gain and the relative entropy and is defined in its asymmetric version as

$$J = D_{JKL}(p_1 \parallel p_2) = \int_x p_1(x) \log\left(\frac{p_1(x)}{p_2(x)}\right) dx, \quad (3)$$

where

p_1 and p_2 denotes the probability density functions for P_1 and P_2 . The JKLD measures the difference between a “true” probability distribution P_1 and an arbitrary probability distribution P_2 . Even though it is often used as a distance measure the JKLD is not a true metric since it is not symmetric, thereby being referred to as divergence rather than distance. There is, though, a symmetric version of the measure, which is defined as

$$D_{SKL}(p_1, p_2) = \int_x (p_2(x) - p_1(x)) \log\left(\frac{p_1(x)}{p_2(x)}\right) dx. \quad (4)$$

2.3. Clustering techniques

Different approaches for data clustering can be described with the help of the hierarchy in figure 2-6. At the top level the first distinction is between hierarchical and partitional approaches (hierarchical approaches produce a nested series of partitions whereas a partitional approach produces only one partition). Aside from the taxonomy in figure 2-6 there are a few more issues that may affect all the different approaches regardless of their placement.

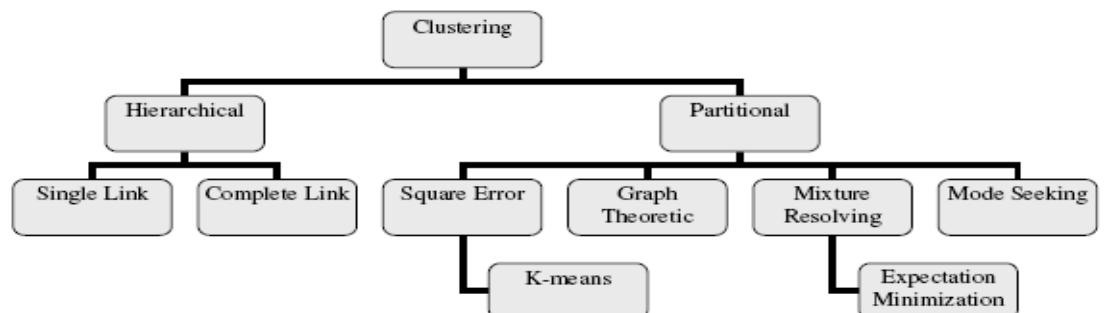


Figure 2: A Taxonomy of Clustering approaches

Agglomerative vs. divisive: This is related to the algorithmic structure and operation. An agglomerative approach involves that all data samples are initially defined as single clusters. They are then gradually merged until a specified criterion is fulfilled, leading to an arrest of the process. A divisive method starts with all samples in a single cluster and then proceeds by splitting that cluster into groups until a stopping criterion is met.

Monothetic vs. polythetic: This aspect relates to the sequential or simultaneous use of features in the clustering process. In other words, does the algorithm take all features into consideration when clustering data samples or does it look at them one at a time? Most algorithms are polythetic and thus use all features to decide the distances between groupings. Monothetic algorithms have a major drawback in that they generate 2d clusters where d is the dimensionality of the data.

Hard vs. Fuzzy: A hard clustering algorithm assigns each sample in a data set to a single cluster, whereas a fuzzy clustering method assigns various degrees of membership in several clusters. A fuzzy clustering can be converted to a hard counterpart by assigning all samples to the clusters in which they have the strongest membership. In the following text a few different clustering methods and their operation will be described in greater detail.

K-means

K-means is a partitional clustering algorithm. The number of clusters that the algorithm will find is defined by the user. K-means starts by creating an initial set of cluster centers using any among several sorts of initiation methods. Then the following two steps are alternated until convergence is achieved:

- Each sample is assigned to the cluster, to which cluster centre it is closest.
- The mean value of all features is calculated for each cluster. The resulting vector becomes the new centre for that cluster.

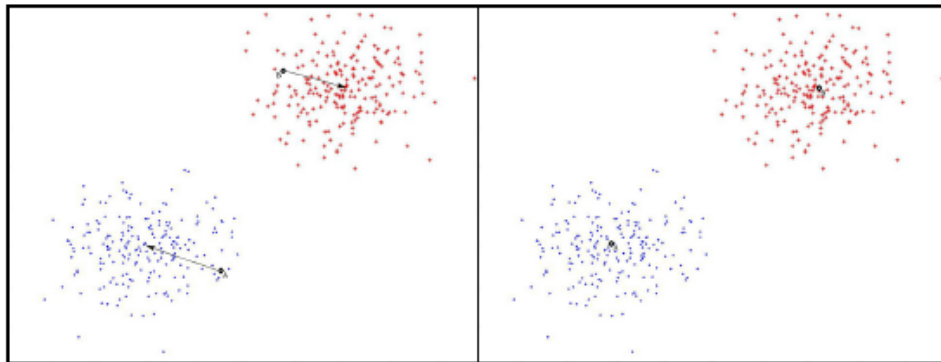


Figure 3: Movement of Cluster centers in k-means.

Vector Quantization

A family of algorithms, closely related to K-means clustering, is referred to as vector quantization (VQ). Perhaps most well known of these are the Linde-Buzo-Gray (LGB) and self organizing map (SOM) algorithms. VQ algorithms aim at minimizing the objective function

Hierarchical Clustering

The operation of a hierarchical clustering algorithm is illustrated in figure 2-10. This figure depicts five samples - A, B, C, D and E - grouped into three clusters and the dendrogram that is the result of the agglomerative clustering. A dendrogram illustrates the nested groupings of samples and the similarity levels at which groupings change. Thus, it is possible to split a dendrogram to obtain the corresponding clusters for that particular level of similarity. The most popular forms of hierarchical clustering algorithms are the single-link and complete-link algorithms. These two algorithms differ with respect to how they characterize similarity between a pair of clusters. The single-link method defines the distance between two clusters as the minimum of distances between all pairs of samples. Based on the example, as depicted in figure 4, the distance between the sample A and the cluster containing the B and C samples would be equal to the distance between A and B. By contrast the complete-link algorithm describes the distance as the maximum of all pairwise distances between samples in two clusters which, if applied on the same example as above, would correspond to the distance between A and C. The characteristics of the resulting dendrogram are accordingly highly dependent on the type of linkage used. The complete-link algorithm tend to define tightly associated and compact clusters whereas that of single-link suffers from a chaining effect and has a tendency to produce clusters that are straggly or elongated.

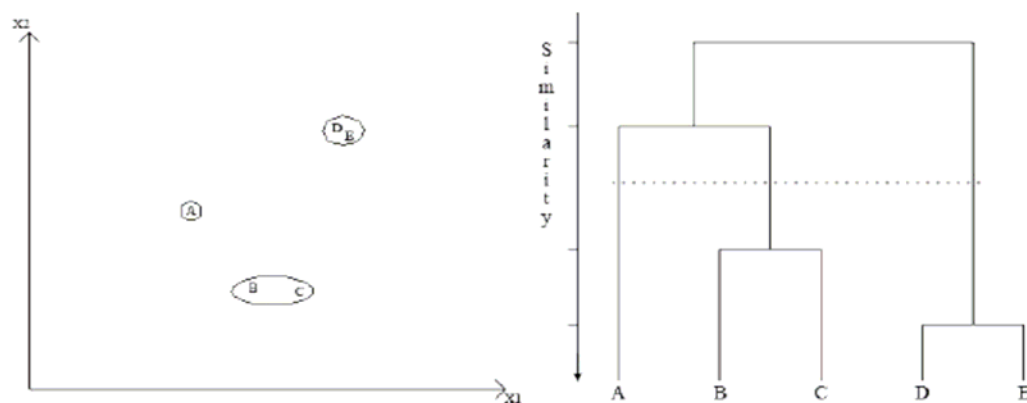


Figure 4. Points falling into three clusters (left) and the resulting dendrogram obtained using the single-link algorithm (right).

Hierarchical clustering is a commonly used method, familiar to most biologists through its application to sequence and phylogenetic analysis. Apart from the dendrogram there is another way to visualize results from clustering, called clustergram. A clustergram combines a dendrogram created through hierarchical clustering with a heatmap, which translates variation into colors, e.g. fluorescence intensity reading. This is often performed over time so that an increase in e.g. fluorescence can be seen in the heatmap over the course of a test. This technique enables visualization and identification of genes that are active under certain conditions.

Cluster validation techniques

Clustering algorithms can produce a partitioning of data into k clusters and different approaches as to how this can be achieved have been outlined in the previous chapter. However, it can be difficult to identify an appropriate k . The human eye is a good validating tool for data represented in two and maybe three dimensions, but it is not a feasible way to verify data of higher dimensionality. Consequently, in order to find the best number of clusters a user either has to have a deep understanding of the data beforehand and by trial and error optimize parameters to obtain an understandable clustering or acquire an empirical measurement of the “correctness” of the clustering result by means of a cluster validation technique. There are different approaches to the problem of cluster validation.

- o Some techniques rely on a difference measurement between two clustering results, using different number of clusters.
- o Others only depend on the actual partition of the samples and return a numerical indication on the correctness for each value of k . Three approaches to cluster validation will be presented below.

Silhouettes

The silhouette method is a graphical cluster validation technique, which represents each cluster from a partition with a so-called silhouette. The silhouette method works by comparing clusters with respects to tightness as well as separation from other clusters

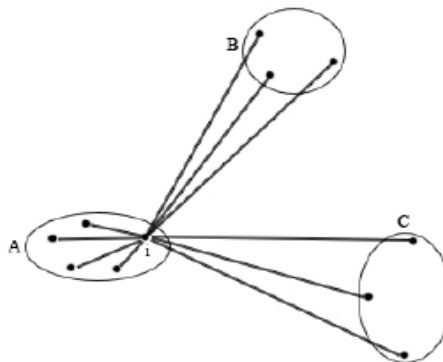


Figure 5. An illustration of the elements involved in the computation of $S(i)$, where the object i belongs to cluster A.

Ratio

The ratio method finds the optimal number of clusters by looking for what is referred to as a “knee” of an error curve. This curve is typically generated from values of a criterion function for a series of cluster processes, using an increasing number of clusters. The result of the criterion function (e.g. sum of squares,

J) is plotted against the number of clusters in order to produce a graph, similar to that of figure 6. The ratio method then divides the error value for k ($k = 2, 3, 4, \dots$) with the error value for $k-1$ in order to get the ratio between them. The “knee” is defined as the k with the lowest ratio value, because this indicates the number of clusters that leads to the greatest increase in correctness when compared with the number of proceeding clusters. This can be compared to a procedure, which finds a point that shows the largest deviation from a standard curve drawn through the points in the plot. In the figure this corresponds to $k = 4$, which clearly coincides with a local bend in the curve. This method is a local knee finding technique, as it only looks at two points at a time.

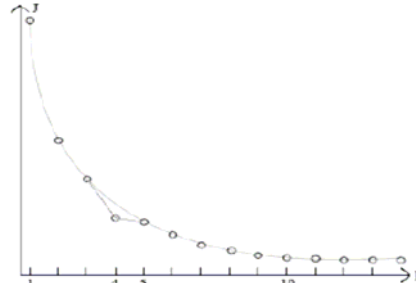


Figure 6. An “error” / “number of clusters” curve indicating a “knee” at $k = 4$.

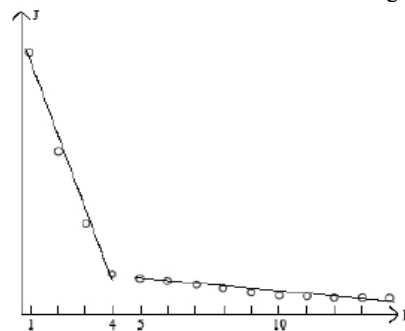


Figure 7. The result of the L-method adapting two lines to the plotted data.

Material and methods

Data set

Two different data sets were used in this study to evaluate various methods for cluster analysis, choice of initiation points and selection of optimal number of clusters, k . A two dimensional data set was manually generated (simulated), figure 8, with the purpose of providing a clear visualization of clustering results. This data set was constructed in such a way that it should return a meaningful hierarchical tree-structure after cluster analysis, i.e. the resulting tree should start by branching into three followed by further segmentation into two, four and six groups respectively.

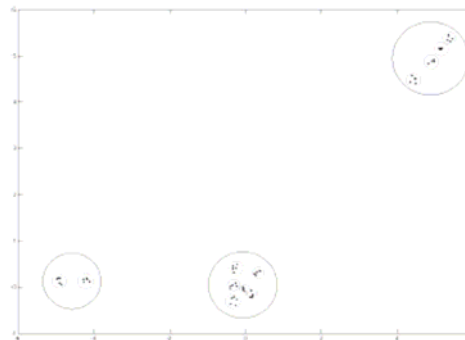


Figure 8: Two dimensional data set used for development and testing.

The second data set was an excerpt of a repository containing measurements of IgE reactivity in human sera from sensitized, but otherwise unidentifiable individuals. The entire database contains about 49,000 samples and has been compiled for the research, development, and quality assessment of ImmunoCap™ (Phadia AB, Uppsala, Sweden). The reduced data set possessed two characteristics that made it especially appropriate for the development and testing of the study outlined in this thesis. Firstly, it was of a suitable size (1011 individuals with IgE values for 89 different allergen preparations) and, secondly, it had been previously studied in a published article, [29]. Accordingly, it was well known beforehand, making interpreting results derived from this study much easier. Moreover, the individuals typically show rather high levels of specific IgE antibodies (sIgE) to multiple allergenic extracts. The types of allergen sources targeted by serum IgE molecules of this cohort are food, insect, grass pollens, weed pollens, mites, micro-

organisms (e.g. *Penicillium notatum*) and tree pollens. Because of a great variation in the IgE values between different individuals/allergens the data was also ranked prior to further analysis. Ranking involves a categorization, which implicates that the individual with the lowest IgE value for a certain allergen is assigned the value “1” for that allergen, the individual with the second lowest IgE value for the same allergen is assigned the value “2” and so on. Conversion of raw data to logarithm values is a common alternative to ranking and was also used in this work to homogenize IgE read-outs.

Evaluation of initiation techniques

In order to obtain a clear view on the importance of starting method three different initiation techniques were evaluated: (see also chapter 2.2.1)

- _ *Sample*
- _ *Uniform distribution*
- _ *Modified leader algorithm.*

The original leader algorithm lacks the ability to specify a number of clusters beforehand, something that was essential for this project. Therefore, a modified version, enabling this function, had to be implemented. In conformity with the original leader algorithm the modified leader version initially uses a radius to find those samples that are close enough to a randomly selected leader object. These data are thereafter grouped as a cluster and are excluded from the next iteration. The modified leader does, however, terminate only on the condition that all samples have been assigned to any one cluster and the specified number of clusters have been achieved. Thus, when the scenario, specified by the middle image in figure 3-2 arise, the modified leader algorithm takes the sample that is furthest from its assigned cluster centre and reassigns it as a member of the newly formed cluster. Then, in a final step, all samples are reassigned to become a part of the cluster centre closest to them. If the opposite scenario, that not all samples are assigned when the predefined number of clusters has been calculated, should arise, the algorithm is restarted using a larger radius.

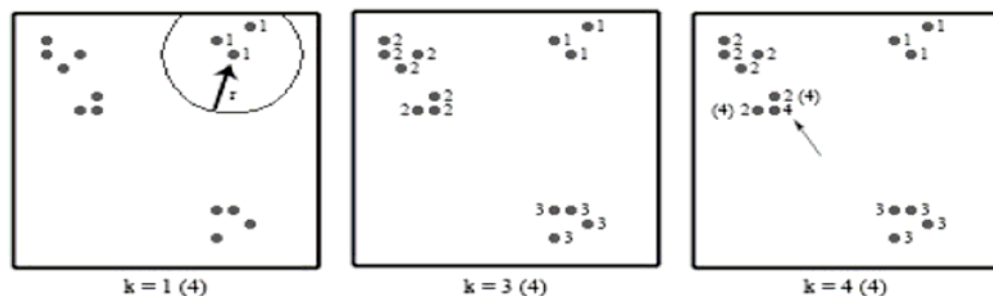


Figure 9: Step by step overview on how the modified leader algorithm groups the initial data.

In the first step a random point is chosen and the samples within the calculated radius are assigned to the first cluster (left). If all samples have been assigned to clusters before the sought of number of clusters have been achieved (middle) the data point furthest from its cluster centre is assigned as the next cluster centre. Then new cluster partitions are then calculated, the numbers surrounded by parenthesis indicates the new partition (right).

Comparison of clustering methods

The selection of a robust and reliable clustering algorithm was of great importance for this project. Therefore the selection of clustering method was a central part of the development. Subsequent to a serious consideration of a variety of optional directions, four different clustering methods were eventually selected for evaluation:

- _ *K-means*
- _ *Vector quantization*
- _ *Log-likelihood*
- _ *JKLD facilitated clustering*

The symmetric Jeffrey-Kullback-Liebler Distance was implemented to be used as a clustering method. As the JKLD measures the separation between Gaussian distributions the idea was that clustering could be achieved through the optimization of the JKLD for a certain amount of clusters

Evaluation of k-selection techniques

The identification of the optimal number of clusters at each level of operation was another important challenge of this study. Three different types of k-selection techniques were evaluated:

- _ Silhouette
- _ Ratio
- _ L-method

Clustering visualization

One among the most important objectives of this project was to enable adequate visualization of the clustering process. Therefore it was considered essential that a tool, which could illustrate each step of the iterative hierarchical clustering algorithm, be created. There was also an interest to arrive at a tool that would give a quick overview of aggregations in observed data. These two needs ultimately resulted in the creation of two visualisation aids, a dendrogram-like illustration tool and a clustergram-like illustration tool.

Dendrogram-like illustration

Plot is a dendrogram inspired tool and it comes as both an interactive and a non-interactive version. The algorithm can draw trees from stem to leaf, showing each clustering step that has been made along the process. Because the iterative hierarchical clustering algorithm is able to arrange a data set into n new clusters ($n \gg 2$) there existed no comprehensible way to use a distance measurement between points as that used when drawing a dendrogram (see chapter 2.2.3). So, in order to acquire any kind visual information related to the clustering procedure a measure of the extent of scatter within clusters was designed. The measure was calculated as follows: the mean of the within cluster distances between member samples, x_q , and their corresponding centroid, m_k , was divided with value the same measurement for all N initial data points. This can also be formulated as

$$r = \frac{\frac{1}{N_k} \sum_{x_q \in C_k} \|x_q - m_k\|^2}{\frac{1}{N} \sum_{i=1}^N \|x_i - m\|^2}, \quad (14)$$

where C_k represents cluster k , N_k represents the number of members in cluster k and m represents the mean value for all initial samples. Thus, each branch length represents the ratio, r , of scatterness between its corresponding cluster and total data-set. From a visual aspect this translates to comparatively extended lines on the plot for widespread clusters and vice versa.

This was found useful, since it clearly showed clusters containing outliers or, alternately, clusters containing relatively unrelated data. An example of the resulting plot can be seen in figure 3-3.

Clustergram-like illustration

Clustergram is a modification of the MATLAB™ (Mathworks, Natick, MA) function clustergram, being part of the bioinformatics toolbox. The rewrite replaces the dendrogram function, being the default clustering method for the MATLAB™ version, with the iterative hierarchical clustering algorithm, developed throughout this project.

4. Results

Comparison of clustering methods

The development and evaluation of different clustering techniques was perhaps the most time consuming part of this project. Each method was applied to both the two-dimensional and the ImmunoCap™ data-sets (chapter 3.1) to see how well they grouped data. All four clustering algorithms listed in chapter 3.3 were exhaustively tested. Accordingly derived results are outlined below. Because K-means is such a common clustering method it was an obvious candidate to include in the evaluation procedure. It comes with the statistics toolbox from MATLAB™ and was readily available without much preparation. The MATLAB™ implementation of K-means uses a two phase algorithm while clustering. Apart from the standard K-means algorithm (see chapter 2.2.3) this version also accomplishes a second phase, called the “on-line” updating phase. In this phase points are individually moved to other clusters in order to assess whether a lower sum of errors can be found. This second phase can correct many inappropriately identified configurations where K-means have found a local minimum, thereby making the algorithm more stable. The MATLAB™ version is also fast and offers a range of different alternatives with regard to distance measurements, and a multitude of solutions to various issues that might arise.

Because no prewritten vector quantization algorithm existed in MATLAB™ the algorithm had to be implemented. The basic VQ algorithm (see chapter 2.2.3), initiated using the sample method, soon proved to be both unstable and unreliable. The most serious disadvantages identified are slow operation time and a tendency to stall in local minima with no ability to escape. This, in turn, led to meaningless cluster-

partitions. The performance was to a certain degree increased through the modified leader algorithm applied as an initiation method which, through a better initial partition, provided VQ with a better chance to find optimal cluster partitions. A semi-stochastic version was developed in a final attempt to make the VQ algorithm more stable and more efficient. This version randomized the order of all samples in a list. The algorithm then ran through the list doing the standard operations for each point. When all samples had been taken into account the new sum of errors was calculated; if the new value was better than that before the operations were continued and accordingly derived changes were kept. If the changes failed to improve the sum of errors value the samples were run again in a new randomized order. The tendency to arrest in local minima was somewhat reduced by this algorithm, but it significantly increased process time. When a sum of errors comparison was made between the semi-stochastic VQ algorithm and K-means the results showed a lower performance for VQ (. Also when compared with K-means on the two data sets described in chapter 3.1 the VQ-algorithm was considerably slower as can be seen in table.

	Sum of errors, $\times 10^9$
K-means	1.84
VQ (leader)	2.01
VQ (sample)	2.12

Table 10 : The resulting mean sum of errors after 25 attempts of clustering the ImmunoCap™ data set into 5 clusters. The semi-stochastic version of VQ was used for this test.

Data set / Clustering algorithm	2-dimensional data	ImmunoCap™ data
K-means	Ca. 0.03 s	Ca. 0.8 s
VQ	Ca. 0.3 s	Ca. 36 s

Table 11 : Time required to group two data sets into 12 clusters using K-means and VQ, both initiated using the sample algorithm.

Because of problems with performance and reliability little gain from a continued work on the vector quantization algorithm was expected. The log-likelihood algorithm seemed initially applicable to matters of this study because of its use of Gaussian probability distribution estimates, which solved the problem with Euclidean distance as mentioned in above chapter. The method showed great promise in initial tests both with respect to speed and accuracy. After further tests, however, it was found that the algorithm did not perform well on the type of operation needed. When the log-likelihood method was applied on the ImmunoCap™ data it became clear that the algorithm seemed to create clusters of close to even size, rather than well grouped clusters

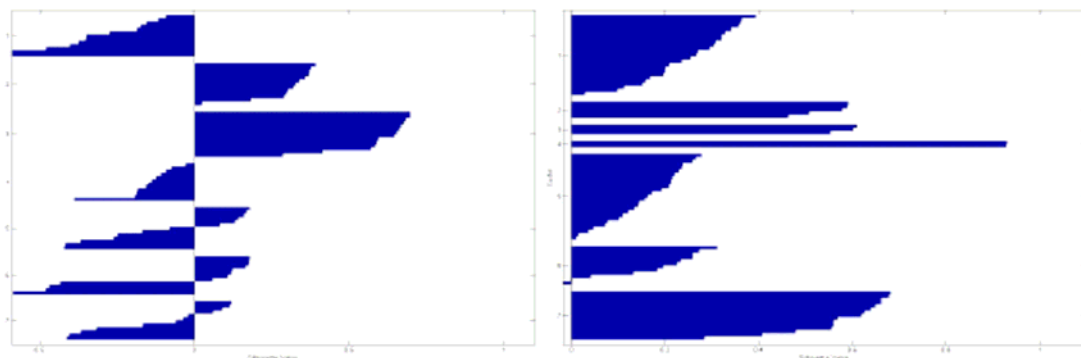


Figure 12: A silhouette comparison of the clustering result from the log-likelihood method (left) and k-means (right).

It is clear that the log-likelihood method favors the formation of clusters of close to similar size, even if it means forming clusters that are poorly grouped. As is the case with e.g. clusters one and four in the log likelihood image.

The application of the Jeffrey-Kullback-Liebler distance to a clustering method would enable maximum separation of Gaussian distributions of clusters. Also, as with the log-likelihood method, the distribution of each cluster would be taken into account. Another advantage lies in that it would reduce the need for a separate cluster validation method, because the clustering result for each k would also return a parameter useful for comparisons. Implementation of an algorithm that used the JKLD was expediently accomplished. It was clear, however, from the initial tests that several hurdles remained. There were doubts as to how a distance measurement between n different clusters ($n > 2$) was supposed to be conducted practically, as the JKLD only is defined as the distance between two Gaussian distributions. In

order to solve this problem the mean distance between all clusters was used, but as mentioned above, this did not work well. The next problem can be seen in figure 4-2, which illustrates the result of a clustering run performed by means of the JKLD clustering method on the two-dimensional data set. It is quite clear that the clustering algorithm failed to achieve a meaningful result.

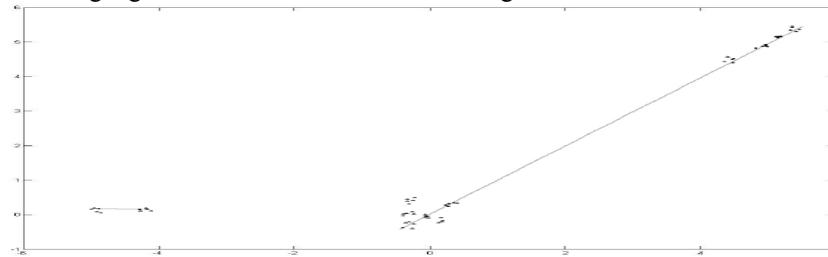


Figure 13 : The three resulting clusters from the JKLD cluster algorithm when applied to the two-dimensional data set.

Here it is clearly visible that two of the clusters (squares and circles) have formed linear structures whereas the third (triangles) has been used as a “leftover” cluster.

Test results

As a final step the iterative hierarchical clustering algorithm was tested on various data in order to see how well the algorithm performed. The first test was to run the algorithm on the ImmunoCap™ data set using the same sample representation and proximity measure as used in the article where the data set was first analyzed [29]. The result of the test run can be seen. The result showed an impressive correlation with results derived from the same data as reported in the above-mentioned article. Actually, the algorithm made even more rational assumptions on individual allergen placement in a few instances. For example, the ‘Bee’- allergen was placed in the group ‘Plant food general’ in the referred study, whereas the iterative hierarchical clustering algorithm placed it in the seafood and insect group, seemingly a more sensible choice. An interesting segmentation of the allergen data can be noted in the first step of the tree-structure. This initial aggregation shows a clear differentiation between the group of allergens containing plants, insects and seafood and the group containing animals and fungi. An inconsistency can be found, though, in the ‘Shrimp’ allergen. The algorithm has placed this allergen within the ‘Animal and fungi’ group rather than in the more logical ‘Seafood and insects’ group.

5. Discussion

Many automated analysis methods in practice today produce very large amounts of data. Most of that data is pre-processed by some type of computer analysis tool prior to presentation to the user. Multivariate data analysis, which is a highly relevant approach to identify meaningful patterns in complex data, typically includes any among a range of clustering algorithms. The most representative reference method for this study is the standard hierarchical clustering algorithm. As also mentioned in previous chapter it is a popular and widely used method to cluster data.

Some advantages of hierarchical clustering: There is no requirement to predefine the number of clusters, different partitions can be achieved by “cutting” the dendrogram at a given level and the dendrogram can in some instances be used as taxonomy are just a few examples.

Limitation of Hierarchical clustering: However, this clustering is limited to joining one pair of clusters at a time, a feature that can easily lead to messy graphical representations, e.g. when a tight group containing many samples is added one by one. From the onset, this study aimed at creating a significantly improved clustering algorithm which combines good qualities inherent in existing methods. More specifically, a key objective was to achieve a hierarchical method of operation that did not limit the number of clusters, formed in each partitioning, to two. The algorithm would require the presence of some kind of optimization algorithm, which would ensure that all groupings formed were optimal, based on current prerequisites. Additionally, an intuitive and informational graphical representation of accordingly produced cluster configurations should be presented to the user. Throughout the development process of the iterative hierarchical clustering algorithm most of the work was focused on initiation-, clustering- and k-selection techniques

5.1. Evaluation of initiation, clustering and k-selection

Initially, the initiation method used for all development was based on the sample algorithm. This was the standard means of initiation for several clustering methods that required a minimum amount of time to operate. It was found, however, that the sample method could lead to poor clustering results, especially if

no repetitions were used. Therefore, two alternative initiation methods, uniform distribution and the modified leader algorithm, were implemented and tested in order to obtain alternatives. Both seemed slightly more stable than the sample algorithm and the leader algorithm was chosen as the preferred option (table 4-1). K-means became the initial choice of clustering algorithm due to the fact that it was already implemented in the “in house” installation of MATLAB™. Alternate clustering algorithms were sought, however, starting with the principally similar VQ-algorithm. Initially the VQ algorithm showed promising results, but it was soon realized that the algorithm was too slow to be appropriate in this context. The slow progress was caused by high numbers of iterations needed to find minima in error curves. Despite many optimization attempts this method could not be sufficiently improved. Two clustering methods based on probability density functions were implemented: maximum likelihood and JKLD. Both these methods, however, posed several problems, which made them unsuitable for this type of application. The most serious problem pertained to calculation of the probability density functions. This required the calculation of the covariance matrix corresponding to each cluster. This, in turn, prohibited data from being singular, i.e. the number of samples ought to be larger than the number of variables. Since the iterative hierarchical clustering algorithm runs from stem to leaf small groups of samples will automatically occur. Hence, in order for the algorithm to work properly the data dimension had to be reduced (by means of principal component analyses), which in turn involves loss of information. In some situations, when the number of dimensions was very large, this reduction led to an appreciably lessened reliance in the resulting partition. Aside from this problem, which affected both methods, each method displayed other weaknesses as well. The maximum likelihood method displayed a preference for even-sized rather than well grouped clusters (figure 4-1). This unfavorable behavior was connected to the fact that the estimation of the Gaussian distribution becomes uncertain when clusters contain a small number of samples. Even though it is principally advantageous (overall likelihood increases) to disrupt large clusters by transferring samples into smaller ones, rather than to keep small clusters un-changed, a consequence is that poorly grouped clusters may be obtained. In the case of JKLD facilitated clustering, one of the main hurdles was that the algorithm failed to achieve meaningful results, but rather produced elongated clusters like those appearing in figure. The problem is that the most efficient way for the algorithm to create clusters with high mutual separation, seems to be to form $k-1$ clusters ($k > 2$) that in the 2-D dataset corresponds to lines and then assign the remainder of the samples to a “leftover” cluster. This causes a significant overlap between the “leftover” cluster and the line-forming clusters and, thus, low JKLD divergence. However, the divergence between different line-forming clusters, as computed by the JKLD, will yield high values, which outweigh the low values obtained between line-forming clusters and the “leftover” cluster. In the end no clustering algorithm managed to replace K-means as a preferred method. K-means was robust for small clusters as well as fast for larger ones. The MATLAB™ version also offered a variety of distance measures, aside from those mentioned in this report. One possible issue is that the MATLAB™ implementation of the K-means algorithm is quite complex, which makes it difficult to modify should a future need arise. The final part of the project algorithm configuration involved a choice of k-selection technique. In the end the silhouette method became the primary method. The ratio method was kept as a valued alternative because it can offer different representations, thereby complementing results derived by the silhouette method. The ratio method is however laden with a problem. In order to obtain a value for $k = 2$ the sum of errors value for $k = 2$ had to be compared with the same value for $k = 1$, i.e. the sum of distances for the entire data set to the mean. Because that value can be very large relative to the $k = 2$ value, the method can, when applied to certain data sets, tend to choose $k = 2$ in most initial steps, see for example figure 5-2. While this is not a major problem, further elaboration on this issue would deviate somewhat from the main pathway of the study. Further development of the alternative to the ratio method, called the L-method, was discontinued because it was clear that it failed to add anything on top of what had already been achieved through the ratio method. After specifying initiation, clustering and k-selection techniques for the algorithmic package the result could be described as follows: an iterative method that hierarchically classifies a dataset from entire group to single point clusters and - at every step - uses K-means in tandem with the silhouette criterion to divide the current data into an optimal number of clusters. The algorithm was dubbed iterative hierarchical K-means (IHKM).

5.2. Application of IHKM to IgE-reactivity data

One of the several purposes with this study was to apply the IHKM algorithm to an allergen data set in an attempt to delineate co-recognition in IgE-mediated allergy, including IgE cross-reactive patterns. A full scale test, using the setup detailed above, gave some very interesting results. Almost consistently these results verified those published in the referred article and, additionally, accomplished a more rational cluster assignment for certain allergen preparations.

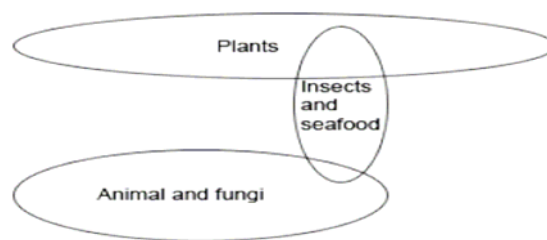


Figure 14: A generalization of the groupings of the ImmunoCap™ data set when the number of dimension has been reduced to two.

Another interesting aggregation contains rice, maize, union and garlic structured as two pairs. Also, when observed groupings among the fungi cluster are more closely studied some correlations between IgE-reactivity and taxonomy can be seen. For example, Baker's Yeast (*Saccharomyces cerevisiae*) is grouped together with *Candida albicans*, which is another yeast species. Moreover *Rhizopus nigricans* and *Mucor racemosus*, which both stem from different lineage than the other fungi, form a separate group. A result that does not quite follow what would be deemed logical is the placement of shrimp, appearing as a single cluster directly under the animal and fungi group. Upon reduction of the dataset to two dimensions only the result attains a highly generalized configuration.

References

- [1] Asharaf, S., Shevade, S. K., and Murty, N. M. (2006). Scalable rough support vector clustering. Technical report, Indian Institute of Science, Bangalore, 560012.
- [2] Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., and Modha, D. (2004a). A generalized Maximum Entropy approach to Bregman co-clustering and matrix approximation. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD), pages 509–514.
- [3] Ban, T. and Abe, S. (2004). Spatially chunking support vector clustering algorithm. In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, volume 1.
- [4] Banerjee, A., Dhillon, I. S., Ghosh, J., Merugu, S., and Modha, D. S. (2007). A generalized Maximum Entropy approach to Bregman co-clustering and matrix approximation. Journal of Machine Learning Research, 8:1919–1986.
- [5] Bertoni, A. and Valentini, G. (2006). Discovering significant structures in clustered data through bernstein inequality. In Conferenza Italiana Sistemi Intelligenti (CISI 06), Ancona, Italy.
- [6] Camastra, F. (2007). A private e-mail conversation about the Kernel Grower clustering algorithm. Camastra, F. and Verri, A. (2005). A novel kernel method for clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5):801–805.
- [7] Carbonell, J. G., Deodhar, M. and Ghosh, J. (2007). A framework for simultaneous co-clustering and learning from complex data. Technical report, Intelligent Data Exploration & Analysis Laboratory, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas 78712.
- [8] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7:179–188.
- [9] Hartigan, J. and Wong, M. (1979). Algorithm as136: A K-means clustering algorithm. Applied Statistics, 28:100–108.
- [10] Modha, D. S. and Spangler, W. S. (2003). Feature weighting in k-means clustering. Machine Learning, 52(3):217–237.
- [11] Slonim, N. and Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In Research and Development in Information Retrieval, pages 208–215.
- [12] Wang, J.-S. and Chiang, J.-C. (2008). A cluster validity measure with a hybrid parameter search method for the support vector clustering algorithm. Pattern Recognition, 41(2):506–520.
- [13] Yang, J., Wang, H., Wang, W., and Yu, P. (2003). Enhanced biclustering on expression data. In Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering (BIBE), pages 321–327.