TECHNISCHE UNIVERSITÄT
CHEMNITZ

# Cluster Algorithms: Theory and Methods

D. Akume, G.-W. Weber

Preprint 2001-9

Fakultät für Mathematik

# Cluster Algorithms: Theory and Methods

D. Akume [*]
Computer Science Department
University of Buea
P.O. Box 63, Buea
Cameroon
daniel.akume@minesup.gov.cm

G-W. Weber [†]
Faculty of Mathematics
Chemnitz University of Technology
Reichenhainer Strasse 41
D-09126 Chemnitz
Germany
weber@mathematik.tu-chemnitz.de

July 19, 2001

**Abstract**

The aim of this work is to study the suitability of some techniques in the clustering of loan banking contracts. We start by presenting two optimization problems that eventually lead to two clustering algorithms - *the single-link* and the *k-means.* We further compare both methods based on certain assessment techniques, space and time complexity, and conclude that the k-means method is more appropriate in view of forecasting customer behaviour.

**Keywords:** *cluster, loan banking, algorithm, graph theory, single-link, K-means, complexity*

## 1 Statement of Problem

In economics, social problems, science and technology, numerous questions of clustering finite sets arise. The present survey article was stimulated by

[*] Supported by DAAD under grant number A/001/31321, at Chemnitz University of Technology.

[†] Visiting professor at Chemnitz University of Technology.

the cluster analysis from loan banking [18], done in cooperation between German "Bausparkassen" and the Center for Applied Computer Science, Cologne (ZAIK). Here, the contracts (accounts) with completely known "saving phase" have to be partitioned for purposes of liquidity planning. Our article gives a small survey and an outlook as well.

A solution to the cluster problem is usually to determine a partitioning that satisfies some optimality criterion. This optimality criterion may be given in terms of a function $f$ which reflects the levels of desirability of the various partitions or groupings. This target is the objective function.

Assuming there are $n$ accounts (objects) and $m$ features for each account we seek to partition these $n$ accounts in $m$ dimensional spaces into meaningful clusters, $K$ in number. The clustering is achieved by minimizing intracluster similarity and maximizing intercluster dissimilarity.
Mathematically, the problem can be formulated as an optimization problem as follows:
For a given or to be suitably chosen $K \in \mathbb{N}$,

$$\min f(C) \tag{1}$$

$$\text{subject to} \quad C = (C_1, \ldots, C_k), \quad C_1 \dot{\cup} \ldots \dot{\cup} C_k = \Pi,$$

whereby $\Pi = \{x_1, \ldots, x_n\}$ is the set of objects to be grouped in $K$ disjoint clusters $C_k$. Finally, $f$ is a nonnegative objective function. Its minimization aims at optimizing the quality of clustering.

## 2 The Method

Generally, it is possible to measure similarity and dissimilarity in a number of ways, such that the quality of the partition will depend on the function $f$ in (1). In this paper, we investigate clustering based on two different choices for $f$.

In our first method,

$$f_{MST}(C) := \max_{x_i, x_j} d(x_i, x_j) - \min_{x_i \in C_v} \min_{x_j \in C_\mu \neq C_v} d(x_i, x_j) \tag{2}$$

The minimization of $f_{MST}$ means the maximization of the minimum distance between any two clusters. Note that the first term on the right hand side does not depend on $C$. The criterion $f_{MST}$ can be interpreted[1] as the "compactness of the clustering". There are efficient (polynomial time) algorithms for this kind of problem.[2]

In our second method,

$$f_{\sum}(C) := \sum_{k=1}^{K} \sum_{i=1}^{n_k} \sqrt{\sum_{j=1}^{m} (x_{ij} - \widehat{z}_{kj})^2} \tag{3}$$

which measures the distance of the objects from the centroids $\widehat{z}_k$ of their respective clusters. Hereby,

$$n_k := \mid C_k \mid, \quad C_k = \{x_{1k}, \dots, x_{n_k k}\} \quad (k = 1, \dots, K)$$

and

$$d(x_i, x_j) := \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2},$$

is the Euclidian metric. The problem of minimizing $f_{\sum}$ has been shown to be $NP$-hard.[3]

A direct way of solving the cluster problem is to evaluate the objective function for each choice of the clustering alternatives and then choose the partition yielding the optimal (minimum) value of the objective function. However, this procedure is not practical even for small values of $n$ and $K$ as the following lemma[4] shows.

**Lemma 1** *The number of ways of partitioning n elements into K groups is determined by Stirling's number of the second kind S:*

[1] See Vannahme, 1996, [18]

[2] See Vannahme, [18].

[3] No one has so far been able to develop any polynomial time decision algorithm for this problem. It has been shown that it corresponds to the hardest problems in the $NP$-class. See [18], page 58: reduction of 3SAT problem. For more information about complexity see Garey and Johnson, 1979, [10].

[4] A proof of this lemma can be found in [2].

$$S(n,K) = \frac{1}{K!}\sum_{i=1}^{K}(-1)^{K-1}\binom{K}{i}(i)^n$$

That is, with $n = 10$ and $K = 4$, there are 34,105 ways of partitioning 10 objects into 4 clusters. This number becomes computationally explosive as $n$ increases, making it impractical to solve for the optimal partition by complete enumeration. The computational running time for such an optimization problem increases exponentially. This leads us use heuristic algorithms, which in many cases will provide only good approximate solutions.

# 3 Clustering Algorithms

### Hierarchical Algorithms

Clustering techniques are referred to as hierarchical if the resultant subdivision has an increasing number of nested clusters. Otherwise, they are non-hierarchical.

Hierarchical techniques can be further classified as either divisive or agglomerative. A divisive (deglomerative) method begins with all objects in one cluster. The cluster is gradually broken down into smaller and smaller clusters.

In an agglomerative hierarchical clustering process one starts with each of the $n$ objects in a single object cluster and groups the two nearest (or most similar) objects into a cluster, thus reducing the number of clusters to $n-1$. The process is repeated until all objects have been grouped into the cluster containing all $n$ objects.

A formal definition of a hierarchical technique for our purposes is presented as follows:

**Definition 1** *Let $\Pi = \{x_1, \ldots, x_n\}$ be a set of $n$ objects. A system $S = (C_1, C_2, \ldots, C_K)$ of subsets of $\Pi$ is called a* hierarchy *of $\Pi$ if all sets $C_1, C_2, \ldots, C_K \subseteq \Pi$ are mutually different and if for any two sets $C_k, C_l \subseteq \Pi$ with $C_k \neq C_l$ only one out of the following three possibilities can occur*

$$C_k \cap C_l = \emptyset \quad or \quad C_k \subset C_l \quad or \quad C_l \subset C_k$$

*The sets in $S = (C_1, C_2, \ldots, C_K)$ are known as the* classes *of $\Pi$.*

Hierarchical techniques are used when the number of clusters is not specified. A serious disadvantage of this technique is that the fusing or dividing process

cannot be reversed.
We shall be concerned in this paper with a hierarchical agglomerative method.

**Partitioning Algorithms**
Clustering techniques are referred to as partitioning if process leads to the object set $\Pi$ being grouped into $K$ clusters. The number of clusters $K$ is predetermined and one starts with an arbitrary start partition into $K$ clusters. The idea is to improve on this partition step by step. The advantage of this method is that objects can move freely from one cluster to another. By so doing, the final partition will be good even if the start partition was poor. The difficulty here is a priori to fix the number of clusters $K$ that would be reasonable. This is a difficult question; the best thing to do is to vary $K$ suitably as a parameter.

## 3.1 The Single-link[5] Hierarchical Clustering Algorithm

The hierarchic agglomerative[6] single-link algorithm which is used to solve optimization problem (1) with the objective function (2), interpreted within the context of graph theory, is the search for a minimum spanning tree[7] (MST) from which edges are deleted in order of decreasing length [1].

The connected sets after deletion are the single-link-clusters. The order of deletion and the structure of the MST ensure that the clusters will be nested into hierarchy.

The objects to be clustered are regarded as the set of vertices $\Pi$ of a graph.

[5] Single-link methods are hierarchical techniques which search for d-clusters in a graph defined by the set $\Pi = \{x_1, \dots, x_n\}$ of objects (vertices).

[6] At each step, an agglometative singe-link algorithm fuses $d$ closest connected components to a new connected component. It starts with each object as a cluster and ends with all objects in one cluster. The distance between two components or clusters is defined as follows: $\text{dist}(C_k, C_l) := \min_{x_i \in C_k, x_j \in C_l} d(x_i, x_j)$.

[7] Given $n$ points in the $m$-dimensional Euclidean space, a tree spanning these nodes (vertices) is a set of edges joining pairs of vertices such that (1.) no cycles occur, (2.) each point is visited by at least one line, (3.) the tree is connected. Here, (3.) means: any two points are connected by a finite sequence of neighbouring edges (i.e., by a "polygon" ). The length of a tree is the sum of the lengths of the edges which make up the tree. The minimum spanning tree is then defined to be the tree of minimum length. These ideas come from graph theory.

These together with the set $E(\Pi)$ of induced edges (i.e., all possible) form a *complete* graph $G = (\Pi, E(\Pi))$. The length of the edge between any two vertices $x_i$ and $x_j$ is the dissimilarity $d(x_i, x_j)$ between both objects.

Let $A = (x_{ij})_{i\in\{1,\dots,n\},j\in\{1,\dots,m\}}$ be an object matrix of size $n \times m$ with $n$ objects each with $m$ attributes[8]. In our specific case of the loan bank ("Bausparkassen") the objects represent accounts and in general, $n$ ranges from $500,000$ to $3,000,000$. A hierarchical clustering method is a procedure for transforming the dissimilarity matrix into a sequence of nested partition [11]. The direct input to the hierarchical clustering is the dissimilarity matrix $D$, which is usually generated from an object matrix $A$. Each entry of the dissimilarity matrix $D = (d_{ik})_{i,k\in\{1,\dots,n\}}$ represents the pairwise indices according to the rows and columns of object matrix $A$. Because the Euclidean distance is the most common Minkowski metric, we use the Euclidean distance to measure the dissimilarity between objects. That is,

$$d_{i,k} = \sqrt{\sum_{k=1}^{m}(x_{ij} - x_{kj})^2}, \quad 0 \leq i, k \leq n$$

The output of a hierarchical custering algorithm can be represented by a dendogram (i.e., a level tree of nested partitions). Each level (denoted as $l_i, 1 \leq i < n$, consists of only one node (different to the regular tree), each representing a cluster. We can cut a dendogram at any level to obtain a clustering.

**Definition 2** *Two vertices of a graph are said to be $d$-*connected *if there exists a sequence of vertices $x_i = x_1, \dots, x_k = x_j$, such that the distance between $x_l$ and $x_{l+1}$, $l \in \{1, \dots, k-1\}$, is always less than $d$.*

**Definition 3** *A subset $C$ of vertices of $\Pi$ is called a $d$-*cluster *in any of the following cases:*

1. *each two vertices from $C$ are $d$-connected,*

2. *vertice $x_i \in C$ being $d$-connected with vertice $x_j$ implies: $x_j$ is also in $C$.*

[8] See [20].

Therefore, a $d$-cluster is a connected component with each edge having length less than $d$. Prim's theorem [16] says

**Theorem 1** *Let $\Pi$ be a set of vertices, $T$ a minimum spanning tree of $G = (\Pi, E(\Pi))$. Moreover, $C_1, \ldots, C_K$ be the clusters we obtain after deleting from $T$ all edges longer than $d$. Then, $C_1, \ldots, C_K$ are all $d$-clusters.*

Therefore, the problem is reduced to one of determining a minimum spanning tree of $\Pi$ and delete all edges longer than $d$.
The problem of finding the minimum spanning tree can be solved by an efficient algorithm which leads to an optimal solution [1].

In the following, we present Prim's algorithm for the single-link technique. The idea is to first obtain the minimum spanning tree and delete the longest edges from it successively.

**Step 1** *Fix an upper bound $r_{max} \in \{i, \ldots, n\}$ for the number of clusters. Put all objects in one cluster.*

**Step 2** *Sort the longest $r_{max}$ edges in decreasing order. Put $t{:=}1$.*

**Step 3** *Delete the longest edge from the tree and place each of the objects of the resultant subtrees in a separate cluster.*

**Step 4** *If $t = r_{max}$: stop, else put $t{:=}t{+}1$ and go to step 3.*

This algorithm requires time complexity of $O(n^2)$ and $O(n^2)$ space complexity to group $n$ objects into $K$ clusters [16].

## 3.2 K-Means

Optimization problem (1) with the objective function (3) is usually referred to as error sum of squares or centroid clustering method.

The centroid method minimizes the objective function $f_{\sum}$ for a given $K$ and partition $C = (C_1, \ldots, C_K)$ of the $n$ objects $\Pi = \{x_1, \ldots, x_n\}$ into $K$ clusters. Here, $\widehat{z}_j$ is the centroid of the cluster $C_j$ $(j = 1, \ldots, K)$, such that

$$f_{\sum}(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \| x_i - \widehat{z}_k \|_2^2 \tag{4}$$

Note that here we have eliminated the square root from (3). The centroid $\widehat{z}_j$ of each cluster $C_j$ is defined as follows:

$$\widehat{z}_{ji} := \frac{1}{\mid C_k \mid} \sum_{l=1}^{|C_k|} x_{li} \quad \text{for} \quad i = 1, \ldots, m$$

This definition of the centroid as a mean value vector of the elements in a cluster *necessarily* guarantees optimality for that single cluster.[9] The problem of obtaining a partition $C$ of all objects in $K$ clusters minimizing (3) has been proven to be $NP$-hard [14]. Therefore we are left with the choice of heuristic algorithms which yield good (not necessarily optimal) solutions in acceptable time.

A general algorithmical partitioning technique for this problem can be stated as follows:

**Phase 1** *Put the number $K$ of clusters equal to a selected integer and choose the maximum number of iterations. Furthermore, choose an initial partition and compute the centroids.*

**Phase 2** *Create a new partition by assigning each object to its nearest centroid.*

**Phase 3** *Compute the new centroids.*

**Phase 4** *Repeat phases 2 and 3 until either the objective function no longer improves or the maximum number of iterations is attained.*

**Phase 5** *Delete empty clusters.*

**Minimum Distance Method**

The process in phases 2 and 3 is referred to as minimum distance method and produces a minimum distance partition.

**Definition 4** *A partition $C = (C_1, \ldots, C_K)$ is called a* minimum distance partition *if each object is assigned to the cluster from whose centroid its Euclidean distance is the shortest:*

$$x_i \in C_j \Leftrightarrow d(x_i, \widehat{z}_j) = \min_{k=1,\ldots,K} d(x_i, \widehat{z}_j)$$

[9] See [17], page 18.

The following theorem [17] implies that clustering due to minimum distance partition is uniquely determined by the cluster centroids.

**Theorem 2** *The minimum distance method produces a separating hyperplane between any two clusters with different centroids of a minimum distance partition.*[10]

In fact, the proof verifies that the hyperplanes

$$\{x \in \mathbb{R}^m \mid \quad \| x - \hat{z}_q \|_2 = \| x - \hat{z}_p \|_2\} \quad (q \neq p)$$

fulfill the desired properties.

**Theorem 3** *The iterated minimum distance method converges.*[11]

Here, we repeat the proof.

**Proof**
Let $C^{(t)}$ be the partition after $t$ iterations. It follow that

$$\begin{aligned} f_{\sum}(C^{(t)}) &= \sum_{k=1}^{K} \sum_{x_i \in C_k^{(t)}} \| x_i - \hat{z}_k \|_2^2 \\ &\geq \sum_{k=1}^{K} \sum_{x_i \in C_k^{(t+1)}} \min_{l=1,\dots,K} \| x_i - \hat{z}_l \|_2^2 \\ &= \sum_{k=1}^{K} \sum_{x_i \in C_k^{(t)}} \| x_i - \hat{z}_k^{(t)} \|_2^2 \\ &\geq \sum_{k=1}^{K} \sum_{x_i \in C_k^{(t)}} \| x_i - \hat{z}_k^{(t+1)} \|_2^2 \\ &= f_{\sum}(C^{(t+1)}) \end{aligned}$$

This algorithm will stop since $f_{\sum}(C)$ falls monotonically and it is bounded below by zero, and the number of ways to group $n$ objects in $K$ clusters is finite. □

[10] For a proof see [17], page 34.

[11] See Spaeth, 1983, [17], page 29.

### The Exchange Procedure

This technique serves to improve the minimum distance method. This technique systematically moves an object to a new cluster. In order to get the effect of this movement on the objective function it is necessary to have updating formulae that indicate the change in the square-error norm and centroid of a cluster $C_p$ when an object $x_i$ is either added or removed from it.

### The Updating Formular when an Object is Added

Let $C_q = C_p \cup \{x_i\}, \quad x_i \notin C_p$. The new centroid for $C_q$ will be

$$\hat{z}_q = \frac{1}{n_q} \sum_{x \in C_q} x = \frac{1}{n_p + 1} \left( \sum_{x \in C_q} x \quad + x_i \right) = \frac{1}{n_p + 1}(n_p \hat{z}_p + x_i) \tag{5}$$

For the error sum of squares norm $f^*_{\sum}(C_q)$ of a cluster $C_q$ we obtain the following formula

$$f^*_{\sum}(C_q) = f^*_{\sum}(C_q) + \frac{n_p}{n_p + 1} \parallel x_i - \hat{z}_k \parallel_2^2 \tag{6}$$

### The Updating Formular when an Object is Removed

Let $C_q = C_p \diagdown \{x_i\}, \quad x_i \in C_q, \quad n_p > 1$. Then

$$\hat{z}_q = \frac{1}{n_p - 1}(n_p \hat{z}_p - x_i) \tag{7}$$

and

$$f^*_{\sum}(C_q) = f^*_{\sum}(C_p) - \frac{n_p}{n_p - 1} \parallel x_i - \hat{z}_k \parallel_2^2 \tag{8}$$

The exchange procedure can now be defined by means of these updating formulas.

- **Step 1** Fix the number of clusters and the maximum number of iterations. Choose an initial partition and and compute the centroids.

- **Step 2** Create a new partition by assigning each object to the closest centroid.

- **Step 3** For each object $x_i \in C_k$, test systematically to find out whether a cluster $C_l$ exists for which the square-error norm improves if $x_i$ is moved into it. That is, if

$$\frac{n_l}{n_l+1} \parallel x_i - \widehat{z}_l \parallel_2^2 < \frac{n_k}{n_k-1} \parallel x_i - \widehat{z}_k \parallel_2^2$$

  occurs, $x_i$ is moved from cluster $C_k$ to cluster $C_l$. If at least one such cluster exists, then move $x_i$ to the cluster that causes maximum improvement in the objective function. Compute the new centroids.

- **Step 4** Repeat steps 2 and 3 until either no other exchanges are necessary or the maximum number of iterations is attained.

**Theorem 4** *The exchange algorithm converges to a minimum distance partition. This partition may not be globally optimal but may be a local minimum.*

The proof of this theorem can be found in [18], page 71.

**Running Time and Space Considerations**
The space complexity increases linearly since only the objects themselves are stored. The running time of the algorithm is $O(cnK)$, with $c$ being an iteration constant that depends on the number of objects, the norm and the structure of the data. Each iteration requires $O(n+K)$ steps to compute the centroids and $O(nK)$ to compute a new minimum distance partition. The switching algorithm also requires $O(nK)$ steps [18].
Based on time and space complexity considerations, therefore, the centroid algorithm is appropriate for handling large data sets.

# 4 The Clustering Process

The clustering results are to be used to carry out simulations of future customer behaviour. The aim is to identify groups of customer accounts that behave similarly within a specific period based on known data from a real loan bank "Bausparkasse". This should enable the forecasting of customer behaviour for a future period.

Let $A = (x_{ij})_{i\in\{1,\ldots,n\},j\in\{1,\ldots,m\}}$, as suggested above, be an object matrix of type $n \times m$ with $n$ objects each with $m$ attributes. In our specific case of a loan bank the objects represent accounts. In general, $n$ ranges from $500,000$ to $3,000,000$.

The relevant account attributes are separated into two groups - the nominal and the rational ones.[12] The entire set of accounts is first of all filtered into subgroups based on the following nominal attributes : *account owner natural or legal entity*[13], *tarif*[14], *tax discounts*[15], *loan advance*[16], *major account*[17] and *phase*[18].

The data is then partitioned using either the centroid or single-link method. We dwell on the following five attributes: nominal amount, savings, saving coefficient, eligibility coefficient and age of customer. Since the attributes are not all of the same units, they have to be scaled in order to be comparable and also normalized.[19]
The following attributes are worthy of explanation:

- **Nominal amount:** amount to be saved by customer plus loan to be obtained from building society (loan bank) as specified in the contract upon opening of account.
- **Savings coefficient:** savings in an account as a fraction of total nominal amount of accounts not yet approved for loan.
- **Eligibility coefficient:** an assessment of the intensity of saving with time $= \int_0^T \text{savings}(t)dt$, $t$ being time.

[12] Vannahme [18], page 80. If data is on a nominal scale, then one can only compare them if they are the same or different. On the other hand, if data is rational, then one can even measure distances between objects.

[13] Legal entities are allowed to own accounts at the loan bank.

[14] See Vannahme, [18], page 133: A tarif describes the conditions underlying each class of accounts, e.g., loan interest etc.

[15] The German government grants incentives to some tarifs.

[16] There is the possibility of obtaining an early loan at an interest rate higher than that guaranteed by the contract at maturity.

[17] Here we mean accounts that at the beginning make a huge one-off saving and thereafter continue saving in very small bits.

[18] The phase considered here is the saving phase. In this phase the customer has to a great extent free decision.

[19] For a more detailed discussion of scaling and normalization of variables see Vannahme [18], pages 83 and 110.

| cluster | number | nominal amount | saving | saving coefficient |
|---|---|---|---|---|
| 1 | 182 | 7 | 27.1 | 116 |
| 2 | 92 | 11 | 11.1 | 366 |
| 3 | 28 | 59 | 8.4 | 152 |
| 4 | 108 | 15 | 8.1 | 209 |
| 5 | 151 | 14 | 10.6 | 154 |
| 6 | 185 | 15 | 5.4 | 24 |
| 7 | 158 | 19 | 8.7 | 95 |
| 8 | 34 | 59 | 5.9 | 35 |
| 9 | 257 | 10 | 14.9 | 55 |
| 10 | 127 | 7 | 40.4 | 189 |

Table 1: Distribution of accounts in clusters, clustered by nominal amount, saving saving coefficient [18]

The dissimilarity measure used is the weighted Euclidean metric

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} \lambda_k \cdot (x_{ik} - x_{jk})^2},$$

whereby $\lambda_k \geq 0, \quad \sum_{k=1}^{m} \lambda_k = 1$.
From Section 3, it is clear that the single-link method is not suited for huge (more than 5,000 objects) data sets. Therefore, we concentrate more efforts on the centroid method.

### Centroid Method

Clustering by the centroid method is carried out on data, initially filtered into subgroups using nominal criteria as indicated above.

The number of objects in each subgroup is greater than 100 on the average (see Table 1). This is necessary for later forecasting if meaningful probability distributions are to be achieved.
This method is implemented as follows: Keep applying the minimum distance method until the error sum of squares norm no longer gets improved. Then apply the exchange procedure. If the exchange procedure causes improvement, apply the minimum distance method again.
For 50,000 accounts the number of minimum distance iterations recorded lies

between 200 and 300. The number of exchange iterations recorded lies between 5 and 10.

Carrying out the centroid method on a SUN SPARC server 1,000 computer with a SPARC processor to group 50,000 objects in 100 clusters took 225.75 minutes [18].

**Single-Link**
The single-link algorithm is particularly suited for identifying geometrically non-elliptical structures in a set of objects. The forecast obtained by implementing the single-link algorithm to cluster loan banking accounts is not very meaningful. Almost all combinations of attributes will contain an account that saves at the regular rate.[20]

## 5 Clustering Assessment Techniques

Due to their diverse mathematical representation, it is extremely difficult to compare clustering algorithms from a purely mathematical perspective. Besides, the suitability of an algorithm will usually depend on the dissimilarity measure and type of data[21], as well as the relevance to the investigation under study. For the purposes of this paper, the relevance is associated with good forecasting.

The following indices[22] have been used to measure the relative suitability of a clustering method as compared to another in this paper .

**Huberts Γ-Index**
The natural structure of the data is represented by the symmetrical dissimilarity matrix

$$D = (d_{ij})_{i,j \in \{1,\ldots,n\}} \quad \text{with} \quad d_{ij} = \| x_i - x_j \|_2 \quad \text{for two objects} \quad x_i, x_j$$

[20] Each tarif fixes a percentage of the nominal amount that should be saved each agreed period.

[21] See Vannahme, [18]: The single-link method for instance easily detects non-elliptical structures in data. For elliptical structures, however, the results are sometimes useless.

[22] See Jobson, [8] and [9].

The structure obtained after clustering is also represented as a symmetrical $n \times n$ matrix defined as follows:

$$\begin{aligned} Y &= (y_{ij})_{i,j \in 1,\ldots,n} \\ \text{with} \quad y_{ij} &= c_{cl(x_i),cl(x_j)} \\ \text{whereby} \quad c_{k,l} &= d(\widehat{z}_k, \widehat{z}_l) \\ \widehat{z}_k &= \frac{1}{n_k} \sum_{x_i \in C_k} x_i \quad \text{and} \\ cl(x_i) &= k, \quad \text{if object } x_i \text{ lies in cluster } k \end{aligned}$$

That is, the distances between cluster centroids associated with the respective objects make up the elements of the matrix. The simple Γ- index takes the following form

$$\Gamma = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X(i,j)Y(i,j) := \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}y_{ij}$$

The larger the value of the Γ-index is, the closer is the clustering, to the structure represented by the matrix $D$.

The normalized Γ-index takes on values between $-1$ and 1 and it is of the following form:

$$\Gamma = \frac{\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(X(i,j) - m_x)(Y(i,j) - m_y)}{s_x s_y}$$

whereby

$$\begin{aligned} M &= \frac{n(n-1)}{2} \\ m_x &= \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X(i,j) \\ m_y &= \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} Y(i,j) \\ S_x^2 &= \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X^2(i,j) - m_x^2 \\ S_y^2 &= \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} Y^2(i,j) - m_y^2 \end{aligned}$$

A disadvantage of the Γ-index is that it requires quadratic running time owing to the fact that the dissimilarity matrix is used.

### Davies-Bouldin Index

This index relates the looseness and the compactness of a cluster. For two clusters $C_j$ and $C_k$ it is defined as follows:

$$R_{j,k} = \frac{e_j + e_k}{dist(C_j, C_k)}$$

Here, $e_j$ is the mean distance from the centroid and $dist(C_j, C_k)$ is the distance between the centroids of cluster $j$ and cluster $k$.
The index of the $k$th cluster is

$$R_k = \max_j(R_{j,k})$$

The Davies-Bouldin index for the entire clustering is

$$DB(K) = \frac{1}{K}\sum_{k=1}^{K} R_k, \qquad K > 1$$

The smaller the index, the better the clustering with regards to its compactness. This is because in this case the cluster diameter is small as compared to the cluster distances from one another. The index value is zero in case the number of clusters equals the number of objects.

### Sum of Squares Norm

The objective function $f_{\sum}$ of the centroid method offers a possibility of comparing clusterings. The sum-of-squares norm measures the compactness of the clusters. The smaller its value, the more compact the clusters are with regards to the attributes.

### Component Index

The component index $F^i$ measures the contribution of each object component (attribute) to the clustering. The sum-of-squares norm is computed for each attribute $i$

$$E^{(i)} = \sum_{k=1}^{K} \sum_{x_j \in C_k} (x_{ji} - \widehat{z}_{ki})^2 \quad \text{for} \quad i = 1, \ldots, m$$

Now, $E^{(i)}$ is used as the index for each attribute $i$ to be considered in an overall index $b^i$ involving all the $k$ clusters. For $i = 1, \ldots, m$ we get

$$b^i = \sum_{k=1}^{K} n_k \cdot \left( (\widehat{z}_i^k)^2 - \widehat{Z}_i^2 \right) = \left( \sum_{k=1}^{K} n_k \cdot (\widehat{z}_i^k)^2 \right) - n \cdot \widehat{Z}_i^2,$$

whereby $\widehat{Z} = (\widehat{Z}_1, \ldots, \widehat{Z}_m)$ is the mean vector of all objects. The weighted quotient

$$F^i = \frac{\frac{b^i}{K-1}}{\frac{E^i}{n-K}} \quad \text{for} \quad i = 1, \ldots, m$$

is an index for the looseness of the individual clusters with regards to an attribute.
The larger the value for $F^i$, the more the attribute $i$ will be similar in the individual clusters.
It is always advisable to assess with as many indices as possible, since the various indices consider differing aspects of clustering.

# 6 Results and Discussion

The single-link algorithm is particularly suited for identifying non-elliptical structures in a set of objects. The forecast obtained by implementing this algorithm to cluster loan banking accounts is not very meaningful. Almost all combinations of attributes will contain an account that saves at the regular rate[23]. The single-link algorithm always isolates these, since it fuses any two clusters having the nearest neighbours. This implies the fact that the algorithm detects customers with very irregular behaviour and places each of them them in a single cluster while the rest of the accounts are placed in a single common cluster.
Upon grouping 150 accounts in 30 clusters one gets about 20 clusters each containing just one or two customer accounts. The rest is in the remaining 10 clusters. Even changing the clustering criteria would not change anything.

The fact that a few clusters are heavily loaded whereas most are almost

[23] Each tarif fixes a percentage of the nominal amount that should be saved in each agreed period.

|  | nominal amount | saving | saving coefficient |
|---|---|---|---|
| cluster 1 | 29.75 | 4.31 | 29.46 |
| cluster 2 | 97.40 | 0.05 | 12.77 |
| cluster 3 | 9.34 | 16.04 | 51.06 |
| cluster 4 | 99.20 | 0.06 | 25.75 |
| cluster 5 | 99.55 | 28.20 | 39.80 |
| cluster 6 | 24.42 | 5.04 | 58.03 |
| cluster 7 | 99.11 | 1.66 | 4.82 |
| cluster 8 | 25.37 | 23.56 | 40.22 |
| cluster 9 | 99.65 | 1.77 | 15.86 |
| cluster 10 | 47.09 | 30.89 | 40.48 |

Table 2: Centroids of the first 10 clusters [18]

empty, does not guarantee meaningful statistical results, for which the clustering is actually intended. Therefore, the single-link method cannot be used to cluster customer accounts of loan banking. This is because the clusters would subsequently have to be used to do forecasts based on probability distributions.

Table 2 presents results obtained from the centroid method for the first 10 clusters.
On the other hand, this algorithm makes use of the dissimilarity matrix or it has to compute object distances at each iteration. The running time increases quadratically as a function of the data size. Therefore, this method is not very useful for the data set currently under investigation.

Results obtained by using these two algorithms can be compared with the help of the indices discussed in Section 5.

Both methods correspond to two different objective functions of the general clustering optimization problem. For comparison purposes 3, 686 customer accounts are considered, each of which is in the saving phase[24] for one year. The clustering criteria were nominal amount, saving and saving coefficient for each contract.

[24] In loan banking, a contract evolves in three main phases: saving phase, assignment phase and loan phase [13].

The Davies-Bouldin index for the single-link method is lower. This is because the Davies-Bouldin index gives higher values for clusters with more objects. The single-link method sorts out the "outliers" in individual clusters, resulting to small numbers of objects in many clusters.
On the other hand, the Davies-Bouldin index measures the looseness of the cluster and obviously obtains low values for the single-link method.
The centroid method distributes the objects relatively uniformly amongst the clusters.

The $\Gamma$-index compares the distance of the objects amongst one another to the distance in-between the cluster centroids. The $\Gamma$-index of the single link method differs greatly from that of the centroid method. This index assesses the single-link method to be worse than the centroid method. This is due to the fact that the single-link method builds rings which do not stand out very clearly on this data set.

The error sum of squares norm for the single-link method is higher than that of the centroid method.

A comparison of the component index of the attributes being used reveals that the centroid method is far better than the single-link method. The component indices of the single-link method are a lot worse than those of the centroid method.

# 7 Conclusion and Outlook

Based on the various indices, one can conclude that the centroid method is suitable for the forecasting. The single-link method can be used to sort out contracts with perculiar properties (outliers) amongst the contracts.

The hierarchical clustering algorithm *single-link* is not suitabe for forecasting for two main reasons.
Firstly, it requires too much space (storage). An attempt to minimize it leads to an equivalent increase in running time. This is caused by the fact that

this method uses the dissimilarity matrix as a basis of its computations. Secondly, the single link method cannot be used to identify clusters in the basis year against the year for which forecasing is sought. This is because clustering is strongly related to the distances between contracts.
The single-link method could, however, be useful for other problem sizes and other objectives. Even the running times for large problems seem to be within acceptable limits.

As two further approaches of evaluating discrete data we mention *automatical classification* (see [4]) and *formal concept analysis* (see [7]).

The authors intend carrying out future research on the following topics:

- application of cluster methods on portfolio optimization (stocks and bonds), life insurances and pensions,
- comparison of exact and heuristic algorithms under the criteria of both quality of the solutions (error bounds) and complexity,
- investigation of traffic on highways by means of cluster methods: finding "typical drivers". This project of ZAIK and momatec company (Aachen, Germany) aims at traffic regulation and, finally, navigation systems.

**Acknowledgements:**
We want to express our gratitude to the Faculty of Mathematics of Chemnitz University of Technology for giving us the opportunity of scientific collaboration, especially Prof. Dr. K. Beer and Prof. Dr. B. Luderer. Furthermore, we thank the Center for Applied Computer science, Cologne, for making the second author familiar with loan banking, and to Prof. Dr. Yu. Shokin and Dr. L. Chubarov and for encouragement and support.

# References

[1] K. Ahuja, L. Magnanti and J. Orlin, *Network Flows*, Prentice Hall, 1993.

[2] S. Aldenderfer and K. Blashfield, *Cluster Ananlysis*, SAGE Publications, 1985.

[3] M. Anderberg, *Cluster Ananlysis for Applications*, Academic Press, 1973.

[4] H. H. Bock, *Automatische Klassifikation*, Vandenbroeck & Ruprecht, 1974.

[5] P. Brucker, *On the complexity of clustering problems*, in: Optimizations and Operations Research, 45-54, Springer, 1974.

[6] S. Duran and L. Odell, *Cluster Analysis - a Survey*, Springer Verlag, Berlin, Heidelberg, New York, 1974.

[7] B. Ganter and R. Wille, *Formale Begriffsanalyse: Mathematische Grundlagen*, Springer, Berlin, Heidelberg, 1996.

[8] J. D. Jobson, *Applied Multivariate Data Analysis, Vol I: Regression and Experimental Design*, Springer, 1992.

[9] J. D. Jobson, *Applied Multivariate Data Analysis, Vol II: Categorical and Multivariate Methods*, Springer, 1992.

[10] M. Garey and D. Johnson, *Computers and Intractability, A Guide to the theory of NP-Completeness*, Freedman and Company, Ney York, 1979.

[11] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Englewood Cliffs, NJ: Prentice-Hall, 1988.

[12] B. Knab, R. Schrader, I. Weber, K. Weinbrecht and B. Wichern *Mesoskopisches Simulationsmodell zur Kollektivfortschreibung*, Center for Applied Computer Science, Report 97.295, 1997.

[13] W. Lehmann, *Die Bausparkassen*, Fritz Knapp Verlag, Frankfurt am Main, 1965.

[14] N. Megiddo and K. Supowit, *On the complexity of some common geometric locations problems*, SIAM Journal on Computing, 13(1), 182-196, 1984.

[15] B. Mirkin, *Mathematical Classification and Clustering*, Kluwer Academic Publications, 1996.

[16] E. Prim, *Shortest connection network and some generalization*, Bell System Technical Journal, 36, 1389-1401, November 1957.

[17] H. Spaeth, *Cluster-Formation und -Analyse*, Oldenbourg Verlag, 1983.

[18] I. M. Vannahme, *Clusteralgorithmen zur mathematischen Simulation von Bausparkollektiven*, doctoral thesis, University of Cologne, Cologne, 1996.

[19] G.-W. Weber, *Mathematische Optimierung in Finanzwirtschaft und Risikomanagement - diskrete, stetige und stochastische Optimierung bei Lebensversicherungen, Bausparverträgen und Portfolios*, lecture held at Chemnitz University of Technology, summer semester 2001.

[20] C. Wu, S. Horng and H. Tsai, *Efficient parallel algorithms for hierarchical clustering on arrays with reconfigurable optical buses*, Journal of Parallel and Distributed Computing 60, 1137-1153, 2000.