Sequentially Grouping Items into Clusters of Unspecified Number

Maytiyanin Komkhao[1], Mario Kubek[2], and Wolfgang A. Halang[2(✉)]

[1] Faculty of Science and Technology,
Rajamangala University of Technology Phra Nakhon, Bangkok, Thailand
maytiyanin.k@rmutp.ac.th
[2] Faculty of Mathematics and Computer Science, Fernuniversität in Hagen,
Hagen, Germany
{mario.kubek,wolfgang.halang}@fernuni-hagen.de

Abstract. When run, most traditional clustering algorithms require the number of clusters sought to be specified beforehand, and all clustered items to be present. These two, for practical applications very serious shortcomings are overcome by a straightforward sequential clustering algorithm. Its most crucial constituent is a distance measure whose suitable choice is discussed. It is shown how sequentially obtained cluster sets can be improved by reclustering, and how items considered as outliers can be removed. The method's feasible applicability to text analysis is shown.

Keywords: Clustering · Number of clusters · Distance measures · Sequential clustering · Single-linkage · Reclustering · Outlier removal · Text analysis

1 Introduction

Clustering is successfully used in exploratory pattern analyses, in data mining, machine learning and in pattern classifications to build concise models of large datasets. The effect of clustering is to group individual items in such a way that the values of their corresponding feature vector components have high similarity to one another within the same cluster, but are rather dissimilar to the components' values in other clusters. An abundance of clustering algorithms has been devised [8], of which the classical and most widely used ones are *k-Means* and, although a classifier by its nature, k-Nearest-Neighbours (*k-NN*).

Most clustering algorithms including k-Means and k-NN require to specify and fix right from the very beginning the number of clusters to be generated for a given dataset. This is too serious a restriction for important application areas such as general recommender systems, because it necessitates visualisation of

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2017*, Advances in Intelligent Systems and Computing 566,
DOI 10.1007/978-3-319-60663-7_28

the underlying datasets and intervention by human experts prohibiting recommender systems to be offered on a continuous basis and automatically operated in an unattended mode. Hence, to adequately build cluster models reflecting the characteristics of given settings, the number of model elements must be adjustable and, thus, employed clustering algorithms are only suitable if they can determine the number of a model's clusters themselves.

Indeed, hierarchical clustering—both agglomerative [11] and divisive—is able to dynamically determine the number of clusters modeling a given dataset. It suffers, however, from another drawback impairing its applicability for many practical purposes, viz. that a set of items to be clustered must be available for processing in its entirety. In contrast to this, the items considered by recommender systems are added one by one, and such systems are expected to be operational all the time and permanently available as web services.

Although incremental clustering algorithms such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), a faster variant of it considering the density of databases [7], or one for information retrieval purposes based on hierarchical agglomeration and considering the maximum cluster diameters regardless the employed distance function [4] do process one item at a time, they adhere to a priori specified numbers of clusters.

To eliminate these two weaknesses of clustering methods, in this paper a heuristic algorithm will be presented, which is able to continuously form cluster models of sequentially arriving items with the number of clusters being adjusted when need be. The algorithm is based on a graph-theoretical and a point-density interpretation of the feature vectors' locations.

Both in proper clustering and in matching feature vectors with the constituents of cluster models, the measures for distance or similarity, respectively, are also quite decisive elements. Since sequential clustering with not a priori prescribed numbers of clusters is the topic of this paper, we do not specify certain distance functions here, but discuss some aspects to be considered in making suitable choices and give a recommendation.

Efficiency and accuracy of modelling also depend highly on how well a model's clusters capture the intrinsic characteristics of the underlying dataset in feature space, and whether this representation is free of redundancies. To this end, outliers may be removed from the input data if the latter are known to be susceptible to noise or errors such as measured values. Therefore, in the sequel it will also be considered how to remove outlying feature vectors and, based on this, how to obtain higher-density and lower-volume clusters.

2 Measuring Distances and Similarity

A plentitude of applications requires to determine the distance of items in feature spaces of any kind. Often distance measures are employed to find item agglomerations (clusters) which are, in turn, used to form classifications of objects and behavioural models of certain phenomena. Items are called most similar, when a distance between them is minimised. When items are similar to a certain

extent, they are often grouped into a common cluster, and dissimilar ones are grouped into different clusters. Employing the metrics induced by the vector norms $\|\cdot\|_m,\ m = 1, 2, \infty$ silently assumes that the component spaces are more or less equal, and that the attributes are totally unrelated.

In most application domains, however, neither the classical metrics are feasible to measure distance nor are the items' attribute spaces similar or, at least, numeric. On the contrary, the components of multi-dimensional feature spaces may be as heterogeneous as continuous set of numbers, discrete sets, Boolean sets, fuzzy sets, structures, graphs, or even continuous functions such as spectra and many others more. Consequently, suitable distance measures can only be selected on the basis of sound knowledge of the particular application domains and of the real semantics of the data [2, p. 26], and utmost care must be exercised when combining different and mutually independent quantities with different physical dimensions in a single arithmetic expression finally giving rise to just a single number. Moreover, in most cases it will be impossible to find an optimal distance measure.

According to the large variety of attribute types and scales, distance measures must be chosen very carefully. Preprocessing may be needed to transform the characteristics of natural phenomena into feature spaces where a notion of distance can be defined. Generally, the physical dimensions of the different attribute spaces should be transformed to similar scales. A distance function $d : F \times F \to \mathbb{R}_+$ on a feature space F must have the properties $d(x, x) = 0$, $x \in F$ and $d(x, y) = d(y, x),\ x, y \in F$, but does not need to be a metric, i.e. the triangular inequality $d(x, y) + d(y, z) \geq d(x, z),\ x, y, z \in F$ is not required to be fulfilled. A suitable distance function does not even have to be continuous.

An empirical study [3] has revealed that for feature spaces with numerical components the Manhattan metric expresses the notion of distance rather well. In comparison to the other metrics based on the vector norms $\|\cdot\|_m,\ m > 1$, for many applications it leads to results of the same or even higher quality, but requires less computational effort. Hence, it is advisable to structure knowledge-based distance functions similar to the Manhattan metric by taking absolute values of two items' attribute differences, but then normalise and multiply them with positive factors to express different weighting of the component spaces in the subsequent summation yielding just one number as distance. For non-numerical attributes, component differences must be defined analogously, e.g. as 1 or 0 when discrete values coincide or not.

3 A Heuristic Algorithm for Sequential Clustering

To form cluster models $\mathbf{M}$ of data points with the number of resulting clusters not set a priori, particularly in application domains where the sets of data points are not available at one time, but the data points are arriving one by one and the cluster models are to be built incrementally, we suggest to employ the following heuristic algorithm, which determines appropriate numbers of clusters itself, i.e. it comprises preclustering as an integral part. The algorithm works sequentially

on a set of data points or feature vectors **F**, either available upon its initialisation or growing by arriving new feature vectors joined with the set. Comparing a feature vector under consideration with known clusters, the algorithm either associates the vector with the cluster matching best, called the winning cluster, already existing in the corresponding model and updates the cluster's parameters accordingly, or it inserts the vector into the model as a new cluster.

Initialisation: Given an input vector f_1, which may be selected randomly in **F** for $|\mathbf{F}| > 1$, let the cluster $\{f_1\}$ form the model **M** initially.
Loop: Execute for any newly arriving or for all further feature vectors $f \in \mathbf{F}$:

1. Calculate the membership of f in all clusters of **M**.
2. Determine the winning cluster as the one for which f assumes the highest membership value.
3. **If** the value of f's membership in the winning cluster does not exceed a given threshold,
   then merge f with the winning cluster,
   else extend the model by a new cluster containing just f ($\mathbf{M} := \mathbf{M} \cup \{f\}$).

The decisive aspect of this algorithm is determining a feature vector's membership in clusters. For this, a distance measure as discussed in the last section will be used. It still remains a design choice to which point in a cluster the distance from a vector is considered. One could, for instance, compare the vector's distances to the centroids of a model's clusters.

Another choice [2, pp. 298–307] is to decide on cluster membership—as in step (3) above—based on the vector's distances to its nearest neighbours in each cluster, respectively, which is called single-linkage method in the literature. Experience revealed that the straightforward and very simple algorithm above works quite satisfactorily with this choice. Clusterings generated are also rather robust with respect to the distance measure and the threshold selected, because not the absolute distance values are crucial, but only their order. The algorithm cannot only recognise circular or ellipsoidal clusters, but also quite differently shaped ones, e.g. with branches, curves or elongate, and two members of a cluster are always connected by a path fully contained in the cluster. The method thus emphasises connectivity of items rather than their similarity.

The last-mentioned property follows from a graph-theoretical interpretation of this kind of clustering. Let a complete graph be formed with the elements of a data set to be clustered as vertices, and the edges between any two vertices weighted by their distance. Removing from the complete graph all edges weighted by distances exceeding a given threshold yields a subgraph whose connectivity components, i.e. the sets of vertices linked by edges contained in the subgraph, are identical with the clusters produced by the algorithm.

4 Outliers and Reclustering

In order to model the intrinsic characteristics of given sets of feature vectors with as low redundancy as possible, model-constituting clusters should be rather

densely filled with data points and should have clear boundaries. Although sequential clustering according to the single-linkage method is able to cope with clusters of a large variety of shapes, it can also lead to the undesirable property of connecting rather dissimilar agglomerations of homogeneous items by chains of intermediately located items and placing them into common clusters. Since sequential clustering is unsuitable to recognise outliers and to form compact clusters all the time, it is advisable to postprocess item sets in an integral way.

An approach to do so is based on a probabilistic interpretation, which considers feature vectors of items as observations of a mixed population constituted by several overlapping populations, the sum of whose single unimodal distribution densities is a multimodal distribution density, i.e. has several local maxima. Under the condition, that the single populations are sufficiently separated, it is assumed, that the local maxima characterise the regions in feature space where the single populations are concentrated, i.e. where clusters are expected.

Based on this interpretation, the method proposed in [12] is able to detect clusters of very complex shapes—just like sequential clustering as discussed above. According to the method those locations in feature space are searched, where a given data set exhibits local point concentrations with higher densities than in the respective vicinities. The search works by iteratively translating with a small step-size all feature vectors towards regions of higher point density. By this process the vectors gradually approach the local maxima. Merging into a single cluster all feature vectors thus arriving in the neighbourhood of a certain location, an exhaustive and disjoint clustering of the data set is produced, with the number of these clusters derived from the characteristics of the data set, but not specified a priori.

Let the set $\mathbf{F} = \{f_1, ..., f_n\}$ of n feature vectors with m dimensions and a distance function $d : F \times F \to \mathbb{R}_+$ be given. With the variance σ the gradient used in the above-mentioned translation of an $f \in \mathbf{F}$ is defined as

$$\nabla f = \frac{1}{(2\pi)^{m/2}\sigma^{m+2}} \sum_{i=1}^{n} (f_i - f) \cdot exp\left[-\frac{d(f, f_i)^2}{2\sigma^2}\right] \tag{1}$$

The scaling parameter σ shapes the Gaussian distributions occurring in this expression. It has to be selected carefully as it determines number and contents of clusters. A clustering appearing "natural" for a certain data set may be sought running the above method [12] for a variety of σ values. When the number of clusters remains constant over a relatively wide range of σ, such a clustering reflecting the data set's intrinsic properties is assumed to be found.

As Gaussian functions are idealised, but computationally demanding approximations of real distributions, and since their values are negligible short distances away from their centres, in the method of [12] we replace Gaussian by other bell-shaped and very similar looking curves, namely B-splines. With

$$(x)_+^k := \begin{cases} x^k, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad k \in \mathbf{N}$$

they are defined by

$$b_k(x) = \frac{1}{k!} \cdot \sum_{i=0}^{k+1} (-1)^i \cdot \binom{k+1}{i} \cdot \left(x + \frac{k+1}{2} - i \right)_+^k, \; x \in \mathbf{R} \tag{2}$$

In general, it will not be necessary to employ B-splines of high degree k, but the bell-shaped ones of lowest degrees will suffice, i.e. the cubic ($k = 3$) or even the just once continuously differentiable quadratic ($k = 2$) B-spline (cp. Fig. 1).

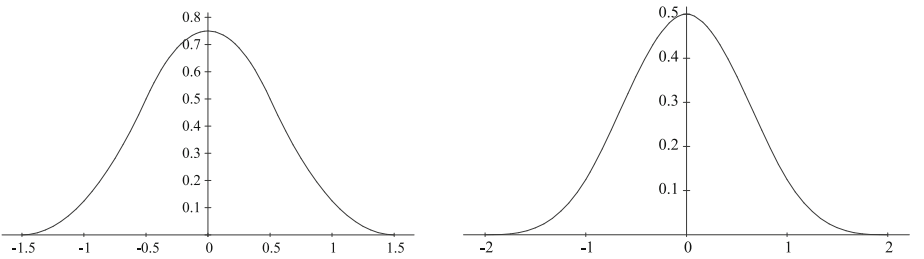

Fig. 1. B-splines of degrees 2 and 3

The method described in [12] lends itself for outlier removal as, after running it, the clusters with low point density are simply removed. This yields a more concise disjoint, but non-exhaustive clustering of the original item set.

Another approach to outlier removal is to eliminate all items from this set in whose neighbourhoods lie only very few, if any, other items. The resulting subset can then be reclustered by the sequential single-linkage method above. The clusters obtained will not contain any dissimilar agglomerations of homogeneous items connected by chains of intermediately located items anymore.

5 Case Study: Clustering Text Documents

The heuristic algorithm presented above can be regarded as a generic solution to group arbitrary kinds of data objects. It performs its task without relying on the number of clusters to be generated as an input parameter, which is usually guessed by an experienced human domain expert. The advantage of not having to estimate this parameter is especially beneficial when information on the heterogeneity or homogeneity of data objects is insufficient. This is particularly true for the domain of automatic text analysis. For instance, a book such as conference proceedings could cover a variety of major and minor topics with each one having its own domain-specific terms. It would be—even depending on the granularity required—hard to estimate the correct number of such topics.

Traditionally, data objects to be clustered are given as vectors with weighted features, making it easily possible to determine their distance or similarity in Euclidean space by applying standard measures such as Euclidean distance or

cosine similarity. In text processing, however, it is also very common to represent them (mostly terms or documents) and their semantic relationships by graphs. For instance, the nodes in so-called co-occurrence graphs usually represent terms, and the (usually undirected) weighted edges indicate semantic relationships between them as well as their significance. Normally, an edge is only drawn when the respective terms co-occur frequently, e.g. on sentence level. In order to determine the significance of co-occurrences using a weight function $g(w_a, w_b)$ for any two co-occurring words w_a and w_b, measures such as the Dice coefficient [5], the Poisson collocation measure [10] or the log-likelihood ratio [6] can be applied. A distance $d(w_a, w_b)$ between w_a and w_b is then defined by

$$d(w_a, w_b) = \frac{1}{g(w_a, w_b)} \tag{3}$$

The distance d of any two nodes (terms) w_a and w_b in a fully connected co-occurrence graph G is obtained by computing the shortest path between them:

$$d(w_a, w_b) = \sum_{i=1}^{k} d(w_i, w_{i+1}) \;\rightarrow\; minimum \tag{4}$$

with $d(w_a, w_b) = \infty$ in case of a partially connected co-occurrence graph.

Unsupervised graph-based clustering approaches such as the Chinese Whispers algorithm [1] can efficiently find useful clusters of semantically connected terms (topics) in co-occurrence graphs. This algorithm relies on a label propagation technique and—just like the algorithm presented in the previous section—does not require a pre-set number of clusters/topics for this purpose. However, in order to achieve the same result on the document level, and taking into account the valuable term relations found in co-occurrence graphs, a new measure to determine the semantic distance between text documents is needed.

By analogy with the physical centre of mass of complex bodies consisting of several single mass points, it was shown in [9] that text documents can be represented by their centroid terms found in a preferably large and topically well-balanced co-occurrence graph G (which acts as reference corpus). In order to determine the centroid term of a document D using G, the distance $d(D, t)$ between a given term $t \in G$ and D containing N words $w_1, w_2, .., w_N \in D$ reachable from t in G must be computed by

$$d(D, t) = \frac{1}{N} \sum_{i=1}^{N} d(w_i, t) \tag{5}$$

Thus, $d(D, t)$ returns the average length of the shortest paths between t and all words $w_i \in D$ that can be reached from it in G. Note that—differing from many methods found in the literature—it is not assumed that $t \in D$ holds. The term $t \in G$ is called the centre term or *centroid term of* D when $d(D, t)$ is minimal. Thus, the semantic distance ζ between any two documents.

$$\zeta(D_1, D_2) = d(t_1, t_2). \tag{6}$$

D_1 and D_2 with their respective centroid terms t_1 and t_2 in G can be expressed as $\zeta(D_1, D_2) = d(t_1, t_2)$. The centroid terms obtained this way generally represent their documents very well. This distance measure is also able to detect a similarity between topically related documents that, however, do not share terms or have only a limited number of terms in common. The cosine similarity measure (when relying on the bag-of-words model) would not be able to accomplish this. Generally, the 25 most frequent words of a medium-sized document, such as a Wikipedia article, are sufficient to properly determine its centroid term. Thus, it is very well possible to determine suitable centroid terms of short documents or even search queries.

As a precondition to successfully apply the sequential clustering algorithm on text documents, it must be examined first whether this new distance measure for documents can actually find pairs of semantically close documents. Also, there are some noteworthy and general remarks to be made when doing so:

1. the data objects to be clustered (the text documents) are represented by only one feature (the centroid term),
2. the distance measure operates on a non-Euclidean space (the Euclidean metric cannot be applied, because the data points are not assigned a coordinate in a multi-dimensional space) and
3. for the co-occurrence graph generated, the triangular inequality does not hold (unequal node distances).

Thus, the aim of the following experiment was to show that most of a reference document's k closest neighbours according to the centroid distance measure share its topical category. The experiment was carried out 100 or 200 times, respectively, for all documents in the following two datasets, whereby each document in these sets was used as reference document. The datasets consist of on-line news articles having appeared between September and November 2015 in the German newspaper "Süddeutsche Zeitung". Dataset 1.1 contains 100 articles covering the topics 'car' (25), 'money' (25), 'politics' (25) and 'sports' (25); dataset 1.2 contains 200 articles on the same topics with 50 documents for each topic. The articles' categories (tags) were manually set by their respective authors. On the basis of these assignments (the documents/articles to be processed act as their own gold-standard for evaluation), it can easily be found out how many of the k nearest neighbours of a reference document according to the centroid distance measure share its topical assignment (needless to say that these topical tags were not considered by the distance measure). The desired result is that this number is as close to $k = 5$ or $k = 10$, respectively, as possible. For this purpose, the fraction of documents with the same topical tags was computed. Furthermore, linguistic preprocessing was applied on the documents to be analysed, whereby stop words were removed and only nouns (in their base form), proper nouns and names were extracted. In order to build the undirected co-occurrence graph G (as reference for the centroid distance measure) using all documents of these two

datasets[1], co-occurrences on the sentence level were extracted. Their significance values were determined using the Dice coefficient [5].

As an interpretation of Table 1, for dataset 1.1 and $k = 5$, on average the centroid distance measure returned 3.9 documents with the reference document's topical assignment first. For $k = 10$, on average 7.6 documents shared the reference document's tag. In both cases the median is even higher. Similar results were obtained for dataset 1.2.

Table 1. Average number of documents sharing the reference documents' category with their $k = 5,\ 10$, respectively, most similar documents

	$k = 5$		$k = 10$	
	Aver. no. of doc.	Median	Aver. no. of doc.	Median
Dataset 1.1	3.9	5	7.6	9
Dataset 1.2	3.9	5	7.5	9

These good values indicate that it is indeed possible to identify semantically close documents with the centroid distance measure. Furthermore, the measure is able to group documents with the same topical tags. Its application in classification systems considering nearest neighbours seems therefore beneficial. The findings further suggest that the centroid distance measure can successfully be applied in document clustering methods, too. Although they represent documents, centroid terms are basically nodes in the co-occurrence graph G used. This means that graph-based clustering algorithms applied on G are inherently able to return both term clusters and document clusters at the same time.

Having said this, the heuristic clustering algorithm presented here—maybe due to its graph-theoretical background outlined above—is well-suited to be used in conjunction with the centroid distance measure, too. In doing so, however, there are two questions remaining to be answered:

1. How can the membership value be calculated?
2. How to set the threshold to assign a document to the cluster with the highest membership value (or to create a new cluster)?

To answer the first question, the membership values for all existing clusters can be computed by determining, in G, the average distance between a document (centroid term) and all centroids existing in a cluster during the algorithm's execution. The threshold's (in the sense of a distance) value, however, is the most important factor to influence the size (number of assigned documents) and the overall number of clusters generated. A high value will likely lead to few large clusters, whereas a low value will cause the generation of many small clusters. In an interactive document clustering solution, this value could be the only input

[1] Interested readers may download these datasets (1.3 MB) from http://www.docanalyser.de/cd-clustering-corpora.zip.

parameter needed from the users. With respect to implementation, one might think of a graphical element such as a slider by which users can easily adjust this value, causing the algorithm to recompute the clusters afterwards.

In fully unsupervised settings, however, this threshold must be determined automatically. For this, several approaches may be sensible. A fixed, semantically motivated, threshold for all centroid terms could be used. In the given graph-based setting, one could speak of a node's 'radius', in which it is likely to find similar documents. Another option is to make this threshold individually dependent, e.g. on the nearest neighbours of a centroid term. The average distance from the nearest neighbours to this term is a good indication for an actual cluster membership. In order to prevent a bias towards only these nearest neighbours, an additional factor should be multiplied with this average distance value to increase the mentioned 'radius' and, in doing so, be able to put more related documents in the same cluster. Future research will investigate these computation options in detail.

Furthermore, when using the given graph-based setting, a model M can initially be filled with two clusters each containing one of the two most distant centroid terms of the so-called antipodean documents in the co-occurrence graph G. Owing to their distance, it is very likely that they are topically unrelated, especially when G is large. For the remaining documents, the cluster assignment and creation can be carried out according to the algorithm presented.

6 Conclusion

Clustering is a means of exploratory pattern analysis and classification aiming to build concise models of large item sets. Most clustering algorithms' property to require the number of clusters sought to be specified beforehand, and all considered items to be present upon clustering, contradicts the exploratory nature of this process and constitutes a serious drawback for many practical applications, which are to operate automatically and continuously. Therefore, it was shown that these shortcomings can be overcome by a heuristic, straightforward and very simple, albeit rather powerful sequential clustering algorithm. After a larger number of items has been collected, it becomes possible to improve cluster models generated sequentially. For this purpose, a feasible algorithm determining an appropriate number of concise clusters by itself, and two approaches for removing items considered as outliers from the models were presented. As measures for distance and similarity are the factors most decisive for the success of clustering algorithms, it was advocated for founding them on domain knowledge and data semantics. As a case study the feasibility of applying a centroid distance measure and the sequential clustering algorithm to find and group semantically similar documents in text analysis was shown.

Acknowledgement. This work was supported by Rajamangala University of Technology Phra Nakhon.

References

1. Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: HLT-NAACL 2006 Workshop on Textgraphs, pp. 73–80. Association for Computational Linguistics, Stroudsburg (2006)
2. Bock, H.H.: Automatische Klassifkation. Vandenhoeck & Ruprecht, Göttingen (1974)
3. Breuer, D.: Abstandsmaße für die multivariate adaptive Einbettung. MSc Thesis, Fernuniversität in Hagen (2014)
4. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. SIAM J. Comput. **33**(6), 1417–1440 (2004)
5. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
6. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Comput. Linguist. **19**(1), 61–74 (1993)
7. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press (1996)
8. Estivill-Castro, V.: Why so many clustering algorithms - a position paper. ACM SIGKDD Explor. Newsl. **4**(1), 65–75 (2002)
9. Kubek, M., Unger, H.: Centroid terms as text representatives. In: ACM Symposium on Document Engineering, pp. 99–102. ACM (2016)
10. Quasthoff, U., Wolff, C.: The Poisson collocation measure and its applications. In: 2nd International Workshop on Computational Approaches to Collocations, Vienna. IEEE (2002)
11. Rasmussen, E.: Clustering algorithms. In: Frakes, W.B., Baeza-Yates, R. (eds.) Information Retrieval: Data structures and Algorithms, pp. 419–442. Prentice-Hall, Upper Saddle River (1992)
12. Schnell, P.: Eine Methode zur Auffindung von Gruppen. Biometrische Zeitschrift **6**, 47–48 (1964)