Journal of Global Optimization **25:** 91–111, 2003.

A Decision Criterion for the Optimal Number of Clusters in Hierarchical Clustering ⋆

YUNJAE JUNG[1], HAESUN PARK[2,3], DING-ZHU DU[2] and BARRY L. DRAKE[4]

[1]*Qwest Communications, 600 Stinson Blvd., Minneapolis, MN 55413, USA (e.mail: yunjae@cs.umn.edu)* [2]*Department of Computer Science and Engineering University of Minnesota, Minneapolis, MN 55455, USA (e-mail: hpark@cs.umn.edu)* [3]*Korea Institute for Advanced Study 207-43 Cheongryangri-dong, Dongdaemun-gu Seoul 130-012, South Korea;* [4]*CDT, Inc., Minneapolis, MN 55454, USA (e-mail: bldrake1@yahoo.com)*

Jan. 21, 2002

Abstract. Clustering has been widely used to partition data into groups so that the degree of association is high among members of the same group and low among members of different groups. Though many effective and efficient clustering algorithms have been developed and deployed, most of them still suffer from the lack of automatic or online decision for optimal number of clusters. In this paper, we define clustering gain as a measure for clustering optimality, which is based on the squared error sum as a clustering algorithm proceeds. When the measure is applied to a hierarchical clustering algorithm, an optimal number of clusters can be found. Our clustering measure shows good performance producing intuitively reasonable clustering configurations in Euclidean space according to the evidence from experimental results. Furthermore, the measure can be utilized to estimate the desired number of clusters for partitional clustering methods as well. Therefore, the clustering gain measure provides a promising technique for achieving a higher level of quality for a wide range of clustering methods.

1. Introduction

Clustering refers to the process of grouping patterns so that the patterns are similar within each group and remote between different groups [1]. The distribution of groups can be defined as a cluster configuration. The cluster configuration is valid if clusters cannot reasonably occur by chance or as a beneficial artifact of a clustering algorithm [2]. An optimal cluster configuration is defined as an outcome of all possible combinations of groupings, which presents a set of the most "meaningful" associations. Even if the definition of clustering is that simple, evaluation of clustering performance is well known as a fundamental but difficult problem. One reason is that clustering should be performed without *a priori* understanding of the internal structure of the data. In addition, it is impossible to determine which distribution of clusters is best given certain input patterns without an objective measure for

⋆ A part of this work was carried out while H. Park was visiting the Korea Institute for Advanced Study, Seoul, Korea, for her sabbatical leave, from September 2001 to July 2002.

clustering optimality. Thus, there have been many attempts to formulate a measure of optimal clustering in the past . However, only a small number of independent clustering criteria can be understood both mathematically and intuitively [2,12]. Consequently, the hundreds of criterion functions proposed in the literature are related and the same criterion appears in several disguises [3–7].

Even though an objective measure is given, the difficulty of optimal clustering stems from the astounding number of possible combinations of cluster configurations [8]. The number of ways of generating k clusters from n patterns is a Stirling number of the second kind [9–11]:

$$\mathcal{S}_n^{(k)} = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{(k-i)}\begin{pmatrix} k \\ i \end{pmatrix} i^n.$$

In particular, the huge volume of data and the potentially high dimensionality of the patterns increase the difficulty of achieving a measure for optimal clustering. Furthermore, it is hard to select a criterion that translates into an intuitive notion of a "cluster" from a reasonable mathematical formula [12]. Feature selection before clustering and cluster labeling after clustering are also challenging problems. As a result, many clustering algorithms to date have been heuristic or ad hoc [2,12].

Since no ideal solution to the optimal clustering problem has existed from early clustering research [13], recently proposed algorithms have focused mostly on efficiency [14–19] and scalability [20–25] to reduce the computational cost and increase processing capability, respectively. It may be possible to produce a cluster configuration very quickly and process huge amounts of data at once. However, often there is no guarantee of achieving an optimal or close-to-optimal clustering configuration.

We propose a method to measure clustering optimality quantitatively with a purpose to use it to determine an optimal number of clusters in various clustering algorithms. The method has been designed based on the assumption that the optimal cluster configuration can be recognized only by the intuitive and subjective interpretation of a human. Since intuitive validation of clustering optimality can be maximized in two dimensional feature space, it is useful to consider two dimensional Euclidean space for the sake of an objective decision as depicted in Figure 1.

In order to quantify clustering optimality, we introduce clustering gain, which has been designed to have a maximum value when intra-cluster similarity is maximized and inter-cluster similarity is minimized [8,26]. Thus, the optimal cluster configuration can be identified by the maximum of the clustering gain curve. This measure can be directly used to explore an optimal configuration for all hierarchical clustering algorithms as they proceed. The measure can also be useful for performance comparison among clustering algorithms since the clustering performance is also measured by clustering gain.

Since discovering all possible combinations of cluster configuration is computationally prohibitive [8,12], most partitional clustering algorithms are dependent on

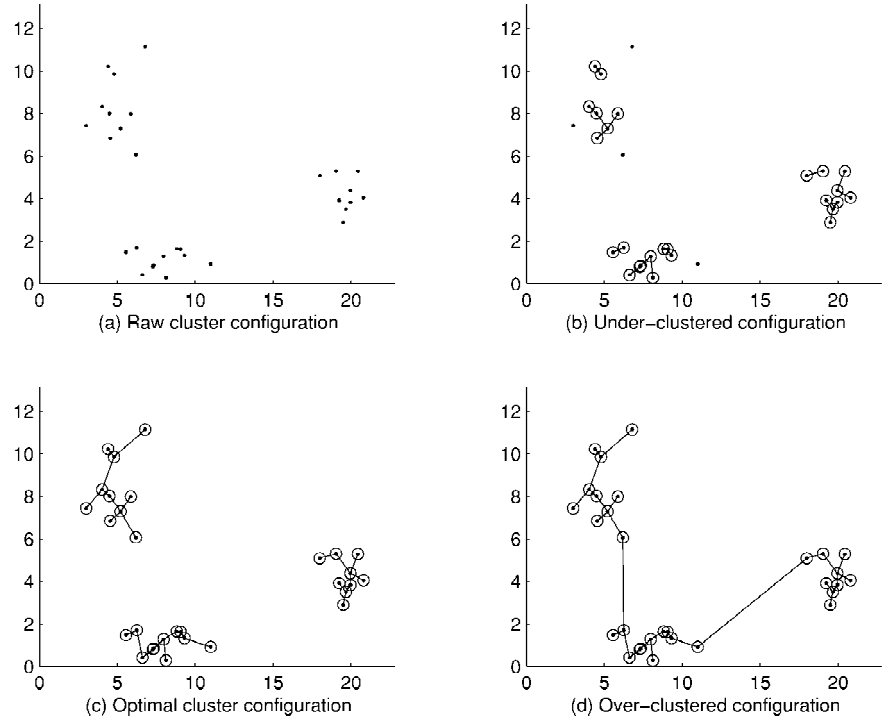

Figure 1. A series of snapshots of clustering configurations in Euclidean distance.

users to determine the target number of clusters. We show how the desired number of clusters can be estimated based on the data using the clustering gain measure. The best cluster configuration will be one which can be produced by any specific hierarchical clustering algorithm. Consequently, the measure is used to estimate the desired number of clusters in partitional clustering algorithms.

According to the experimental results, most commonly used hierarchical clustering algorithms are able to produce intuitively reasonably clustered configurations using our clustering measure when the input patterns are distributed in a well isolated fashion. Moreover, the desired number of clusters for partitional clustering methods, e.g., k-means, has been successfully estimated experimentally.

The rest of this paper is organized as follows. In Section 2, some background information on previous work is presented to derive optimal clustering measures. The design scheme of the optimal clustering gain measure is discussed in Section 3. Section 4 discusses how the proposed method can be used to evaluate the performance of clustering algorithms. Finally, we discuss how to estimate the optimal number of clusters for partitional clustering algorithms using our new scheme in Section 5.

2. Optimal Clustering

Stopping criteria for optimal clustering have been the topic of considerable past research effort [27]. Specifically, deciding the optimal level of a dendrogram and estimating the number of target clusters remains as a challenging and fundamental problem. For hierarchical agglomerative clustering, some decision rules have been provided by Milligan and Cooper [28] to determine the appropriate level of the dendrogram [29,30] for optimal clustering. In addition, Milligan compared and described objective functions for optimal agglomerative clustering. However, the functions are based on strong assumptions, heuristics, and experimental dependency. Recently, a stopping rule for the hierarchical divisive clustering method has been suggested in the Principal Direction Divisive Partitioning (PDDP) algorithm [31]. While the PDDP algorithm is proceeding, a dynamic threshold based on a so-called centroid scatter value is calculated. The rule is to stop partitioning when the centroid scatter value exceeds the maximum cluster scatter value at any particular point. This approach relies on experimental observations. In general, currently used stopping criteria for hierarchical clustering methods are based on predefined thresholds including the number of iterations, the number of clusters, average dissimilarity [32] within a cluster, maximum distance between patterns, and relative inter-connectivity and relative closeness [33].

For non-hierarchical partitional algorithms, Dubes [27] provided a separation index:

$$S(k) = \frac{\mid f(k+1,k) - f(k,k-1) \mid}{1+ \mid f(k+1,k) f(k,k-1) \mid}, \tag{1}$$

where

$$f(k+1,k) = MH(k+1) - MH(k).$$

The value MH is the point serial correlation coefficient between the matrix of Euclidean distances for patterns and a "model" matrix, and k is the number of clusters. The model matrix sets the distance between two patterns to be the distances between the centers of clusters to which the patterns belong. A stopping rule is adopted to search for a significant knee in the curve of $MH(k)$ as k varies from $k_{\max}$ to 2 where $k_{\max}$ is the maximum possible number of clusters.. However, a threshold that distinguishes the knee from other anomalies is difficult to determine. In addition, the rule is not able to avoid premature stopping, i.e., the convergence to local minima problem. Similarly, Boulder and Odell [34] introduced a cluster separation measure

$$R_{ij} \equiv \frac{S_i + S_j}{M_{ij}}, \tag{2}$$

where S_i is a dispersion measure of cluster i such as the squared error sum and M_{ij} is the distance between two centroids. The separation measure will be that which

minimizes the average similarity as follows

$$\overline{R} \equiv \frac{1}{n} \sum_{i=1}^{k} n_i R_i, \tag{3}$$

where $R_i \equiv$ maximum of $R_{ij}, i \neq j$, n_i is the number of paterns in cluster i, and n is the total number of the patterns to be clustered. However, a unique optimal clustering condition can not be detected by the separation measure. According to the experimental results, there are many minimum points in the hierarchical system [34]. Furthermore, there is no theoretical basis for the feasibility of the measure and no reasonable separation measure for partitional clustering seems to exist at present [19].

3. Design of a measure for optimal clustering

3.1. CLUSTERING BALANCE

The clustering problem is to partition the given input patterns into a specific number of groups (clusters) so that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized in a particular metric space [8,26]. Throughout the paper, we will use the following notations. Pattern i is a feature vector in an m dimensional space, denoted as

$$p_i = [p_{i1}, p_{i2}, \ldots, p_{im}]^T,$$

and a cluster C_j is a set of patterns grouped together by a clustering algorithm and expressed by

$$C_j = \{p_1^{(j)}, p_2^{(j)}, \ldots, p_{n_j}^{(j)}\},$$

where n_j is the number of patterns in cluster C_j. We will assume that there are total of n vectors to be clustered and the total number of the clusters is k. Accordingly, $\sum_{i=1}^{k} n_i = n$. In addition, $p_0^{(j)}$ denotes the centroid of the cluster j, which is defined as

$$p_0^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} p_i^{(j)}.$$

The centroid is a mean vector of the cluster and provides a compressed representation of the cluster in a *simpler* form. The centroid is often used for cluster data compression.

Cluster configuration is a random variable whose possible outcome is a particular assignment of input pattern sets. The problem of optimal clustering is to find a cluster configuration that is optimized according to some evaluation criterion. However, as mentioned before, the number of ways of clustering n observations

into k groups is enormously large [35]. In fact, a combinatorial search of the set of possible configurations for optimal clustering is clearly computationally prohibitive [27] and, is NP-complete [36,37]. Accordingly, currently used agglomerative clustering algorithms take an approximation approach by merging more similar patterns prior to grouping less similar patterns to construct a cluster hierarchy. A measure of the similarity between two patterns drawn from the same feature space plays an essential role in these clustering procedures [12,35].

The most popular metric for measuring similarity between patterns is the Euclidean distance since it is more intuitive and applicable, especially in two dimensional feature space [2,13]. The most intuitive and frequently used criterion function in clustering techniques is the squared error criterion which is the sum of squared distances from the centroid of a group to every pattern in the group [13,14,19,38–43] which can be expressed using the Euclidean distance [2,44].

The intra-cluster error sum Λ is defined by the squared error e as

$$\Lambda = \sum_{j=1}^{k} \sum_{i=1}^{n_j} e(p_i^{(j)}, p_0^{(j)}).$$

which, using the Euclidean distance, can be denoted as

$$\Lambda = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \| p_i^{(j)} - p_0^{(j)} \|_2^2. \tag{4}$$

It is also called the within-group error sum [12]. Ward used the error sum of squared errors to quantify the loss of information by grouping [38].

The inter-cluster error sum takes into account error sums between clusters by considering the collection of cluster centroids to be a global pattern set, which also has a global centroid. The inter-cluster error sum, in case of Euclidean space is defined as

$$\Gamma = \sum_{j=1}^{k} e(p_0^{(j)}, p_0) = \sum_{j=1}^{k} \| p_0^{(j)} - p_0 \|_2^2, \tag{5}$$

where p_0 is the global centroid defined as

$$p_0 = \frac{1}{n} \sum_{i=1}^{n} p_i.$$

Now, we present some characteristics of these two conflicting error sums, to be utilized in designing a measure for optimal cluster configuration as well as a stopping criterion in hierarchical clustering algorithm. We will assume that the hierarchical algorithm we are consiering is agglomerative. In case of divisive algorithms, an analogous but opposite trend can easily be proved. We can assume

that in the initial state of any agglomerative clustering algorithm, each pattern is the only pattern in its own cluster. It is clear that the singleton clusters have no contribution to the intra-cluster error sum Λ, and the minimum value that Λ can take is zero. On the other hand, Λ is maximized when there is only one cluster that contains all patterns. More interesting fact is that while clustering process proceeds, the value of Λ cannot decrease. Suppose two clusters C_i and C_j are merged in a step of agglomerative clustering. Let the new cluster C_{ij} be the cluster obtained by mering C_i and C_j. Then, the centroid c_{ij} of the new cluster C_{ij} is

$$c_{ij} = \frac{n_i p_0^{(i)} + n_j p_0^{(j)}}{n_i + n_j}.$$

Let Λ_b and Λ_a be intra-cluster error sums of the items that belong to the clusters C_i and C_j only, before and after merging, respectively. Then

$$\Lambda_b = \sum_{l=1}^{n_i} \| p_l^{(i)} - p_0^{(i)} \|_2^2 + \sum_{l=1}^{n_j} \| p_l^{(j)} - p_0^{(j)} \|_2^2,$$

and

$$\Lambda_a = \sum_{l=1}^{n_i} \| p_l^{(i)} - c_{ij} \|_2^2 + \sum_{l=1}^{n_j} \| p_l^{(j)} - c_{ij} \|_2^2.$$

Since, there is no split of a cluster in a path of agglomerative clustering, intra-cluster error sum would be nondecreasing as the clustering proceeds if $\Lambda_a - \Lambda_b \geqslant 0$. We have

$$\begin{aligned}
\Lambda_a - \Lambda_b \;=\; & \sum_{l=1}^{n_i} \| p_l^{(i)} \|_2^2 - 2c_{ij}^T \sum_{l=1}^{n_i} p_l^{(i)} + n_i c_{ij}^T c_{ij} + \sum_{l=1}^{n_j} \| p_l^{(j)} \|_2^2 \\
& -2c_{ij}^T \sum_{l=1}^{n_j} p_l^{(j)} + n_j c_{ij}^T c_{ij} \\
& -[\sum_{l=1}^{n_i} \| p_l^{(i)} \|_2^2 - 2(p_0^{(i)})^T \sum_{l=1}^{n_i} p_l^{(i)} + n_i \| p_0^{(i)} \|_2^2 + \sum_{l=1}^{n_j} \| p_l^{(j)} \|_2^2 \\
& -2(p_0^{(j)})^T \sum_{l=1}^{n_j} p_l^{(j)} + n_j \| p_0^{(j)} \|_2^2].
\end{aligned}$$

Using $\sum_{l=1}^{n_i} p_l^{(i)} = n_i p_0^{(i)}$ and $\sum_{l=1}^{n_j} p_l^{(j)} = n_j p_0^{(j)}$, we have the desired result

$$\begin{aligned}
\Lambda_a - \Lambda_b \;=\; & 2n_i \| p_0^{(i)} \|_2^2 - n_i \| p_0^{(i)} \|_2^2 - 2n_i (p_0^{(i)})^T c_{ij} + n_i \| c_{ij} \|_2^2 \\
& +2n_j \| p_0^{(j)} \|_2^2 - n_j \| p_0^{(j)} \|_2^2 - 2n_j (p_0^{(j)})^T c_{ij} + n_j \| c_{ij} \|_2^2 \\
=\; & n_i \| p_0^{(i)} - c_{ij} \|_2^2 + n_j \| p_0^{(j)} - c_{ij} \|_2^2 \geqslant 0.
\end{aligned}$$

Similarly, the inter-cluster error sum satisfies the following characteristics which show the opposite trend to that of the intra-cluster error sum. Note that the global centroid p_0 does not change throughout the clustering process. The inter-cluster error-sum Γ is maximized when there are n singleton clusters, which occurs at the beginning of clustering. Then Γ is minimized when all n patterns belong to one cluster at the end of clustering. It is easy to show that the value of Γ is nonincreasing as the clustering proceeds using the triangular property of the L_2 norm, the Euclidean distance.

Our design scheme is based on the fact that intra-cluster similarity is nondecreasing and inter-cluster error sum is nonincreasing as the agglomerative clustering algorithm proceeds. When the clustering algorithm is divisive, the trend is the other way around, which is that intra-cluster error sum is nonincreasing and inter-cluster similarity is nondecreasing as the divisive clustering algorithm proceeds.

We transformed the optimal clustering problem into a problem for finding the point where the two similarities are balanced by representing these similarities by the squared error sum in Euclidean space. We define the *clustering balance* as

$$\mathcal{E}(\chi) = \alpha\Lambda + (1-\alpha)\Gamma, \tag{6}$$

where Λ and Γ denote intra-cluster and inter-cluster error sums for a specific clustering configuration χ, respectively, and $0 \leqslant \alpha \leqslant 1$ is a scalar that determines the weight between these two sums. The clustering balance $\mathcal{E}(\chi)$ has been formulated with the idea that intuitively optimal clustering is achieved when the error sums have reached equilibrium. We will concentrate on the special case for $\alpha = 1/2$ which provides an even balance and accordingly assume that

$$\mathcal{E}(\chi) = \Lambda + \Gamma. \tag{7}$$

Thus clustering behavior can be interpreted as a procedure seeking the global minimum of clustering balance. With the definitions of clustering balance based on the error sums, what follows will use the trade-off between inter-cluster and intra-cluster balance to define a measure for the optimal clustering configuration.

3.2. CLUSTERING GAIN

The clustering balance can be computed in each step of a hierarchical clustering algorithm to determine the optimal number of clusters. However, a major disadvantage is the high computational cost of computing clustering balance. In this section, we introduce *clustering gain* which has an interesting relation to cluster balance. In addition, clustering gain is cheap to compute. Therefore, it can be computed in each step of clustering process to determine the optimal number of clusters without increasing the computational complexity.

Clustering gain Δ_j for C_j is defined as the difference between the decreased inter-cluster error sum γ_j compared to the initial stage and the increased intra-

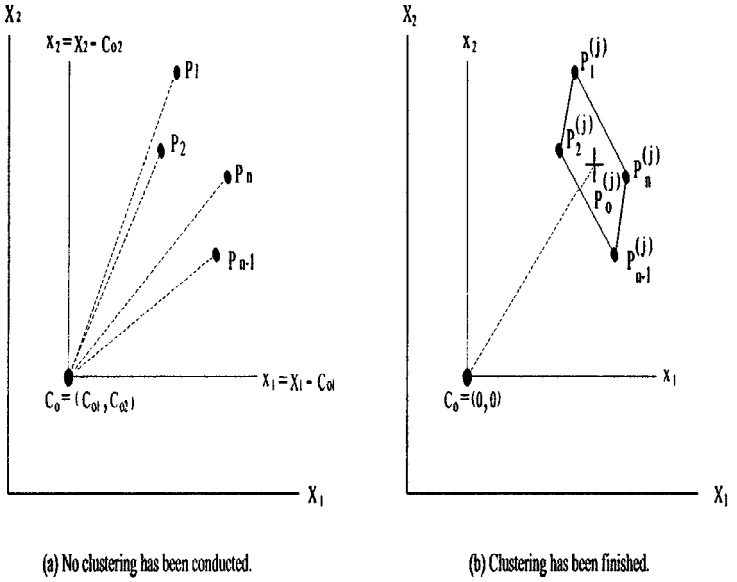

Figure 2. Clustering gain defined by difference between error sums. (a) Initial configuration. (b) Final configuration of cluster C_j.

cluster error sum λ_j compared to the initial stage. Specifically, the gain is defined as

$$\Delta_j = \gamma_j - \lambda_j.$$

In the above equation, an equal weighting factor, one, has been assigned to both error sums. Clustering gain is graphically illustrated in Figure 2 using cluster configurations. In particular, the number of patterns of the final configuration of cluster C_j can vary from 1 to n.

The decreased portion of the inter-cluster error sum compared to the initial stage is denoted by

$$\begin{aligned}\gamma_j &= \sum_{i=1}^{n_j} e(p_i^{(j)}, p_0) - e(p_0^{(j)}, p_0) \\ &= \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0\|_2^2 - \|p_0^{(j)} - p_0\|_2^2\end{aligned}$$

In addition, the increased portion of the intra-cluster error sum compared to the initial stage is defined by

$$\lambda_j = \sum_{i=1}^{n_j} e(p_i^{(j)}, p_0^{(j)}) = \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0^{(j)}\|_2^2. \tag{8}$$

Expanding the gain for cluster C_j gives

$$\begin{aligned}\Delta_j &= \gamma_j - \lambda_j \\ &= \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0\|_2^2 - \|p_0^{(j)} - p_0\|_2^2 - \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0^{(j)}\|_2^2 \\ &= (n_j - 1)\|p_0 - p_0^{(j)}\|_2^2\end{aligned}$$

since $\sum_{i=1}^{n_j} p_i^{(j)} = p_0^{(j)} n_j$. Therefore, the total clustering gain can be computed from

$$\Delta = \sum_{j=1}^{k} (n_j - 1)\|p_0 - p_0^{(j)}\|_2^2. \tag{9}$$

We would like to emphasize that this clustering gain is very cheap to compute since it involves only the centroids and the global centroid, and not the individual data itme. The clustering gain Δ_j is always greater than or equal to zero. Eventually, the clustering gain will be positive, assuming the initial clustering configuration is not optimal.

Apparent from Figure 3 is the fact that the optimal clustering configuration discovered by a hierarchical clustering algorithm has maximum clustering gain. Since clustering gain is minimum at the initial and final clustering stages, an optimal configuration should be found during the middle of the clustering procedure. In order to determine the maximum clustering gain during the middle of the clustering procedure, we propose the *clustering gain* as an effectiveness criterion. Note that clustering gain is analogous to the E value suggested by Jardine and Rijsbergen [45], for clustering effectiveness.

It is interesting to note that the sum of clustering balance and clustering gain is a constant for a given data set since

$$\begin{aligned}\Omega &= \mathcal{E} + \Delta = \Lambda + \Gamma + \Delta \\ &= \sum_{j=1}^{k} \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0^{(j)}\|_2^2 + \sum_{j=1}^{k} \|p_0^{(j)} - p_0\|_2^2] \\ &+ \sum_{j=1}^{k} (\sum_{i=1}^{n_j} \|p_i^{(j)} - p_0\|_2^2 - \|p_0^{(j)} - p_0\|_2^2 - \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0^{(j)}\|_2^2) \\ &= \sum_{j=1}^{k} \sum_{i=1}^{n_j} \|p_i^{(j)} - p_0\|_2^2\end{aligned}$$

which is determined completely based on the data, and not changed by the clustering result. Thus clustering balance can be alternatively expressed using clustering gain as

$$\mathcal{E} = \Lambda + \Gamma = \Omega - \Delta,$$

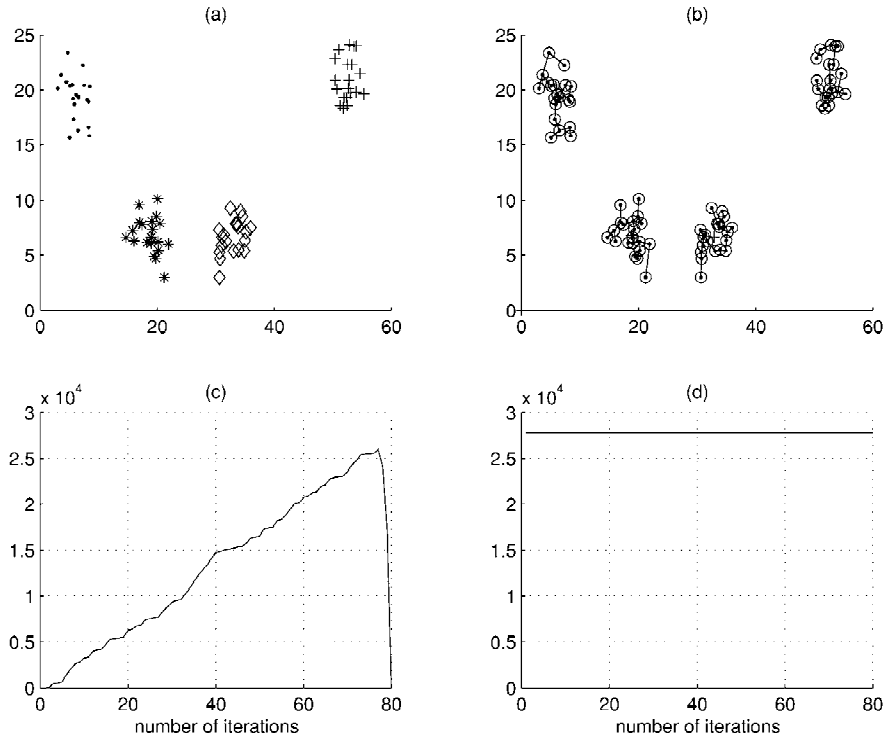

Figure 3. Clustering gain: the opposite concept of clustering balance. (a) Initial configuration of patterns. (b) Intuitively well clustered configuration captured when clustering gain is maximized. (c) Clustering gain (d) The sum of clustering balance and clustering gain.

where $0 \leqslant \Lambda, \Gamma, \Delta \leqslant \Omega$. Now, we are able to find an optimal cluster configuration by tracing clustering gain instead of clustering balance. For visual demonstration, clustering gain Δ and the constant Ω are compared in Figure 3.

In Tracking Algorithm, we summarize how we can obtain the optimal cluster configuration in a given hierarchical agglomerative clustering algorithm while keeping track of the cluster gain value. Note that we need to keep track of the clustering gain $\Delta(\chi)$ since the global maximum value of clustering gain can be discovered only after the clustering is completed.

Application of this method to hierarchical divisive clustering method is straightforward. To demonstrate the performance of Tracking Algorithm, the optimal configuration detected by the complete-link is visualized in Figure 4. Clearly, the configuration containing the lowest value of clustering balance coincides with the optimal configuration produced by a human. Given the same input patterns, the same optimal configuration has been obtained by popular agglomerative clustering algorithms including the single-link, the average-link and Ward's method.

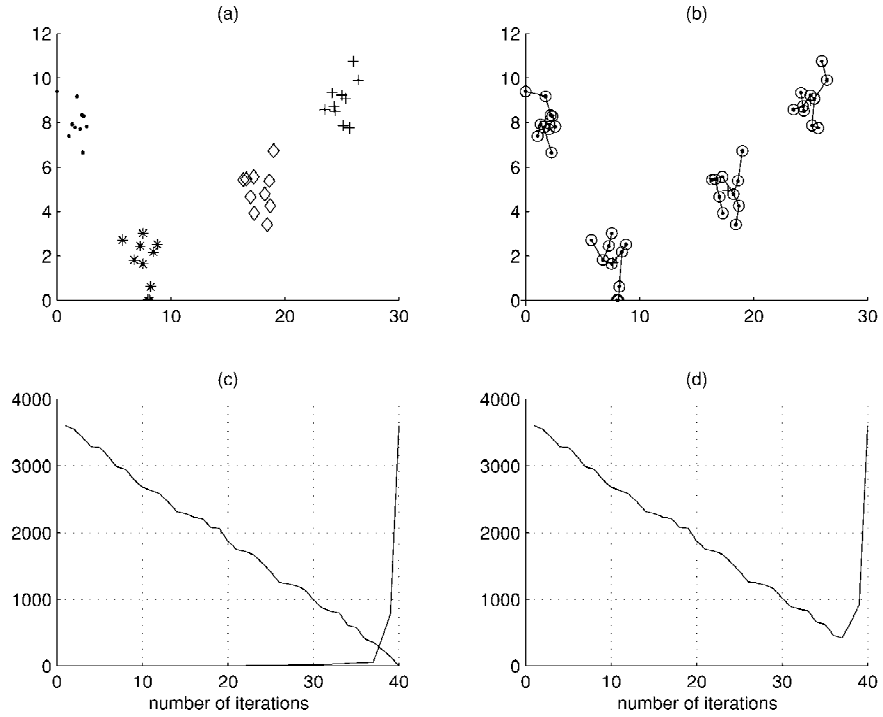

Figure 4. Visual demonstration of optimal configuration discovered by Tracking Algorithm using the complete-link. (a) Initial configuration. (b) Optimal configuration. (c) Intra-cluster versus inter-cluster error sum. (d) Clustering balance.

Tracking Algorithm:

```
1  Choose a hierarchical agglomerative clustering method(HACM)
2  Do while clustering is not complete in HACM
4     merge two clusters according to
      the fusion rule of the HACM
5     keep track of maximum value for Δ(χ) and save χ
6  end while
7  recover the optimal configuration χ
```

4. Performance Evaluation of Clustering Algorithms

Given a hierarchical clustering algorithm, either clustering balance or clustering gain can be used to find optimal configuration. Since these measures represent clustering optimality in an absolute value, they also can be used to compare various clustering algorithms in terms of clustering performance. To give an example we

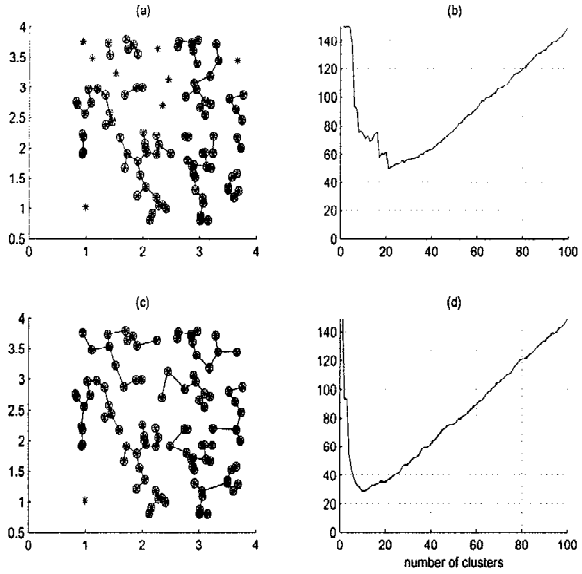

Figure 5. Comparison of the single-link and the average-link. (a) Optimal configuration found by single-link. (b) Clustering balance versus the number of clusters in single-link. (c) Optimal configuration found by average-link. (d) Clustering balance versus the number of clusters in average-link.

estimated the practical distribution in high dimensionality using the method proposed by Bennett [46]. The method is based on the observation that the variance of the distance between patterns chosen randomly in a hyper-sphere is inversely proportional to the dimensionality m as follows.

$$m \text{ x variance(a pair of patterns)} \simeq \text{constant}.$$

According to the equation, the higher the dimensionality is the smaller the variance is. Thus clustering in high dimensional space can be simulated in two dimensional space if patterns are randomly and uniformly distributed. Typically used agglomerative clustering algorithms have been applied to Tracking Algorithm, and then their optimal configurations and clustering balances are represented in Figure 5 and Figure 6. According to the results, the complete-link produces the configuration with the lowest clustering balance. In the experiment, the complete-link outperformed other three clustering algorithms since it produces the best configuration given the same input patterns.

To extend our approach to practical domain, we conducted a simple experiment with practical document vectors. The documents have been downloaded from MEDLINE on-line library. They are divided into eight cancer categories including breast, colon, weightloss, glycolic, heart attack, oral, prostate and tooth-decay. Each category contains 500 documents. After filtering the document set using the stoplist of SMART IR system and the stemming algorithm proposed by Porter [48],

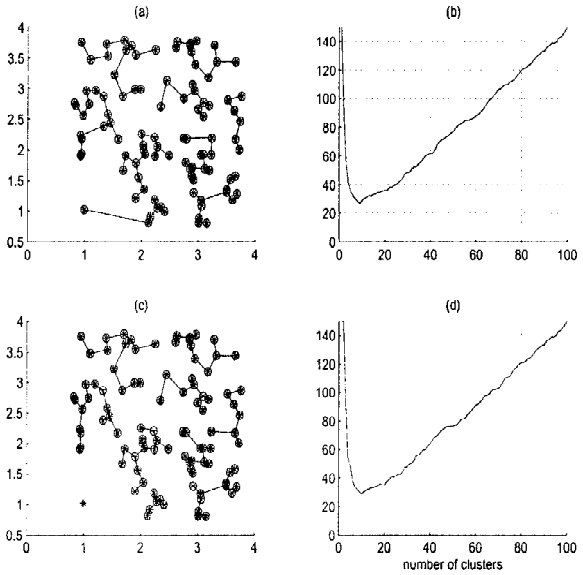

Figure 6. Comparison of the complete-link and Ward's method. (a) Optimal configuration found by the complete-link. (b) Clustering balance versus the number of clusters in the complete-link. (c) Optimal configuration found by Ward's method. (d) Clustering balance versus the number of clusters in Ward's method.

we applied Tracking Algorithm to the combination of Colon and Tooth categories. The results in Euclidean space are graphically illustrated in Figure 7. According to the results, optimal cluster configurations can be found by our measure in Euclidean space.

5. Estimation of the Optimal Number of Clusters for Partitional Clustering Algorithms

A major problem accompanying the use of a partitional clustering algorithm is the choice of the number of desired output clusters [27] and order dependency [13]. The sensitivity to the selection of the initial partition is a problem in most partitional algorithms [2] since the partitional clustering algorithms may converge to a local minimum of the criterion function if the initial partition is not properly chosen [2]. For handling large data set and avoiding such sensitivity, many efficient algorithms have been proposed including CLARANS [19], BIRCH [47], CURE [23], and ROCK [24]. Many clustering algorithms have efficiency and capacity, but most partitional clustering algorithm depend on users to determine the desired number of clusters.

Even though exhaustive enumeration of all possible assignments is not computationally feasible even for small numbers of patterns [12], we can generate all possible cases with very small number of patterns such as ten. In Figure 8, optimal

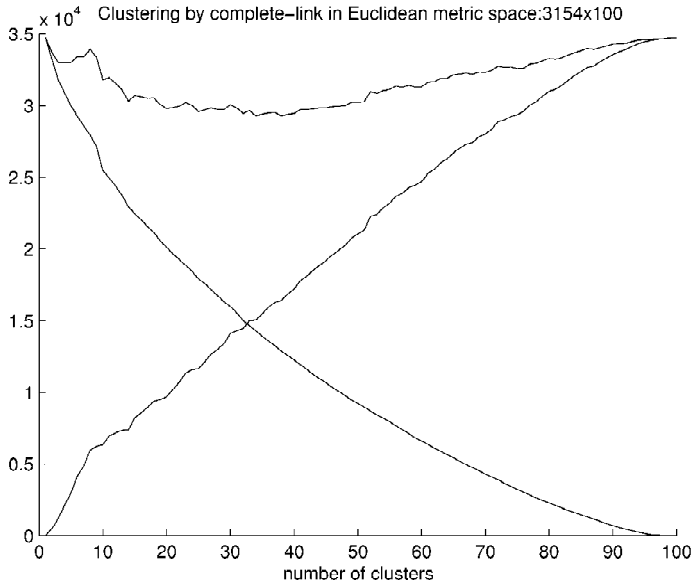

Figure 7. Clustering balance, intra-cluster and inter-cluster squared error sums in Euclidean space.

configurations and their clustering balances are compared with respect to ideal configuration. For each possible number of clusters, the lowest balance is presented in the part (a) of the figure. According to the experimental results, the complete-link discovered the optimal configuration as closely as the ideal. However, it is risky to generalize this result so that the complete-link is superior to all other algorithms for all input patterns since clustering performance may change in accordance with the distribution of the input patterns. A hierarchical clustering algorithm is considered as the best if it produces the lowest clustering balance given particular data patterns.

As we previously described, the best configuration can be selected among optimal configurations produced by hierarchical clustering algorithms. Consequently, the desired number of clusters can be estimated from the best configuration. Also the centroids of the best configuration can be fed to partitional clustering algorithms to avoid random initial assignments for centroids. The basic assumption of this approach is that the best cluster configuration, the winner among configurations produced by hierarchical clustering algorithms, will be an approximation of the ideal cluster configuration for partitional clustering.

It is clear that the estimated number is not the true value. However, the estimated number can contribute to decision of the range of the true number of optimal clusters. To verify this assumption experimentally, we applied k-means algorithm with all possible number of clusters. The averaged clustering balance produced by the k-means is depicted in Figure 9 along with the number of clusters. In this

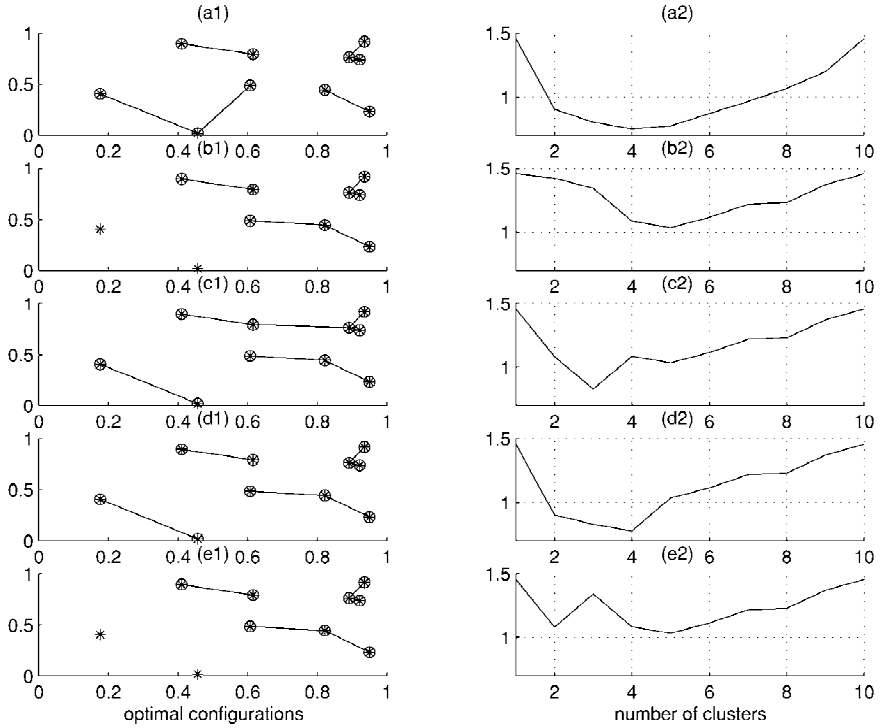

Figure 8. Comparison of currently used agglomerative clustering algorithms to the optimal clustering by using cluster configurations and clustering balances. (a) The optimal (b) The single-link (c) The average-link (d) The complete-link (e) Ward's method.

Table 1. Comparison of lowest balances

Algorithms	The lowest balance	The highest gain	Number of clusters
Single-link	49.8603	100.139651	21
Average-link	29.2123	120.787653	10
Complete-link	27.0904	122.909583	9
Ward's Method	29.2292	120.770844	10

experiment, the clustering balance is the average of five trials. According to the experimental results, the desired number of clusters for the given distribution is nine. When we apply Tracking Algorithm to four popular algorithms including single-link, average-link, complete-link and Ward's method, corresponding optimal configurations are found as in Table 1. Surprisingly, the number of clusters produced by the complete-link is equivalent to the desired number of clusters ob-

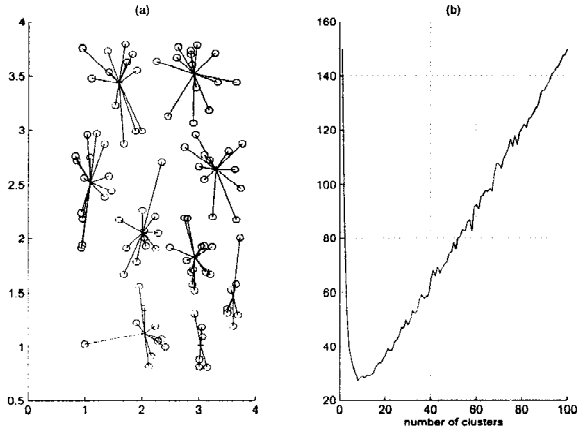

Figure 9. The optimal configuration and clustering balance traced by k-means. (a) Optimal configuration. (b) Averaged clustering balance of all possible number of clusters using k-means.

tained by k-means using all possible k values. This result convincingly illustrates that our clustering measure can be used for partitional algorithms to estimate the number of desired clusters. In addition, more stable configuration and improved performance are demonstrated in Figure 10. When the number of desired clusters and initial centroids are estimated, k-means is able to converge quickly.

For k-medoid algorithm, the results are almost the same as k-means algorithm except some fluctuations of clustering balance before convergence. This result is normal since centroids and medoids are located differently in the same feature space. As a result, the best cluster configuration found by hierarchical clustering algorithms contributes to determining the desired number of clusters and the initial centroids for partitional clustering algorithms.

6. Conclusion

Clustering is not a new problem in computer related disciplines. However, a huge demand for clustering technique represented by a variety of clustering applications demonstrates its importance. Recently, much effort has been presented to achieve clustering efficiency and scalability. In this paper, we proposed a measure for optimal clustering. We defined clustering balance using the squared error sums. By searching the compromising point between intra-cluster and inter-cluster error sums, we are able to detect the optimal clustering configuration for any hierarchical clustering algorithms. For the purpose of finding an optimal configuration, an agglomerative clustering recovers the cluster configuration with the minimum clustering balance.

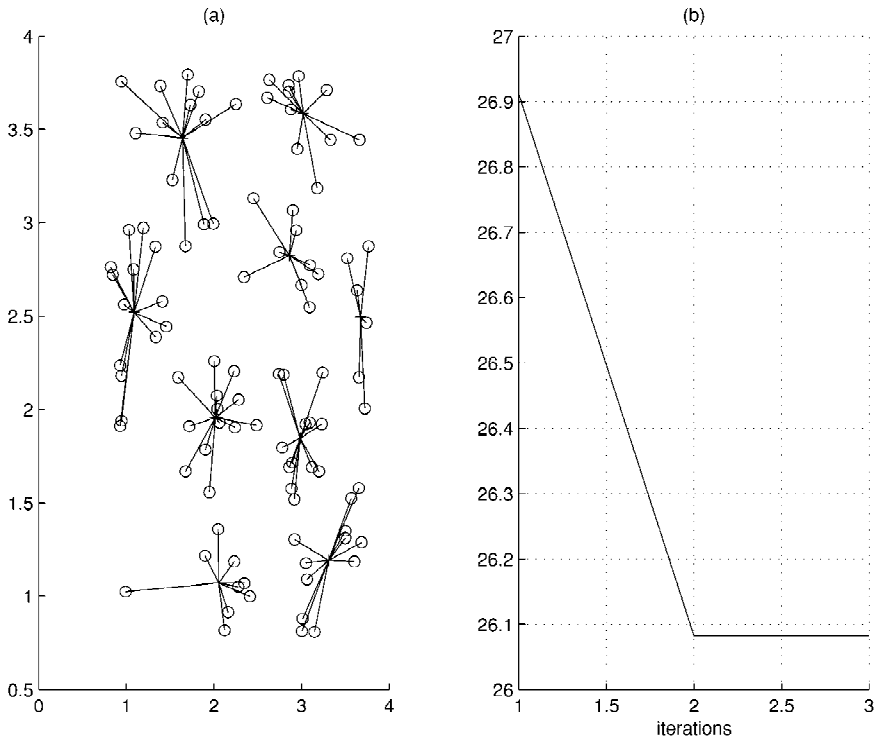

Figure 10. Additional clustering optimization and quick convergence. (a) Optimal configuration found by a hierarchical clustering algorithm. (b) Improved clustering performance.

Our approach is quite different from other traditional approaches. We evaluate clustering optimality using only internal properties of clusters and successfully achieve intuitive agreement for clustering optimality. In particular, the number of desired clusters and initial centroids can be estimated from the optimal cluster configuration, and provided to non-hierarchical partitional clustering methods. As a result, partitional clustering algorithms are able to converge more quickly and give lower clustering balance than those without our clustering measure.

When it comes to classification, multiple centroids in a class can be found using our clustering measure since each class in turn is a cluster. It is natural to assume that those centroids provide us with more accurate information describing the internal structure of a class than that represented by only one centroid. Therefore, we believe that classification performance will be enhanced if we exploit our clustering measure to find optimal sub-centroids in each class. The basic rationale of improved classification is that classification performance is contributed by comparing test data to multiple centroids instead of the single centroid. Therefore, we expect that many unknown contributions of our approach will be discovered in various

applications of clustering while our clustering measure consistently gives feasible solutions to optimal clustering.

Acknowledgements

The work of Y.J. was supported in part by the National Science Foundation grant CCR-9901992. The work of H.P. was supported in part by the National Science Foundation grant CCR-9901992.

References

1. Gose, E., Johnsonbaugh, R. and Jost, S. (1996), *Pattern Recognition & Image Analysis*, Prentice Hall, Upper Saddle River, NJ.
2. Jain, A.K., Murty, M.N. and Flynn, P.J. (1999), Data clustering: a review, *ACM Computing Surveys* 31, 264–323.
3. Shaffer, E., Dubes, R. and Jain, A.K. (1979), Single-link characteristics of a mode-seeking algorithm, *Pattern Recognition* 11, 65–73.
4. Kittler, J. (1976), A locally sensitive method for cluster analysis, *Pattern Recognition* 8, 22–33.
5. Zahn, C.T. (1971), Graph-theoretical methods for detecting and describing gestalt clusters, *IEEE Transactions on Computers* 20, 68–86.
6. Urquhart, R. (1982), Graph theoretical clustering based on limited neighborhood sets, *Pattern Recognition* 15, 173–187.
7. Gowdar, K.C. and Krishna, G. (1978), Agglomerative clustering using the concept for multispectral data, *Pattern Recognition* 10, 105–112.
8. Anderberg, M.R. (1973), Cluster Analysis for Applications, Academic Press, New York.
9. Abramowtiz, M. and Stegun, I.A. (1968), *Handbook of Mathematical Functions with Formulas, Graphics and Mathematical Tables*, US Govt. Printing Office, Washington, D.C.
10. Fortier, J.J. and Solomon, H. (1966), Clustering Procedures, In Krishnaiah, P.R. (ed). Multivariate Analysis, Academic Press, New York. pp. 493–506.
11. Jensen, R.E. (1967), A dynamic programming algorithm for cluster analysis, *Operations Research* 17, 1034–1057.
12. Jain, A.K. and Dubes, R.C. (1988), *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ.
13. Willet, P., (1988), Recent trends in hierarchic document clustering: a critical review, *Information Processing and Management* 24, 577–597.
14. Jancey, R.C., (1966), Multidimensional group analysis, *Austral. J. Botany* 14, 127–130.
15. MacQueen, J.B. (1967), Some methods for classification and analysis of multivariate observations. In: *the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol.1, AD 669871, University of California Press, Berkeley, CA, pp. 281–297.
16. Sibson, R. (1973), SLINK: an optimally efficient algorithm for the single-link cluster method, *Computer Journal* 16, 30–34.
17. Defays, D. (1977), An efficient algorithm for a complete link method, *Computer Journal* 20, 364–366.
18. Day, W.H.E. and Edelsbrunner, H. (1984), Efficient algorithms for agglomerative hierarchical clustering methods, *Journal of Classification* 1, 7–24.
19. Ng, R.T. and Han G., J. (1994), Efficient and effective clustering methods for spatial data mining. In: *Proceedings of the 20th VLDB Conference*, Santiago, Chile.

20. Voorhees, E.M. (1986), Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. In: *Information Processing and Management* Vol. 22, Pergamon, Oxford, pp. 465–476.
21. Li, X. (1990), Parallel algorithms for hierarchical clustering and cluster validity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1088–1092.
22. Bradley, P., Fayyad, U. and Reina, C. (1998) Scaling clusterin algorithms to large databases, *Knowledge Discovery and Data Mining*.
23. Guha, S., Rastogi, R. and Shim, K. (1998), CURE: an efficient clustering algorithm for large databases. In: *ACM-SIGMOD Int. Conf. on Management of Data*, Seattle, WA. 73–84.
24. Guha, S., Rastogi, R. and Shim, K. (1999), ROCK: a robust clustering algorithm for categorical attributes. In: *the 15th Int. Conf. on Data Eng*.
25. Ganti, V., Ramakrishnan, R. and Gehrke, J. (2000), Clustering large datasets in arbitrary metric spaces. *ACM*.
26. Charikar, M., Chekuri, C., Feder, T. and Motwani, R. (1997), Incremental clustering and dynamic information retrieval. In: *STOC'97*, El Paso, TX, pp. 153-180.
27. Dubes, R.C. (1987), How many Clusters are best? – an experiment, *Pattern Recognition* 20, 645–663.
28. Milligan, G.W. and Cooper M.C. (1985), An examination of procedures for detecting the number of clusters in a data set, *Psychometrika* 50, 159–179.
29. Tou, J.T. and Gonzalez, R.C. (1974) *Pattern Recognition Principles*, Addison-Wesley, Miami, FL.
30. Everett, B. (1975), *Cluster analysis*, Addison-Wesley, New York.
31. Boley, D.L. (1998), Principal direction divisive partitioning, *Data mining and knowledge discovery* 2, 325–344.
32. Mirkin, B. and Muchnik, I. (1998), Combinatorial Optimization in Clustering. In: Du, D.Z. and Pardalos, P.M. (eds), *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, Dordrecht pp. 261–329.
33. Karypis, G., Han E.S. and Kumar, V. (1999), CHAMELEON: a hierarchical clustering algorithm using dynamic modeling, *IEEE Computer: Special Issue on Data Analysis and Mining* 32, 68–75.
34. Duran, B.S. and Odell, P.L. (1977) *Cluster Analysis: A Survey*, Springer, Berlin.
35. Diday, E. and Simon, J.C. (1976), Clustering Analysis. In: Fu, K.S. (ed), Digital Pattern Recognition, Springer, Secaucus, NJ, pp. 47–94.
36. Garey, M.R. and Johnson, D.S. (1979), *Computers and Intractability: a guide to the theory of NP-completeness*, W.H.Freeman and Company, San Francisco, CA.
37. Crescenzi, P. and Kann, V. (1995), *A compendium of NP optimization problems*, URL site:http://www.nada.kth.se/∼viggo/problemlist/compendium2.
38. Ward, Jr. J.H. (1963), Hierarchical grouping to optimize an objective function, *Journal of the American Statical Association* 58, 236–244.
39. Forgy, E.W. (1965), Cluster analysis of multivariate data: efficiency versus interpretability of classification. In: *Biometric Society Meetings, Reverside, CA*, Abstract in Biometrics 21, 768.
40. Sebestyen, G.S. (1962), Pattern recognition by an adaptive process of sample set construction. *IRE Trans. on Info. Theory* IT–8.
41. MacQueen, J.B. (1966), Some methods for classification and analysis of multivariate observations. In: *Wester Management Science Inst.*, University of California, pp. 96, 1966
42. Ball, G.H. and Hall, D.J. (1964), Some fundamental concepts and synthesis procedures for pattern recognition preprocessors. In: *International Conference on Microwaves, Circuit Theory, and Information Theory*.
43. Kaufman, L. and Rousseeuw, P.J. (1990) *Finding Groups in Data: an Introduction to Clustering Analysis*, Academic Press, San Diego, CA.

44. Rasmussen, E. (1992), *Clustering Algorithms*. In: Frakes, W.B. and Baeza-Yates, R. (eds), Information Retrieval: Data Structures and Algorithms, Prentice-Hall, Upper Saddle River, NJ, pp. 419–442.
45. Jardine, N. and Rijsbergen, C.J. (1971), The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval* 7, 217–240.
46. Bennett, R.S. (1966), The intrinsic dimensionality of signal collections, *IEEE Transactions on Information Theory* 15, 517–525.
47. Zhang, T., Ramakrishnan, R. and Livny, M. (1996), BIRCH: An efficient data clustering method for very large databases, *SIGMOD Rec.* 25, 103–114.
48. Porter, M.F. (1980), An Algorithm for Suffix Stripping, *Program*, 14, 130–137.