Improving Cluster Method Quality by Validity Indices

N. Hachani and **H. Ounalli**
Faculty of Sciences of Bizerte, Tunisia narjes_hachani@yahoo.fr
Faculty of Sciences of Tunis, Tunisia habib.ounalli@fst.rnu.tn

Abstract

Clustering attempts to discover significant groups present in a data set. It is an unsupervised process. It is difficult to define when a clustering result is acceptable. Thus, several clustering validity indices are developed to evaluate the quality of clustering algorithms results. In this paper, we propose to improve the quality of a clustering algorithm called "CLUSTER" by using a validity index. CLUSTER is an automatic clustering technique. It is able to identify situations where data do not have any natural clusters. However, CLUSTER has some drawbacks. In several cases, CLUSTER generates small and not well-separated clusters. The extension of CLUSTER with validity indices overcomes these drawbacks. We propose four extensions of CLUSTER with four validity indices Dunn, $Dunn_{RNG}$, DB, and DB^*. These extensions provide an adequate number of clusters. The experimental results on real data show that these algorithms improve the clustering quality of CLUSTER. In particular, the new clustering algorithm based on DB^* index is more effective than other algorithms.

Introduction

Clusters operations attempt to partition a set of objects into several subsets. The majority of clustering algorithms partition a data set into a number of clusters based on some parameters such as the desired number of clusters (MacQueen 1967), the minimum number of objects (Ester *et al.* 1996) and the density threshold (Ester *et al.* 1996). Thus, the clustering depends on some criteria assuming that the resulting for the extracted clusters is the optimal. As a consequence, if a clustering algorithm has not been assigned proper values, the clustering result cannot be the partitioning that best fits the underlying data. However, determining the optimal number of clusters and evaluating the clustering result is not a trivial task. Therefore, several validity indices (Halkidi, Batistakis, & Vazirjianis 2001) have been developed to evaluate the clustering quality. The most used indices are those which are based on relative criteria. These indices attempt to evaluate clustering results comparing to other results created by different clustering algorithms or by the same algorithm but using different parameters. In the last case, we must apply the clustering algorithm several times with different parameters. In this article, we propose to use a validity index, underway and not in the end of clustering process, to improve clustering quality of a clustering algorithm called "CLUSTER" (Bandyopadhyay 2004). Thus, we partition the data set and we evaluate its quality at once. CLUSTER is a hierarchical clustering method that can automatically detect the number of clusters. It consists of two steps: partitioning a relative neighborhood graph (RNG) and merging small clusters. CLUSTER allows to identify the situation where the data do not have any natural clusters (one cluster). CLUSTER does not require a priori knowledge of clusters number and often provides good results. However, CLUSTER has some drawbacks. It can identify small clusters not well-separated. We propose to remove the second step of CLUSTER and to use a validity index as a stop condition of CLUSTER algorithm in order to improve its clustering quality. We develop four new clustering algorithms using the validity indices Dunn, $Dunn_{RNG}$, DB and DB^* respectively. Then, we evaluate the results and we choose the most appropriate index. The rest of the paper is organized as follows. Section 2 presents an overview of the existing clustering methods. Section 3 introduces the clustering validity indices. Section 4 describes CLUSTER algorithm. Section 5 provide a description of the new algorithms extending CLUSTER. Section 6 presents experimental results. Conclusions and future work are presented in section 7.

Related Work

Two main categories of clustering methods are distinguished: hierarchical methods and partitioning methods.

Partitioning Methods construct a single partition of the data set. The principal partitioning algorithms are K-means (MacQueen 1967), K-medoids such as PAM (Kaufman & Rousseeuw 1990), CLARA (Kaufman & Rousseeuw 1990) and CLARANS (Ng & Han 1994), density-based algorithms such as DBSCAN (Ester *et al.* 1996) and graph-based algorithms (Bandyopadhyay 2004). The K-means algorithm (MacQueen 1967) provides K clusters by minimizing the intra-cluster distance error criterion. Each cluster is represented by its center of gravity. The K-medoids algorithms represent each cluster by one of his central objects. These algorithms try to improve current clustering by exchanging one of the medoids of the partition with one non-medoid and then compare the total quality of this ”new” clustering with the total quality of the ”old” one. The density-based methods generate clusters of various forms without specifying the number of clusters such as DBSCAN (Ester *et al.* 1996). It defines an area to be a neighborhood determined by an object. However, DBSCAN is very sensitive to the choice of input parameters: density threshold and minimal number of point. Also, it does not generate clusters of different densities. Graph-based clustering algorithms are an important and interesting category of clustering methods. In general, the similarity is expressed by a graph. The basic idea is to start from a neighborhood graph and then to remove the edges whose size is higher than a certain threshold. The method introduced in (Zhang, Ramakrishan, & livny 1996) is based on the relative neighborhood graph (Toussaint 1980) and can detect well-separated clusters. However, it is sensitive to the choice of an input parameter.

Hierarchical Methods construct a hierarchy of clusters and have several advantages:

- It is less sensitive to largely differing densities of clusters.
- It identifies the natural clusters in the database.
- It is less influenced by clusters shapes

Birch (Zhang, Ramakrishan, & livny 1996) stores information about cluster based on a hierarchical data structure (CF-tree). This clustering algorithm groups the data in an incremental and dynamic way according to their order of insertion. Chameleon (Karpys, Han, & Kumar 1999) allows the partitioning of a neighborhood graph into small clusters. It then merges these clusters based on a similarity measure. However, Birch (Zhang, Ramakrishan, & livny 1996) and Chameleon (Karpys, Han, & Kumar 1999) need some input parameters. CLUSTER (Bandyopadhyay 2004) can automatically detect any number of clusters. However, it can generate small clusters because of some minor variation in the distances. We propose an improvement of this method by using a validity index as a stop condition of the clustering process.

Validity Indices

Clustering is an unsupervised process and the majority of the clustering algorithms depend on certain assumptions in order to define the subgroups presented in the data set. As a consequence, in most application, the clustering result requires some sort of evaluation of its validity. Several clustering validity techniques and indices have been developed (Halkidi, Batistakis, & Vazirjianis 2001). There are three categories of validity indices. The first is based on external criteria. This implies that the results of a clustering algorithm are evaluated based on a pre-specified structure, which is imposed on a data set and reflects our intuition about the clustering algorithms. The second is based on internal criteria. We may evaluate the results of a clustering algorithm in terms of quantities that involve the vectors of the data set themselves (proximity matrix). The third approach is based on relative criteria. Its basic idea is the evaluation of the clustering structure by comparing it to other clustering schemes. The first two categories are based on statistical tests and their major drawback is their high computational cost. Therefore, we are interested in the third category of validity indices. Several indices are developed (Halkidi, Batistakis, & Vazirjianis 2001) such as Dunn, DB, SD, SD_{bw} and DB^* (Minho & Ramakrishna 2005). However, SD and SD_{bw} need a very high computational cost (Halkidi, Batistakis, & Vazirjianis 2001). We are interested in the indices Dunn, $Dunn_{RNG}$, DB and DB^*. In (Kovacs, Csaba, & Attila 2002), some validity indices are evaluated with several different data sets. According to this evaluation, Dunn identifies the appropriate result. The Dunn index based on RNG is more robust to the presence of noise than Dunn. Moreover, it is based on RNG concept as CLUSTER algorithm. In the sequel, we present the used indices.

Dunn

It is a cluster validity index for crisp clustering proposed by Dunn (DUNN 1973). It attempts to identify ”compact and well-separated clusters”. The Dunn's validity index, D, is defined as:

$$D_{nc} = \min_{i=1,..,nc}\{\min_{j=i+1...,nc}(\frac{d(C_i,C_j)}{max_{k=1...,nc}diam_k})\} \quad (1)$$

where nc is the number of clusters, $d(C_i,C_j) = \min_{x\in C_i,y\in C_j} d(x,y)$ is a function of dissimilarity between two clusters C_i and C_j defined by $d(C_i,C_j) = \min_{x\in C_i,y\in C_j} d(x,y)$. and $diam_k$ is the diameter of a cluster defined by $diam_k = \max_{x,y\in C_k} d(x,y)$. Based on Dunn definition, we conclude that large value of Dunn's validity index represents a good clustering. Therefore, the number of cluster that maximizes D, represents the optimal number of clusters. Three indices, based on Dunn index, are proposed by Pal and Biswas (Pal & Biswas 1997). They use for their definition the concepts of the minimum spanning tree (MST), the relative neighborhood graph (RNG) and the gabriel graph respectively. The Dunn index based on RNG is defined by the following equation:

$$D_{nc} = \min_{i=1,..,nc}\{\min_{j=i+1,...,nc}(\frac{d(C_i,C_j)}{max_{k=1,...,nc}diam^{RNG}})\} \quad (2)$$

Let e_{max}^{RNG} the edge of the RNG which has the maximal weight. $diam^{RNG}$ is defined by the weight of e_{max}^{RNG}.

Davies-Bouldin (DB) index

DB is based on a similarity measure R_{ij} between the clusters C_i and C_j. R_{ij} uses a measure of dispersion of a cluster C_i and the dissimilarity measure d_{ij}. It must satisfy the following conditions:

- $R_{ij} \geq 0$.
- $R_{ij} = R_{ji}$.
- If $S_i = 0$ and $S_j = 0$ then $R_{ij} = 0$.
- If $S_j > S_k$ and $d_{ij} = d_{ik}$ then $R_{ij} > R_{ik}$.
- If $S_j = S_k$ and $d_{ij} < d_{ik}$ then $R_{ij} > R_{ik}$.

Where d_{ij} defines the distance between clusters C_i and C_j. Usually, it is calculated as the distance between the centers of two clusters. Given the centroid c_i of the cluster C_i, S_i is a scatter distance. It is defined by:

$$S_i = \frac{1}{|C_i|}\sum_{x\in C_i} \parallel x - c_i \parallel . \quad (3)$$

R_{ij} is defined by:

$$R_{ij} = \frac{S_i + S_j}{d_{ij}} \quad (4)$$

$$DB = \frac{1}{nc}\sum_{i=1}^{nc} R_i. \qquad (5)$$

$$R_i = \max_{j=1..nc, i\neq j}(R_{ij}). \qquad (6)$$

DB represents the average similarity between each cluster C_i and its most similar one. Since it is desirable to minimize the similarity between clusters, we seek the clustering that minimizes DB.

DB* index

DB^* (Minho & Ramakrishna 2005) index is an improvement of the DB index. DB is the average of the maximum of R_{ij} of each cluster. R_{ij} has the maximum in one of the following conditions:

- When d_{ij} dominates.
- When (S_i + S_j) dominates.
- By combination of d_{ij} and (S_i+ S_j).

"Dominate" means that it is a decisive factor for determining $max(R_{ij})$. In the first case, d_{ij} has a relatively small value compared with $(S_i + S_j)$ and is usually $min(d_{ij})$. This represents a situation where two clusters are located very close to each other and they need to be merged. In the second case, $(S_i + S_j)$ has relatively very large value usually when it is equal to $max(S_i + S_j)$. This is a situation of unnecessary merging. In the third case, R_{ij} has the maximum value when neither $min(d_{ij})$ nor $max(S_i+S_j)$ occurs. It is some combination of d_{ij} and $(S_i + S_j)$. Thus, DB can effectively have the minimum value when d_{ij} dominates at $nc > nc_{optimal}$ and $(S_i + S_j)$ dominates at $nc > nc_{optimal}$. From the assumption, that in an ideal situation $1/min\{d_{ij}\}$ and $max(S_i + S_j)$ have relatively large value when $nc > nc_{optimal}$ and $nc < nc_{optimal}$. Hence, DB can be redefined as DB*:

$$DB^*(nc) = \frac{1}{nc}\sum_{i=1}^{nc}\left(\frac{max_{k=1,\ldots,nc,k\neq i}\{S_i + S_k\}}{min_{l=1,\ldots,nc,l\neq i}\{d_{il}\}}\right). \qquad (7)$$

Cluster Method

CLUSTER is a hierarchical clustering method based on a partitioning of a relative neighborhood graph. It detects automatically the number of clusters. It can also generate clusters of various densities. CLUSTER identifies the situation where it is useless to apply clustering (unique cluster). To present CLUSTER, it is necessary to introduce the following concepts:

Distance Measure: The distance between two elements of the DB is expressed by the Euclidean distance. In the case when data is represented in d dimensions, the distance between two objects $x = \{x_1, x_2, x_3, ..., x_d\}$ and $y = \{y_1, y_2, y_3, ..., y_d\}$ is defined by:

$$d(x, y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}. \qquad (8)$$

Relative Neighborhood Graph: Let $X = \{x_1, x_2, ..., x_n\}$ be a set of points. Two points x_i and x_j of X are relative neighbors (Toussaint 1980) if the following condition is satisfied:

$$d(x_i, x_j) \leq max[d(x_i, x_k), d(x_j, x_k)], \forall x_k \in X, k \neq i, j. \qquad (9)$$

Intuitively, this means that two points are relative neighbors if they are at least as close to each other as they are to any other point. The relative neighborhood graph (RNG) (Toussaint 1980) is constructed by joining via an edge the points that are relative neighbors. The weight of an edge between x_i and x_j represents the distance $d(x_i, x_j)$.

Description of CLUSTER

CLUSTER algorithm consists of two steps: partitioning of the RNG and merging small clusters. In the first step, CLUSTER constructs an initial RNG and then tries to divide it into several subgraphs based on a threshold which is dynamically computed. This process is applied iteratively for each obtained subgraph until a stop criterion is reached. In the second step, CLUSTER merges small clusters and removes the noise clusters. These steps are detailed below.

Step 1: Partitioning of the RNG. In the following, the term "variation" means the difference between two distances. The initial or the intermediate RNG, noted G, is partitioned into several subgraphs as follows:

1. Sort the distances (weights of the edges) in ascending order.
2. Compute the variations between each two successive distances and sort these variations.
3. Compute the intermediate variation t defined by: $t = (vmin + vmax) \div 2$. Where $vmin$ is the minimal variation and $vmax$ is the maximum variation.
4. Determine the value of the threshold, noted th, which allows partitioning the graph. This threshold is the distance d_i satisfying the following conditions:
   (a) $d_{i+1} - d_i \geq t$
   (b) $d_i \geq 2 \times Min$

   Where $(d_{i+1} - d_i)$ is the difference between two successive distances of the ordered list of distances and Min is the minimal distance.
5. If the threshold is found we remove from G all edges whose weights are strictly greater than th. Hence, we construct a set of subgraphs.
6. The previous operations can be repeated for each constructed subgraph.

The above step terminates when at least one of the following conditions is satisfied:

1. Before computing the variations between distances, we verify the following condition: $Max < 2 \times Min$. Where Min is the minimal distance and Max is the maximal distance. This condition means that the inter-cluster relative neighbors are close to each other.
2. The threshold is not found: $th \leq 2 \times Min$.
3. $\mid Component \mid > \sqrt{\mid G \mid}$ Where $\mid Component \mid$ is the number of the obtained subgraphs and $\mid G \mid$ is the size of G (number of nodes). This rule of the thumb states that the maximum number of clusters that may be present is approximately $\sqrt{\mid G \mid}$.

Step 2: Merging of Small Clusters. In CLUSTER algorithm, a small cluster is defined as a cluster whose size is lower than 5% of the DB size. This cluster is merged with the nearest cluster. However, if the threshold at which the small cluster got partitioned out is larger than $\lambda \times Max$, this cluster is considered as noise and it is removed from DB. Max is the maximum value of the nearest cluster. The value kept for λ in CLUSTER is 3.

Improvement of CLUSTER Quality

We propose to remove the second step of CLUSTER algorithm and to include a new stop condition in the first step based on a validity index. This condition overcomes the CLUSTER drawbacks and evaluates the clustering quality underway and not in the end of the algorithm

The drawbacks of CLUSTER

The first step may generate small clusters because of some minor variation in the distances (Bandyopadhyay 2004). The merging of small clusters resolves this problem partially. We distinguish mainly the two following drawbacks:

- Clusters which must be naturally merged are not merged. We identify these clusters by ”missing fusion”.
- Clusters which are not noise are considered as noise clusters . These clusters are identified by ”not real noise”.

These drawbacks are explained in the following:

1. Missing fusion: in CLUSTER algorithm, a cluster is considered small when its size is smaller than 5% of the size of the DB. However, we can obtain, as consequence of the first step, a cluster whose size is greater than 5% of the DB size and it should be merged with others neighbors clusters.
   Example 1
   We consider the following two clusters C_i and C_j among the 10 generated by applying CLUSTER to the ”Books” DB.
   $C_i = (28.95, 28.95, 29, 29, 29, 29.49, 29.67, 29.67, 29.67,$
   $29.69, 29.69, 29.69, 29.69, 29.69, 29.69, 29.69, 29.7, 29.7,$
   $29.95, 29.95, 29.95, 29.95, 30).$
   $C_j = (30.64, 30.95, 30.95, 30.95, 30.99, 30.99, 31.47, 31.5,$
   $31.95, 31.95, 31.97, 31.99, 32.43, 32.52, 32.95, 32.97, 32.99,$
   $32.99, 32.99, 32.99, 32.9932.99, 32.99, 32.99, 32.99, 32.99,$
   $32.99, 32.99, 32.99, 32.99, 33.08, 33.5, 33.84, 33.95, 33.95,$
   $33.99, 34.14, 34.19, 34.19, 34.5, 34.62, 34.62, 34.62, 34.76$
   $34.95, 34.95, 34.95, 35, 35, 35, 35, 36, 36.06).$

   According to this database, these two clusters must be merged. However, they are not merged because they are not considered as small clusters. In fact, their sizes are respectively 23 and 52 which is greater than 5% of the DB size (400).
2. Not real noise: if a small cluster is not a cluster noise, CLUSTER merges it with its nearest neighbor. However, if the nearest cluster includes only one element, the value of Max is 0. In this case, the threshold is always greater than $3 \times max$. Thus, this small cluster is always considered as a cluster noise and will be removed.
   Example 2
   Consider again the ”Books” DB. As consequence of step 1, we obtain, among the small clusters, the following ones:
   $C_i = (22.65).$
   $C_j = (22.79).$
   These two clusters are not merged because the value of Max is equal to 0. C_i and C_j are considered as clusters noise and they are removed. However, in reality, these clusters are not noise. This generates an erroneous distribution of clusters.

Optimisation of CLUSTER Quality

We propose an extension of CLUSTER based on a validity index. The idea is to include in CLUSTER a stopping condition, based on this index. We test the extension of CLUSTER with different validity indices: Dunn, $Dunn_{RNG}$, DB and DB^*. Then, we evaluate these extensions. This proposal attempts to avoid the generation of small clusters and improve the quality of CLUSTER algorithm. CLUSTER is extended as follows:

1. For each iteration i, we compute the value of the validity index, noted ind_i, according to the overall number of clusters obtained until this iteration.
2. We compare ind_i with the index value of previous step ind_{i-1}. If ind_i decreases(case of Dunn or $Dunn_{RNG}$) or increase(case of DB or DB_*), the algorithm terminates with the clustering result of ind_{i-1}. If ind_i does not decrease (or increase)and none of the other stop criteria is satisfied, the algorithm continues normally.

Experiments have showed that the new algorithms have the following advantages over classic CLUSTER algorithm:

- Avoid generating small clusters.
- Obtain a satisfying number of clusters.
- Increase the effectiveness of CLUSTER by avoiding several merging of small clusters.
- Improve the quality of clustering by identifying compact and well-separated clusters.
- Evaluate the clustering quality underway the clustering process.

Experiments

This section presents the experiments that have been performed. A concise description of the experimentation platform and data sets is also given.

Experimental Set-Up: we have used the following DB:

1. The ”Books” DB includes over 400 prices of books. This base is collected from ”www.amazon.com”. It contains two clusters.
2. The ”Censusage” DB (Blake & Merz 2004) includes 606 objects. We are interested in the value of TSH which allows to identify two clusters.
3. The ”Pima diabets” DB (Blake & Merz 2004) includes 763 objects. We are interested in the values of the plasma glucose concentration a 2 hours in an oral glucose tolerance test which allows to identify two clusters.
4. A selection of 1000 objects from the ”Hypothyroid” DB (Blake & Merz 2004). We are interested in the values of the TSH which allows to identify two clusters.
5. A selection of 5723 objects from the ”Thyroid” DB (Blake & Merz 2004). We consider the value of the TSH which allows to identify two clusters.

We compare the new clustering algorithms, (CLUSTER with Dunn (CLST-Dunn), CLUSTER with $Dunn_{RNG}$ (CLST-$Dunn_{RNG}$), CLUSTER with DB index (CLST-DB) and CLUSTER with DB^* (CLST-DB^*)) with CLUSTER and an improved K-means(deterministic K-means) (Su & Dy 2004). In order to achieve an effective comparison of these algorithms in the same conditions (processor, memory, data), we implement them in C++ language with a Processor INTEL Pentium III 733 MHz and 320 MB of RAM.

Experimental Results: the results, in Tables 1, 2 and 3 show that the different extensions of CLUSTER give better results compared to CLUSTER and deterministic K-means (Su & Dy 2004), noted deter K-means. For example, for the DB used for experimentation, the extension of CLUSTER with validity indices identifies the optimal number of clusters. CLUSTER generates a large number of clusters compared to the real number of clusters (Rnbc). In particular the extension of CLUSTER with DB^* is more effective than the other extensions in term of execution time (TE).

Conclusion

In this work, we have proposed a new clustering approach by extending CLUSTER (Bandyopadhyay 2004) based on validity indices (Halkidi, Batistakis, & Vazirjianis 2001). To evaluate the effectiveness of our approach, we performed experiments on five ”effective” DB. Our results show that the extension of CLUSTER with validity indices produces better result than CLUSTER

Table 1: Clustering results1

Databases	**Rnbc**	**CLUSTER**		**Deter K-means**	
		Nbc	TE	Nbc	TE
Books	2	25	14	2	17
Censusage	3	1	11	2	34
Pima diabets	2	3	12	2	24
Thyroid	2	5	172	2	4665
Hypothyroid	2	7	16	2	24

Table 2: Clustering results2

Databases	**Rnbc**	**CLST-Dunn**		**CLST-$Dunn_{RNG}$**	
		Nbc	TE	Nbc	TE
Books	2	2	20	2	13
Censusage	3	3	15	3	12
Pimadiab	2	2	20	2	12
Thyroid	2	2	949	2	50
Hypoth	2	2	27	2	14

(Bandyopadhyay 2004) and deterministic K-Means. In particular, the extension of CLUSTER with the DB^* validity index is the best in execution time compared to extensions using $Dunn$, $Dunn_{RNG}$ and DB. The new clustering algorithm allows:

1. Automatic detection of the adequate number of clusters.
2. Improving the clustering quality of CLUSTER by generating compact and well-separated clusters.
3. Increasing the effectiveness of CLUSTER by avoiding several intermediate merging which improve the execution time.
4. Evaluating clustering quality underway the clustering algorithm.

In future work, we will evaluate our new algorithms using very large DB and we will use our clustering algorithm to develop an automatic membership function generation approach.

Table 3: Clustering results3

Databases	**Rnbc**	**CLST-DB**		**CLST-DB^***	
		Nbc	TE	Nbc	TE
Books	2	2	8	2	8
Censusage	3	3	9	3	9
Pima diabets	2	2	10	2	10
Thyroid	2	2	30	2	30
Hypothyroid	2	2	11	2	10

References

Bandyopadhyay, S. 2004. An automatic shape independent clustering techniques. *Pattern Recognition* 37:33–45.

Blake, C., and Merz, C. 2004. Uci repository of machine learning databases. http://www.ics.uci.edu/ mlearn/mlrepository.html. accessed 4th march 2004.

DUNN, J. 1973. A fuzzy relative of the isodata process and its use in detecting compact well separated clusters. *J.ccybern* 3:32–75.

Ester, M.; Kriegel, H.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd international Conference on Knowledge Discovery and Data Mining*.

Halkidi, M.; Batistakis, Y.; and Vazirjianis, M. 2001. On clustering validation techniques. *Journal of Intelligent Information Systems.* 107–145.

Karpys, G.; Han, E.; and Kumar, V. 1999. Chameleon: A hierarchical clustering algorithm using dynamic modeling. In *IEEE Computer*, 68–75.

Kaufman, L., and Rousseeuw, P. 1990. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Welley and Sons.

Kovacs, F.; Csaba, L.; and Attila, B. 2002. Cluster validity measurement techniques. *Pattern Recognition* 30.

MacQueen, J. 1967. Some methods for classication and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Minho, K., and Ramakrishna, R. 2005. New indices for cluster validity assessment. *Pattern Recognition Letters* 26:2353–2363.

Ng, R., and Han, J. 1994. Efficient and effective clustering method for spatial data mining. In *In proc.of the 20th VLDB Conference*, 144–155.

Pal, N., and Biswas, J. 1997. Cluster validation using graph theoretic concepts. *Pattern Recognition* 30.

Su, T., and Dy, J. 2004. A deterministic method for initializing k-means clustering. In *Tools with Artificial Intelligence 2004*.

Toussaint, G. 1980. The relative neighborhood graph of a finite planar set. *Pattern Recognition* 12:261–268.

Zhang, T.; Ramakrishan, R.; and livny, M. 1996. Birch: Efficient data clustering method for very large databases. *SIGMOD 96 Montreal,Canada.* 103–114.