Measuring the Quality of Approximated Clusterings

Hans-Peter Kriegel, Martin Pfeifle

University of Munich, Institute for Computer Science
http://www.dbs.informatik.uni-muenchen.de
{kriegel,pfeifle}@dbs.ifi.lmu.de

Abstract. Clustering has become an increasingly important task in modern application domains. In many areas, e.g. when clustering complex objects, in distributed clustering, or when clustering mobile objects, due to technical, security, or efficiency reasons it is not possible to compute an "optimal" clustering. Recently a lot of research has been done on efficiently computing approximated clusterings. Here, the crucial question is, how much quality has to be sacrificed for the achieved gain in efficiency. In this paper, we present suitable quality measures allowing us to compare approximated clusterings with reference clusterings. We first introduce a quality measure for clusters based on the symmetric set difference. Using this distance function between single clusters, we introduce a quality measure based on the minimum weight perfect matching of sets for comparing partitioning clusterings, as well as a quality measure based on the degree-2 edit distance for comparing hierarchical clusterings.

1 Introduction

Knowledge Discovery in Databases (KDD) tries to identify valid, novel, potentially useful, and ultimately understandable patterns in data. Although there exist many different data mining algorithms which extract useful knowledge, many of them are not applicable to large databases of complex objects due to efficiency problems. An important area where this complexity problem is a strong handicap is that of clustering. Effective clustering algorithms which are not efficient are almost as worthless as non-effective clustering algorithms. Therefore, there have been various approaches for accelerating algorithms for complex clusterings.

One very promising approach is approximated clustering. Instead of trying to compute the expensive exact clustering structure, an approximated clustering is computed. The crucial question is how is the quality of the resulting approximated clustering affected. Many of the presented approaches in the literature try to justify the achieved efficiency boost by a tailor-made quality measure demonstrating that the resulting clustering is "quite" similar to an optimal one. In this paper, we suggest objective quality measures suitable for evaluating the quality of all kinds of approximated clusterings, e.g. partitioning clusterings stemming from distributed algorithms [JKP 04], clusterings of mobile objects [LHY 04], or approximated hierarchical clusterings based on data bubbles [ZS 03]. As we do not propose a new effective and efficient data mining algorithm, we do not run into the risk of choosing a "suitable" quality measure by means of which we can justify the "suitability" of our new data mining algorithm. Instead, in this paper, we propose objective quality measures helping to assess the merits of newly proposed approximated data mining algorithms.

The remainder of this paper is organized as follows. In Section 2, we shortly sketch different application ranges of approximated clusterings. In Section 3, we present two similarity measures allowing us to compare approximated and exact clusterings to each other. We motivate the use of the *symmetric set difference* for measuring the similarity between two single clusters. A partitioning clustering algorithm creates a set of clusters. Based on the metric symmetric set difference reflecting the similarity between single clusters, we propose a new distance measure for sets of clusters, i.e. for partitioning clusterings, which is based on the *minimum weight perfect matching of sets*. This distance measure demonstrates to be suitable for defining similarity between partitioning clusterings. In order to measure the similarity between exact and approximated hierarchical clusterings, we introduce a quality measure which is based on the *degree-2 edit distance* [ZWS 96]. We close this paper in Section 4 with a short summary and a few remarks on future work.

2 Application Ranges of Approximated Clusterings

As many of the clustering algorithms presented in the literature [JMF 99] are very time consuming, different approaches have been presented to accelerate them. Often this acceleration goes hand in hand with a decreasing quality. In other new emerging application areas it is, due to security or technical reasons, often not possible to construct a correct clustering. Thus approximated clusterings aim at either accelerating clusterings or at opening up new emerging application areas.

2.1 Accelerated Clusterings

We will now shortly present different approaches which compute an approximated clustering due to efficiency reasons.

Sampling. The simplest approach is to use sampling and apply the expensive data mining algorithms to a subset of the data space. Typically, if the sample size is large enough, the result of the data mining method on the sample reflects the exact result well enough [Ben 04].

Partitioning Approaches. The different variants of k-means [McQ 65] start with a random partition and iteratively reassign the objects to certain cluster representatives until a convergence criterion is met. For example, the iteration may stop when no objects are reassigned from one cluster representative to another one any more, or when an error criterion ceases to decrease significantly, or after a certain number of iterations has been performed. Another approach for accelerating density-based clustering algorithms is based on grid cells [JMF 99].

Hierarchical Approaches. There also exist a variety of approximated hierarchical clusterings [PR 88]. For instance, there exist efficient approximated versions of hierarchical clustering approaches for vector and non-vector data which are based on "data bubbles" [GRG+ 99] [ZS 03]. These approaches augment suitable representatives with additional aggregated information describing the area around the representatives.

2.2 Emerging Clusterings

In our modern world, we have many different situations where it is not possible to analyze the available data once on one single computer. Examples for these emerging application ranges are distributed clustering [JKP 04] and the clustering of mobile objects [LHY 04]. Nevertheless, in both areas the users want to extract knowledge from the available data based on approximated clusterings.

Distributed Clusterings. Traditional KDD algorithms require full access to the data which is going to be analyzed, i.e. the data has to be located at one single site. Nowadays, large amounts of heterogeneous, complex data reside on different, independently working computers which are connected to each other via local or wide area networks, e.g. distributed mobile networks, or sensor networks. The transmission of huge amounts of data from one site to another central site is in some application areas almost impossible. In astronomy, for instance, there exist several highly sophisticated space telescopes spread all over the world. These telescopes gather data unceasingly. Each of them is able to collect 1GB of data per hour [Ha 00] which can only, with great difficulty, be transmitted to a global site to be analyzed centrally there. On the other hand, it is possible to analyze the data locally where it has been generated and stored. Aggregated information of this locally analyzed data can then be sent to a central site where the information of different local sites are combined and analyzed. Obviously the quality of the resulting distributed clustering heavily depends on the used algorithms for extracting aggregated information on the local sites and for combining this information on a server site. In order to evaluate the effectiveness of the used algorithms we need suitable quality measures for comparing distributed clusterings to reference clusterings where all data is available on one central computer.

Clustering Moving Objects. Recently Han et. al. [LHY 04] proposed an algorithm for clustering moving objects. Due to the advances in positioning technologies, the real time information of moving objects becomes increasingly available imposing new challenges to the database research community. The authors studied the problem of clustering moving objects which allows to catch interesting pattern changes during the motion process. By maintaining moving micro clusters it is possible to efficiently compute a clustering at any given time instance with a "relatively high quality".

In the following section, we will present quality measures which help to decide whether the proposed data mining algorithms really produce "high quality" with respect to a given reference clustering.

3 Similarity Measures for Clusterings

In the literature there exist some approaches for comparing partitioning [Mei 03] [BL 04] and hierarchical [FM 83] clusterings to each other. All of these approaches do not take noise objects into consideration which naturally occur when using density-based clustering algorithms such as DBSCAN [EKSX 96] or OPTICS [ABKS 99]. The similarity measures introduced in this paper are suitable for generally measuring the quality between partitioning and hierarchical clusterings even if noise is considered. The quality of the approximated clustering is always measured with respect to a reference clustering which is computed on the exact object representations.

In Section 3.1, we formally introduce the notion of partitioning and hierarchical clusterings. Both definitions rely on the notion of a "cluster". Before discussing suitable similarity distance functions, i.e. quality measures, for partitioning and hierarchical clusterings in Section 3.3 and Section 3.4, we introduce a similarity measure suitable for comparing two single clusters to each other in Section 3.2.

3.1 Modelling of Clusterings

Partitioning Clusterings. Partitioning clustering algorithms obtain a simple partition of the database. This has advantages for extremely large data sets for which the typically more expensive hierarchical algorithms may incur very high runtime costs. The resulting partitioning clusterings can be described by a set of sets of data objects. Each clustering consists of a set of clusters, where the clusters themselves are sets of objects from a database.

Definition 1 (cluster).

A cluster C is a non empty subset of objects from a database DB, i.e. $C \subseteq \text{DB}$ and $\text{C} \neq \emptyset$.

Definition 2 (partitioning clustering).

Let DB be a database of arbitrary objects. Furthermore, let $C_1, ..., C_n$ be pairwise disjoint clusters of DB, i.e. $\forall\ i, j \in 1, ..., n: i \neq j \Rightarrow C_i \cap C_j = \emptyset$ holds. Then, we call $CL_p = \{C_1, ..., C_n\}$ a partitioning clustering of DB.

Note that, for instance, the partitioning clustering algorithm k-means [McQ 65] assigns each object to exactly one cluster. On the other hand, the density-based partitioning clustering algorithm DBSCAN assigns each object either to noise or to a cluster. Thus, due to the handling of noise, we do not demand from a partitioning clustering $CL_p = \{C_1, ..., C_n\}$ that $C_1 \cup ... \cup C_n = DB$ holds (cf. Definition 2).

Hierarchical Clusterings. Hierarchical clustering algorithms produce a nested series of partitions instead of the single, flat partition produced by partitioning methods. Often, the result of such a clustering is represented in the form of a tree, called the *dendrogram*, that iteratively splits the database into smaller and smaller subsets (until each subset contains only one object). According to the approach presented in [SQL+ 03], dendrograms can easily be transformed into reachability plots which are 2D plots computed by the hierarchical clustering algorithm OPTICS [ABKS 99]. By means of suitable cluster recognition algorithms [ABKS 99][BKK+ 04][SQL+ 03] we can generate a hierarchical tree structure from a reachability plot, where each tree node corresponds to one cluster.

Definition 3 (hierarchical clustering).

Let DB be a database of arbitrary objects. A *hierarchical clustering* is a tree t_{root} where each subtree t represents a cluster C_t, i.e. $t = (C_t, (t_1, ...,t_n))$, and the n subtrees t_i of t represent non-overlapping subsets C_{t_i}, i.e. $\forall i, j \in 1, ..., n: i \neq j \Rightarrow C_{t_i} \cap C_{t_j} = \emptyset \wedge C_{t_1} \cup ... \cup C_{t_n} \subseteq C_t$. Furthermore, the root node t_{root} represents the complete database, i.e. $C_{t_{root}} = DB$.

Again, as some hierarchical clustering algorithms take noise into consideration and some not, we do not demand from the n subtrees t_i of $t = (C_t, (t_1, ...,t_n))$ that $C_{t_1} \cup ... \cup C_{t_n} = C_t$ holds.

C_1 set of △
C_2 set of ●

$\Rightarrow$ $d_\Delta^{norm}(C_1, C_2) = (2+4) / 18$
$d_\Delta(C_1, C_2) = 2+4$

Figure 1: Symmetric Set Difference.

3.2 Similarity Measure for Clusters

As outlined in the last section, both partitioning and hierarchical clusterings consist of flat clusters. In order to compare flat clusters to each other we need a suitable distance measure between sets of objects. One possible approach is to use distance measures as used for constructing distance-based hierarchical clusterings, e.g. the distance measures used by *single-link*, *average-link* or *complete-link* [JMF 99]. Although it is advisable to use such distance measures for the construction of hierarchical clusterings, these measures are not suitable when it comes to evaluating the quality of partitioning clusterings. Typically, users are only interested in the clustering result which is a binary relation (*ClusterID*, *ObjectID*). Note that Definition 2 exactly reflects this binary result set. Consequently, the similarity of two clusters with respect to quality solely depends on the number of identical objects contained in both clusters. Therefore, we propose to use the *symmetric set difference* as distance measure between two clusters (cf. Figure 1).

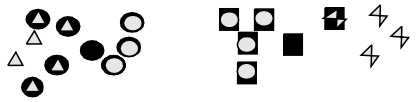
Definition 4 (symmetric set difference).
Let C_1 and C_2 be two clusters of a database DB. Then the symmetric set difference d_Δ: $2^{DB} \times 2^{DB} \rightarrow [0..|DB|]$ and the normalized symmetric set difference version d_Δ^{norm}: $2^{DB} \times 2^{DB} \rightarrow [0..1]$ are defined as follows:

$$d_\Delta(C_1, C_2) = |C_1 \cup C_2| - |C_1 \cap C_2| \quad and \quad d_\Delta^{norm}(C_1, C_2) = \frac{|C_1 \cup C_2| - |C_1 \cap C_2|}{|C_1 \cup C_2|}$$

We would like to state that both the unnormalized and the normalized symmetric set difference form a metric. Note that not every reasonable attempt at normalization results in a metric. For instance, dividing by $|C_1| + |C_2|$ instead of by $|C_1 \cup C_2|$, fails to satisfy the triangle inequality.

3.3 Similarity Measure for Partitioning Clusterings

In the last section, we introduced a metric distance measure suitable for measuring the similarity between two clusters, i.e. between two sets of objects. In this section, we will concentrate on the computation of a similarity measure suitable for measuring the quality of an approximated clustering w.r.t. a reference clustering (cf. Figure 2). Thus, the crucial question is what is a suitable distance measure between sets of sets. In the literature there exist several approaches for comparing two sets S and T to each other. In [EM 97], the authors survey the following distance functions, which are computable in polynomial time: the *Hausdorff distance*, the *sum of minimal distances*, the *(fair-)surjection distance* and the *link distance*. All of these approaches rely on the possibility to match several elements in one set to just one element in the compared set which is questionable when comparing the quality of an approximated clustering to a reference clustering.



approximated partitioning clustering quality Q_{APC}:
$Q_{APC}(Cl^{\approx}, Cl^{ref}) = 1-(((2+4)+(3+2)+(4))/((6+7+4)+(8+6)))$
$\Rightarrow Q_{APC}(Cl^{\approx}, Cl^{ref}) = 1-15/31 = 16/31$

$Cl^{ref} = \{C_1^{ref}, C_2^{ref}\}$, where
C_1^{ref} consists of all ● and C_2^{ref} of all ■

$Cl^{\approx} = \{C_1^{\approx}, C_2^{\approx}, C_3^{\approx}\}$, where
$C_1^{\approx}$ consists of all △, $C_2^{\approx}$ of all ○, and
$C_3^{\approx}$ of all ϟ

Figure 2: A reference clustering Cl^{ref} and an approximated clustering $Cl^{\approx}$.

We will now introduce a metric distance function between a partitioning approximated clustering and a reference clustering which is based on the *minimal matching distance*.

The Minimal Matching Distance. A distance measure on sets of complex objects that demonstrates to be suitable for defining similarity between two partitioning clusterings is based on the *minimal weight perfect matching* of sets. This well known graph problem can be applied here. Let us first introduce some notations.

Definition 5 (weighted complete bipartite graph).
A Graph $G = (V, E)$ consists of a (finite) set of vertices V and a set of edges $E \subseteq V \times V$. A weighted graph is a graph $G = (V, E)$ together with a weight function $w\colon E \to IR$. A *bipartite graph* is a graph $G = (X \cup Y, E)$ where $X \cap Y = \emptyset$ and $E \subseteq X \times Y$. A bipartite graph $G = (X \cup Y, E)$ is called *complete* if $E = X \times Y$.

Definition 6 (perfect matching).
Given a bipartite graph $G = (X \cup Y, E)$ a *matching* of X to Y is a set of edges $M \subseteq E$ such that no two edges in M share an end point, i.e.

$$\forall (x_1, y_1), (x_2, y_2) \in M\colon x_1 = x_2 \Leftrightarrow y_1 = y_2$$

A matching M of X to Y is *maximal* if there is no matching M' of X to Y such that $|M| < |M'|$. A maximal matching M of X to Y is called a *complete* matching if $|M| = \min\{|X|,|Y|\}$. In the case $|X| = |Y|$ a complete matching is also called a *perfect* matching.

Definition 7 (minimum weight perfect matching).
Let $G = (X \cup Y, E)$ be a weighted bipartite graph together with a weight function $w\colon E \to IR$. We call a perfect matching M, a *minimum weight perfect matching*, iff for any other perfect matching M', the following inequality holds:

$$\sum_{(x,y) \in M} w(x, y) \leq \sum_{(x,y) \in M'} w(x, y)$$

In our application we build a complete bipartite graph $G = (Cl \cup Cl', E)$ between two clusterings Cl and Cl'. We set $Cl := Cl \times \{1\}$ and $Cl' := Cl' \times \{2\}$ to fulfill the property $Cl \cap Cl' = \emptyset$. The weight of each edge $((C_i, 1), (C'_j, 2)) \in Cl \times Cl'$ in this graph G is defined by the distance $d_\Delta(C_i, C'_j)$ introduced in the last section between the two clusters $C_i \in Cl$ and $C'_j \in Cl'$. A perfect matching is a subset $M \subseteq Cl \times Cl'$ that connects each cluster $C_i \in Cl$ to exactly one cluster $C'_j \in Cl'$ and vice versa. A minimal weight perfect matching is a matching with maximum cardinality and a minimum sum of weights of its

edges. Since a perfect matching can only be found for sets of equal cardinality, it is necessary to introduce weights for unmatched clusters when defining a distance measure between clusterings. The definition of the "minimal matching distance" is based on the definition of "permutated sets".

Definition 8 (permutation of a set).
Let S be any finite set of arbitrary elements. Then π is a mapping that assigns $s \in S$ a unique number $i \in \{1, ..., |S|\}$. This is denoted by $\pi(S) = (s_1, ..., s_{|S|})$. The set of all possible permutations of S is called $\Pi(S)$.

Definition 9 (minimal matching distance).
Let DB be a database and let $dist: 2^{DB} \times 2^{DB} \rightarrow IR$ be a distance function between two clusters. Let $Cl = \{C_1, ..., C_{|Cl|}\}$ and $Cl' = \{C'_1, ..., C'_{|Cl'|}\}$ be two clusterings. We assume w.l.o.g. $|Cl| \leq |Cl'|$. Furthermore, let $w: 2^{DB} \rightarrow IR$ be a weight function for the unmatched clusters. Then the *minimal matching distance* $d_{mm}^{dist, w}: 2^{2^{DB}} \times 2^{2^{DB}} \rightarrow IR$ is defined as follows:

$$d_{mm}^{dist, w}(Cl, Cl') = \min_{\pi \in \Pi(Cl')} \left(\sum_{i=1}^{|Cl|} dist(C_i, C'_{\pi(i)}) + \sum_{i=|Cl|+1}^{|Cl'|} w(C'_{\pi(i)}) \right)$$

The weight function $w: 2^{DB} \rightarrow IR$ provides the penalty given to every unassigned cluster of the clustering having larger cardinality. Let us note that this *minimal matching distance* is a specialization of the *netflow distance* which is introduced in [RM 01]. Though it was shown in [RM 01] that the netflow distance can be calculated in polynomial time, it is not obvious how to achieve it. Since we are only interested in a minimal matching distance it is sufficient to calculate a minimal weight perfect matching. Therefore, we propose to apply the method introduced by Kuhn [Kuh 55] and Munkres [Mun 57] which has a cubic runtime complexity w.r.t. the cardinality of the two clusterings, i.e. w.r.t. the number of found clusters.

Furthermore, the authors in [RM 01] show that the netflow distance is a metric if the distance function *dist* is a metric and the weight function meets the following two conditions for two clusters $C, C' \in 2^{DB}$:

- $w(C) > 0$
- $w(C) + w(C') \geq dist(C, C')$

Note that the symmetric set difference d_Δ is a metric and can be used as underlying distance function *dist* for the *minimal matching distance*. Furthermore, the unnormalized symmetric set difference allows us to define a meaningful weight function based on a dummy cluster.

Definition 10 (dummy cluster).
Let $V \subset 2^{DB}$ be a set of clusters, and let $C_0 \in 2^{DB} \backslash V$ be a "dummy" cluster. Then $w_{C_0}: V \rightarrow IR: \; w_{C_0}(C) = d_\Delta(C, C_0)$ denotes a family of weight functions based on a dummy cluster.

A good choice of the dummy cluster C_0 in our application is $\emptyset$ since the empty set is not included as an element in a clustering (cf. Definition 2), and, furthermore, each unmatched cluster C is penalized with a value $w_\emptyset(C) = d_\Delta(C, \emptyset)$ equal to its cardinality $|C|$. Thus the metric character of the *minimal matching distance* is satisfied. Furthermore, large clusters which cannot be matched are penalized more than small clusters which is a desired

property for an intuitive quality measure. Based on Definition 9, we can define our final quality criterion. We compare the costs for transforming an approximated clustering $Cl^{\approx}$ into a reference clustering Cl^{ref}, to the costs piling up when transforming $C^{\approx}$ first into $\varnothing$, i.e. a clustering consisting of no clusters, and then transforming $\varnothing$ into Cl^{ref} (cf. Figure 2).

Definition 11 (approximated partitioning clustering quality Q_{APC}).
Let $Cl^{\approx}$ be an approximated partitioning clustering and Cl^{ref} the corresponding reference clustering. The approximated partitioning clustering quality $Q_{APC}(Cl^{\approx}, Cl^{ref})$ is equal to 1 if $Cl^{ref} = Cl^{\approx} = \varnothing$ holds, else $Q_{APC}(Cl^{\approx}, Cl^{ref})$ is equal to

$$1 - \frac{d_{mm}^{d_\Delta, w_\varnothing}(Cl^{\approx}, Cl^{ref})}{d_{mm}^{d_\Delta, w_\varnothing}(Cl^{\approx}, \varnothing) + d_{mm}^{d_\Delta, w_\varnothing}(\varnothing, Cl^{ref})}$$

Note that our quality measure Q_{APC} is between 0 and 1. If $Cl^{\approx}$ and Cl^{ref} are identical, $Q_{APC}(Cl^{\approx}, Cl^{ref}) = 1$ holds. On the other hand, if the clusterings are not identical and the clusters from the two clusterings have no objects in common, i.e. $\forall C_j^{ref} \in Cl^{ref}, \forall C_i^{\approx} \in Cl^{\approx} : C_j^{ref} \cap C_i^{\approx} = \varnothing$ holds, $Q_{APC}(Cl^{\approx}, Cl^{ref})$ is equal to 0.

3.4 Similarity Measure for Hierarchical Clusterings

In this section, we present a quality measure for approximated hierarchical clusterings. As outlined in Section 3.1, a hierarchical clustering can be represented by a tree (cf. Definition 3). In order to define a meaningful quality measure for approximated hierarchical clusterings, we need a suitable distance measure for describing the similarity between two trees t^{ref} and $t^{\approx}$. Note that each node of the trees reflects a flat cluster, and the complete trees represent the entire hierarchical clusterings.

A common and successfully applied approach to measure the similarity between two trees is the degree-2 edit distance [ZWS 96]. It minimizes the number of edit operations necessary to transform one tree into the other using three basic operations, namely the insertion and deletion of a tree node and the change of a node label. Using these operations, we can define the degree-2 edit distance between two trees.

Definition 12 (cost of an edit sequence).
An edit operation e is the insertion, deletion or relabeling of a node in a tree t. Each edit operation e is assigned a non-negative cost $c(e)$. The cost $c(S)$ of a sequence of edit operations $S = \langle e_1, \ldots, e_m \rangle$ is defined as the sum of the cost of each edit operation, i.e. $c(S) = c(e_1) + \ldots + c(e_m)$.

Definition 13 (degree-2 edit distance).
The degree-2 edit distance is based on degree-2 edit sequences which consist only of insertions or deletions of nodes n with $degree(n) \leq 2$, or of relabelings. Then, the degree-2 edit distance between two trees t and t', $ED_2(t, t')$, is the minimum cost of all degree-2 edit sequences that transform t into t' or vice versa:

$$ED_2(t, t') = min\{c(S) \mid S \text{ is a degree-2 edit sequence transforming } t \text{ into } t'\}.$$

It is important to note that the degree-2 edit distance is well defined. Two trees can always be transformed into each other using only degree-2 edit operations. This is true because it is possible to construct any tree using only degree-2 edit operations. As the same is

true for the deletion of an entire tree, it is always possible to delete t completely and then build t' from scratch resulting in a distance value for this pair of trees. In [ZWS 96] Zhang, Wang, and Shasha presented an algorithm which computes the degree-2 edit distance in $O(|t| \cdot |t'| \cdot D)$, where D denotes the maximum fanout of t and t', and $|t|$ and $|t'|$ denote the number of tree nodes of t and t'.

We propose to set the cost $c(e)$ for each insert and delete operation e to 1. Furthermore, we propose to use the *normalized symmetric set difference* d_Δ^{norm} as introduced in Definition 4 to weight the relabeling cost. Using the normalized version allows us to define a well-balanced trade-off between the relabeling cost and the other edit operations, i.e. the insert and delete operations. Based on these costs, we can define our final quality criterion. We compare the costs for transforming an approximated hierarchical clustering $Cl^{\approx}$ modelled by a tree $t^{\approx}$ into a reference clustering Cl^{ref} modelled by a tree t^{ref}, to the costs piling up when transforming $t^{\approx}$ first into an "empty" tree t^{nil}, which does not represent any hierarchical clustering, and then transforming t^{nil} into t^{ref}.

Definition 14 (approximated hierarchical clustering quality Q_{AHC}).
Let t^{ref} be a tree representing a hierarchical reference clustering Cl^{ref}, and t^{nil} a tree consisting of no nodes at all, representing an empty clustering. Furthermore, let $t^{\approx}$ be a tree representing an approximated clustering $Cl^{\approx}$. Then, the approximated hierarchical clustering quality Q_{AHC} is defined as follows:

$$Q_{AHC}(Cl^{\approx}, Cl^{ref}) = 1 - \frac{ED_2(t^{\approx}, t^{ref})}{ED_2(t^{\approx}, t^{nil}) + ED_2(t^{nil}, t^{ref})}$$

As the *degree-2 edit distance* is a metric [ZWS 96], the approximated hierarchical clustering quality Q_{AHC} is between 0 and 1.

4 Conclusion

In this paper, we first motivated the need for objective distance measures allowing us to evaluate the quality of approximated clusterings to reference clusterings. First, we introduced a metric distance measure between single clusters based on the symmetric set difference. Two clusters are the more similar, the more objects they share. Next, we introduced an approximated partitioning clustering quality measure based on the minimal weight perfect matching of sets. The resulting distance measure, i.e. the minimal matching distance, tries to map each cluster of one clustering onto a unique cluster of the other clustering. The unmatched clusters of the clustering with the higher cardinality are penalized by a suitable weight function. By using the symmetric set difference as distance function between single clusters, and the minimal matching distance as distance function between clusterings, i.e. sets of clusters, we construct a meaningful quality measure for approximated partitioning clusterings. Finally, we introduced a quality measure for hierarchical approximative clusterings, where the structural differences between the trees reflecting the hierarchical clusterings are measured by the degree-2 edit distance. The differences between two single nodes, i.e. between two hierarchical clusters, are again measured by the symmetric set difference.

In our future work, we will use these quality measures for evaluating the effectiveness of new approximated clustering algorithms.

References

[ABKS 99] Ankerst M., Breunig M. M., Kriegel H.-P., Sander J.: "OPTICS: Ordering Points To Identify the Clustering Structure". Proc. ACM SIGMOD, Philadelphia, PA, 1999, pp. 49-60.

[Ben 04] Ben-David S.: "A Framework for Statistical Clustering with a Constant Time Approximation Algorithms for K-Median Clustering". COLT 2004, pp. 415-426.

[BL 04] Banerjee A., Langford J.: "An Objective Evaluation Criterion for Clustering". Proc. 10th ACM SIGKDD, Seattle, WA, USA, 2004, pp. 515-520.

[BKK+ 04] Brecheisen S., Kriegel H.-P., Kröger P., Pfeifle M.: "Visually Mining Through Cluster Hierarchies". Proc. SIAM Int. Conf. on Data Mining (SDM'04), Lake Buena Vista, FL, 2004, pp. 400-412.

[EM97] Eiter, T., Mannila, H.: "Distance Measures for Point Sets and Their Computation". Acta Informatica 34 (1997) 103–133.

[EKSX 96] Ester M., Kriegel H.-P., Sander J., Xu X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, 1996, pp. 226-231.

[FM 83] Fowlkes E., Mallows C.: "A method for comparing two hierarchical clusterings". Journal of American Statistical Association, 78, 1983, pp.553-569.

[GRG+ 99] Ganti V., Ramakrishnan R., Gehrke J., Powell A., French J.: "Clustering Large Datasets in Arbitrary Metric Spaces". Proc. 15th International Conference on Data Engineering, Sydney, Australia, 1999, pp. 502-511.

[Ha 00] Hanisch R. J.: "Distributed Data Systems and Services for Astronomy and the Space Sciences". In ASP Conf. Ser., Vol. 216, Astronomical Data Analysis Software and Systems IX, eds. N. Manset, C. Veillet, D. Crabtree, 2000.

[JKP 04] Januzaj E., Kriegel H.-P., Pfeifle M.: "DBDC: Density Based Distributed Clustering". Proc. 9th Int. Conf. on Extending Database Technology (EDBT 2004), Heraklion, Greece, 2004, pp. 88-105.

[JMF 99] Jain A. K., Murty M. N., Flynn P. J.: "Data Clustering: A Review". ACM Computing Surveys, Vol. 31, No. 3, Sep. 1999, pp. 265-323.

[Kuh 55] Kuhn, H.W.: "The Hungarian method for the assignment problem". Naval Research Logistics Quarterly 2 (1955) 83-97.

[LHY 04] Li Y., Han J., Yang J.: "Clustering Moving Objects". Proc. 10th ACM SIGKDD, Seattle, WA, USA, 2004, pp. 617-622.

[McQ 65] McQueen J.: "Some Methods for Classification and Analysis of Multivariate Observation". Proc. 5th Berkeley Symp. on Math. Statist. and Prob., Vol. 1, 1965, pp. 281-297.

[Mei 03] Meila M.: "Comparing Clusterings by the Variaition of Information". Proc. 16th Annual Conference on Computational Learning Theory (COLT'03), pp. 173-187.

[Muh 57] Munkres, J.: "Algorithms for the assignment and transportation problems". Journal of the SIAM 6 (1957) 32–38.

[PR 88] Pitt L., Reinke R. E.: "Criteria for Polynomial-Time (Conceptual) Clustering". Machine Learning, v.2 n.4, 1988, pp.371-396.

[RM 01] Ramon J., Bruynooghe M.: "A polynomial time computable metric between point sets". Acta Informatica 37 (2001) 765–780.

[SQL+ 03] Sander J., Qin X., Lu Z., Niu N., Kovarsky A.: "Automatic Extraction of Clusters from Hierarchical Clustering Representations". Proc. 7th PAKDD, Seoul, Korea, 2003, pp 75-87.

[ZWS 96] Zhang K., Wang J., Shasha D.: "On the editing distance between undirected acyclic graphs". International Journal of Foundations of Computer Science, 7(1):43–57, 1996.

[ZS 03] Zhou J., Sander S.: "Data Bubbles for Non-Vector Data: Speeding-up Hierarchical Clustering in Arbitrary Metric Spaces". 30th International Conference on Very Large Data Bases (VLDB), Berlin, Germany, 2003, pp. 452-463.