An Effective Evaluation Measure for Clustering on Evolving Data Streams

Hardy Kremer, Philipp Kranen, Timm Jansen, Thomas Seidl
RWTH Aachen University, Germany
lastname@cs.rwth-aachen.de

Albert Bifet, Geoff Holmes, Bernhard Pfahringer
University of Waikato Hamilton, New Zealand
{abifet,geoff,bernhard}@cs.waikato.ac.nz

ABSTRACT

Due to the ever growing presence of data streams, there has been a considerable amount of research on stream mining algorithms. While many algorithms have been introduced that tackle the problem of clustering on evolving data streams, hardly any attention has been paid to appropriate evaluation measures. Measures developed for static scenarios, namely structural measures and ground-truth-based measures, cannot correctly reflect errors attributable to emerging, splitting, or moving clusters. These situations are inherent to the streaming context due to the dynamic changes in the data distribution.

In this paper we develop a novel evaluation measure for stream clustering called Cluster Mapping Measure (CMM). CMM effectively indicates different types of errors by taking the important properties of evolving data streams into account. We show in extensive experiments on real and synthetic data that CMM is a robust measure for stream clustering evaluation.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Measurement, Experimentation

1. INTRODUCTION

Mining patterns from data streams is constantly gaining importance in many applications. Streams consist of data tuples that need to be processed as they arrive, and mining these streams is challenging since the data distribution underlying a stream can evolve significantly over time. Stream mining tasks are, for example, classification [4], association rule mining [30], or clustering [1, 8, 23]. In stream clustering, a clustering is constantly adapted to reflect changes in the observed stream. Besides dealing with evolving distributions, stream clustering algorithms have to meet several technical requirements, including limited time, limited memory, and processing the stream in a single pass. A multitude of stream clustering algorithms has been proposed in the literature that satisfy these requirements.

Of major importance is the quality of the resulting clusterings, which can be measured by evaluation measures, also termed criteria, indices, validation measures, or validation indices. Research on evaluation measures for static datasets, i.e. traditional clustering without the streaming context, looks back at a history of more than thirty years. While clustering itself is commonly accepted as a difficult and subjective task [21], the validation of clustering results is even described as the most difficult and frustrating part of cluster analysis [20]. This holds for evaluation no matter whether or not a ground truth is available against which to compare the clustering result. The usage of a ground truth can strictly categorize the proposed measures into internal and external measures [7, 25, 38, 41]: Internal measures consider only the structure and properties of the clusters, e.g. their compactness or the distance between them. External measures compare the resulting clustering against a ground truth, e.g. punishing an algorithm for putting objects from different ground truth clusters into a single cluster.

The MOA Framework [5, 24, 29] is an open source benchmarking software for data stream mining, which contains a set of stream clustering algorithms from the literature as well as an extensive collection of evaluation measures. The experience from implementing and using MOA showed two major disadvantages of existing evaluation measures: First, none can properly handle the peculiarities of evolving data streams such as overlapping due to merging or drifting clusters or noisy data streams. As a consequence the measures cannot effectively reflect the occurring errors. Second, the vast majority of evaluation measures achieve suboptimal results even if the ground truth clustering is tested. The Cluster Mapping Measure (CMM) proposed in this paper overcomes these shortcomings and enables effective evaluation of clustering results on evolving streams.

2. RELATED WORK

We first review existing work on evaluation measures for clustering on static data sets. After that we discuss related work on stream clustering algorithms and identify the evaluation measures employed in the individual approaches.

As mentioned in Sec. 1, evaluation measures can be categorized into internal and external measures, depending on whether or not they employ a ground truth clustering for comparison. A different categorization additionally identi-

KDD'11, August 21–24, 2011, San Diego, California, USA.

fies so called relative measures, comparing the single partitions within a clustering result. However, this group is less recognized and solutions are often specialized to a specific domain [36, 42]; we focus on internal and external measures. Table 1 lists (without claiming completeness) a collection of internal and external measures from the literature with references to formulae and more detailed description. Apart from these, other measures exist that deal with fuzzy clustering [6] or focus on finding the best number of clusters [37, 40]. For readability and space reasons we reduced the amount of measures used in our experiments to the ones highlighted in gray. Our choice of internal measures is based on a study of thirty measures in [28]; the external measures resulted as best choices in three recent studies [7, 38, 41].

Internal measures	External measures
Gamma [2]	Rand statistic [41, 32]
C Index [18]	Jaccard coefficient [12]
Point-Biserial [27]	Folkes and Mallow Index [12]
Log Likelihood [16]	Hubert Γ statistics [17]
Dunn's Index [11]	Minkowski score [7]
Tau [34]	Purity [43]
Tau A [18]	van Dongen criterion [39]
Tau C [18]	V-measure [35]
CDbw [14]	Completeness [35]
Ratio of Repetition [18]	Homogeneity [35]
Sum squared dists SSQ [15]	Variation of information [26]
Adj. Ratio of Clustering [18]	Mutual information [10]
Fagan's Index [18]	Class-based entropy [38]
Deviation Index [18]	Cluster-based entropy [43]
Z-Score Index [18]	Precision [33]
D Index [18]	Recall [33]
Silhouette coefficient [22]	F-measure [33]

Table 1: Internal and external evaluation measures.

Stream clustering algorithms can be categorized from different perspectives, e.g. whether convex or arbitrary shapes are found. Convex stream clustering approaches are based on a k-center clustering [1, 23, 31]. Approaches for arbitrary shaped clusters use kernels [19], fractal dimensions [3], or density based methods [8, 9]. In the evaluation of these papers, the used measures are those for static data sets, most often Entropy (Purity), Precision, or the F-measure. In convex approaches, which are the most prevalent in the literature, often also the sum of squared distances is used (e.g., [1, 31]), i.e. the compactness of spherical clusters is measured, which does not testify the quality of a clustering.

Most stream clustering algorithms use an internal representation, called *micro clustering*, that is constantly updated. Only when required, the output (macro) clusterings are generated by an offline component, e.g. k-means.

In summary, we find that despite the amount of work on developing evaluation measures as well as developing stream clustering algorithms, no attempt has been made to meet the requirements of both tasks.

3. PRELIMINARIES

In this section we provide formal definitions used throughout the paper. We concentrate on spherical clusters, which constitute the most prevalent model. It is e.g. also used in the online components of density based algorithm [9].

An object arriving on a data stream is represented by a tuple of attribute values. We define a stream as follows:

DEFINITION 1. ***Stream.*** *A stream $\mathcal{S} = \{o_1, o_2, \ldots\}$ is an infinite sequence of objects o_i, where each object is a d-dim. tuple of attribute values $o_i = (o_{i1}, \ldots, o_{id})$ with $o_{ij} \in \mathbb{R}$.*

In order to give more influence to recent data, objects are assigned an age-depending weight. We exemplarily define the weight using an exponential decay function as in [1]. Other weighting functions assign a binary weight (sliding window) using either the age or the cardinality for thresholding.

DEFINITION 2. ***Weight.*** *Let t_{now} be the current time and t_o the arrival time of object o with $t_o \leq t_{now}$. Then the weight of o is $w(o) = \beta^{-\lambda \cdot (t_{now} - t_o)}$.*

The parameters β and λ control the form of the aging function, e.g. if $\beta = 2$ then $1/\lambda$ is the half life of o. Apart from the complete stream $\mathcal{S}$ we often only consider a subset of the most recent objects that we call horizon.

DEFINITION 3. ***Horizon.*** *The horizon $\mathcal{H}$ for a stream $\mathcal{S}$ and threshold ξ is defined as $\mathcal{H} = \{o \in \mathcal{S} | w(o) \geq \xi\}$.*

To employ a consistent terminology we also formally define clustering and ground truth.

DEFINITION 4. ***Clustering.*** *A clustering algorithm takes a set $\mathcal{O} = \{o_1, \ldots, o_n\}$ of objects as input and returns a cluster set $\mathcal{C} = \{C_1, \ldots, C_k, C_\emptyset\}$. $o \in C_i$ implies that o lies within the cluster boundary of C_i and $C_\emptyset$ contains all unassigned objects. Objects may fall into several clusters.*

DEFINITION 5. ***Ground truth.*** *For a given object set $\mathcal{O} = \{o_1, \ldots, o_n\}$ a ground truth $\mathcal{CL} = \{Cl_1, \ldots, Cl_l\}$ is a partitioning of $\mathcal{O}$ with $\forall i \neq j : Cl_i \cap Cl_j = \emptyset$ and $\bigcup_{i=1}^{l} Cl_i = \mathcal{O}$. The partitions Cl_i are called classes and $Cl(o)$ is the class of o. In the presence of noise Cl_{noise} is the noise class, $\mathcal{CL}^+ = \mathcal{CL} \cup \{Cl_{noise}\}$ and $\mathcal{O}^+ = \mathcal{O} \cup Cl_{noise}$.*
We define the ground truth cluster Cl_i° as the smallest cluster that contains all objects from Cl_i within its boundary: $\forall o \in Cl_i$: $o \in Cl_i^\circ$.

The ground truth cluster resulting from a class may contain points from other classes as well, i.e. $Cl_i^\circ \supseteq Cl_i$, and two clusters resulting from two different ground truth classes can overlap, i.e. they are not necessarily disjoint since objects may fall into the boundaries of other classes.

4. CLUSTER MAPPING MEASURE

In stream processing, data mining algorithms have to take the special properties of the scenario into account. Likewise, for a reasonable comparison of the algorithms, evaluation measures should also consider these special circumstances:

1. **Aging / decay** Dealing with this property is probably the simplest task, because faults caused by a clustering algorithm can be weighted by the influence of the corresponding points (cf. Def. 2).

2. **Missed points** Moving clusters yield errors for missed points. These errors should reflect the seriousness, e.g. how close the point is to its actual cluster.

3. **Misplaced points** Evolution, merging, and splitting of clusters yield overlapping clusters and thereby easily misplaced points. A measure that punishes these misplaced points equally to misplaced points laying outside of any overlapping region does not account for the special circumstances of evolving streams.

4. **Noise** Including noise in a found cluster is often inevitable in the model of the clustering algorithm and should be accounted for by an effective measure.

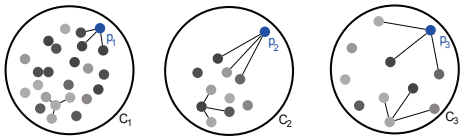


Figure 1: Point connectivity in clusters. p_1 and p_3 have stronger connections to their clusters than p_2.

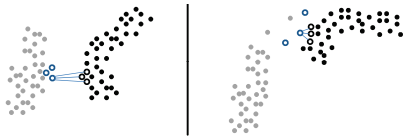


Figure 2: The errors due to misplaced points (blue circles) is less severe in the right example.

Summarizing these properties, three fault cases can be identified that have to be considered in depth, namely missed points, misplaced points, and noise inclusion. The penalty for such errors of stream clustering algorithms should reflect their seriousness and take the age of the points as well as the clustering model into account. CMM is a normalized sum of the penalties for occurring errors that accounts for all aspects mentioned above. Two important prerequisites for the computation are a notion of how well an object fits into a cluster and a mapping from found clusters to ground truth classes. We introduce these prerequisites and CMM in the following section and analyze its behavior in the fault scenarios in Sec. 5.

4.1 Connectivity

A central concept used throughout CMM is the connectivity between points and clusters. This connectivity states how well the point is connected to the cluster, i.e. how it fits the distribution of the cluster in comparison to the other points. We define the connectivity as a distance based concept that ranges from 0 to 1, where 0 indicates no connectivity and 1 indicates a strong connectivity. As a prerequisite we define the average k-neighborhood distance for points and clusters. k is a locality parameter, and we show in Sec. 6 that it has only marginal influence on CMM effectiveness.

DEFINITION 6. ***average k-neighborhood distance.*** *The k-neighborhood (knh) of a point p in a cluster C_i are the k closest points, i.e. $knh(p, C_i) \subseteq C_i \setminus p$ with $|knh(p, C_i)| \leq k$, and for objects $o_{in} \in knh(p, C_i)$ and $o_{out} \in C_i \setminus knh(p, C_i)$ it holds $\forall o_{in} \forall o_{out} : dist(p, o_{out}) \geq dist(p, o_{in})$. The average distance of p to its k neighbors in C_i is then*

$$knhDist(p, C_i) = \frac{1}{k} \sum_{o \in knh(p, C_i)} dist(p, o)$$

and the average distance for a Cluster C_i is

$$knhDist(C_i) = \frac{1}{|C_i|} \sum_{p \in C_i} knhDist(p, C_i)$$

Based on this distance we define connectivity as follows:

DEFINITION 7. ***Point connectivity.*** *The connectivity of a point p to a Cluster C_i is defined as*

$$con(p, C_i) = \begin{cases} 1 & if \quad knhDist(p, C_i) < knhDist(C_i) \\ 0 & if \quad C_i = \emptyset \\ \frac{knhDist(C_i)}{knhDist(p, C_i)} & else \end{cases}$$

Fig. 1 shows examples for connectivity scenarios for spherical clusters using $k = 3$. In the left case the points are densely and rather equally distributed within the cluster. Point p_1 has a strong connectivity to cluster C_1, since its average neighborhood distance is similar to that of C_1. In C_2, the points are not equally distributed. Point p_2 is not strongly connected, since its average neighborhood distance is larger than the average of C_2. In the right example, p_3 also has a comparably large average neighborhood distance, but it is still strongly connected to C_3, because C_3 is less dense.

Fig. 2 illustrates the connectivity of points from the gray cluster (blue circles) to the black cluster. Misplacing gray points in the black cluster seems more severe in the left example, since they are strongly connected to their own (gray) cluster and hardly connected to the black cluster (cf. blue circles). In contrast, on the right they have a stronger connection to the black cluster making the misplacement error intuitively less severe.

4.2 Cluster mapping

To decide whether a point is misplaced we need a mapping from the clusters returned by a stream clustering algorithm to ground truth classes. Before we give a formal definition of the employed mapping we will discuss an example.

As described in Sec. 3 clusters Cl_j° resulting from the ground truth classes (cf. Def. 5) may overlap. Fig. 3(a) shows an example. The gray points belong to class Cl_1, the black points to class Cl_2. The solid circles are the cluster boundaries resulting from a spherical clustering model, i.e. Cl_1° and Cl_2°, and there are black points that fall into the boundary of the gray cluster and vice versa. The figure demonstrates that on evolving data streams a simple majority voting for mapping found clusters to ground truth clusters is not feasible. The core problem are clusters of different densities, frequently occurring in streaming scenarios due to novelty or disappearance of clusters. In the figure, the black cluster is dense and the gray cluster is still emerging, i.e. there are only 10 objects present in the gray concept. The overlapping due to the convex model yields a majority of the black class even in the ground truth cluster Cl_1° of the gray class, i.e. a majority voting would map both C_1 and Cl_1° to the wrong (black) class due to the dominance of the black objects. A majority voting cannot obtain a correct mapping in these scenarios because its decision is based on a single class's objects in a cluster, ignoring the other classes' information that often indicate novel or fading clusters.

To solve this issue we map clusters C_i to ground truth classes Cl_j according to the similarity of C_i's and Cl_j°'s class distributions, i.e. we take the clustering model into account. We define these distributions of classes inside a cluster as frequency histograms.

DEFINITION 8. ***Class distribution.*** *For a given clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and ground truth $\mathcal{CL} = \{Cl_1, \ldots, Cl_l\}$, the histogram of the class distribution in cluster C_i is*

$$\rho(C_i) = (|C_i \cap Cl_1|, \ldots, |C_i \cap Cl_l|)$$

$\rho(C_i)_a$ is the a-th component of $\rho(C_i)$ representing the frequency of class Cl_a in cluster C_i.

Def. 8 analogously applies to ground truth clusters Cl_j° that result from classes Cl_j. The right part of Fig. 3 illustrates the class distributions for the shown clusters.

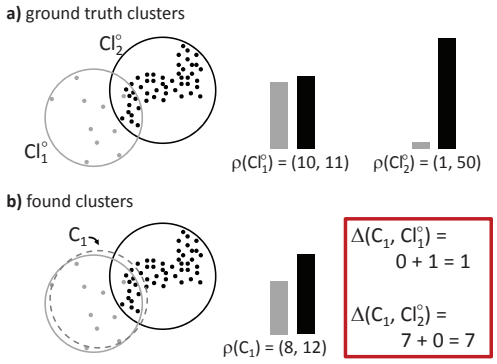


Figure 3: Mapping from found clusters to ground truth clusters is based on class distributions. Mapping clusters based on majority voting cannot recognize emerging or disappearing clusters.

We map clusters based on class distribution similarity. Concretely, we map a cluster C_i to the ground truth class Cl_j, whose ground truth cluster Cl°_j covers the majority of C_i's class frequencies.

DEFINITION 9. ***Cluster mapping.*** *For a given clustering $\mathcal{C} = \{C_1, \ldots, C_k, C_\emptyset\}$ and ground truth $\mathcal{CL} = \{Cl_1, \ldots, Cl_l\}$,*

$$\Delta(C_i, Cl^\circ_j) = \sum_{a=1}^{l} \max\{0, \rho(C_i)_a - \rho(Cl^\circ_j)_a\}$$

is the total surplus of objects from classes $Cl_1, \ldots, Cl_l$ in C_i compared to Cl°_j, where the surplus is the number of class objects from Cl_i not covered by the class distribution of the ground truth cluster Cl°_j. C_i is then mapped to the class

$$map(C_i) = \begin{cases} \underset{Cl_j \in \mathcal{CL}}{\text{argmin}} \{\Delta(C_i, Cl^\circ_j)\} & \text{if } \forall Cl_j : \Delta(C_i, Cl^\circ_j) > 0 \\ \emptyset & \text{if } C_i = C_\emptyset \\ \underset{\substack{Cl_j \in \mathcal{CL} \\ \Delta(C_i, Cl^\circ_j)=0}}{\text{argmax}} \{|C_i \cap Cl^\circ_j|\} & \text{else} \end{cases}$$

The first and most important case ensures that for a set of ground truth clusters, C_i is mapped to the ground truth class that induces the least surplus on class frequencies. If we would also consider negative differences, as e.g. in L_p-norms, micro clusters would always be mapped to the smallest ground truth class. Employing a majority voting is not generally feasible in the streaming context as explained above. The second case ensures that the unassigned object set $C_\emptyset$ is not mapped to any class, enabling a correct penalty for missed non-noise points (cf. the analysis in Sec. 5.1). The third case arises if C_i completely fits into one or more ground truth clusters; in this special case, a majority voting is applied. If a ground truth cluster Cl°_i falls completely into another cluster Cl°_j, i.e. $\forall a : \rho(Cl^\circ_i)_a \leq \rho(Cl^\circ_j)_a$, C_i is mapped to the larger cluster's class; such clusters are inseparable, as we discuss in Sec. 5.3. If we use this mapping in Fig. 3, the found cluster C_1 in (b) as well as the ground truth clusters Cl°_i in (a) are correctly mapped.

4.3 Cluster Mapping Measure

With the mapping from found clusters to ground truth classes, we can now determine the set $\mathcal{F} \subseteq \mathcal{O}^+$ of points that cause *faults*, i.e. missed points, misplaced points, or included noise points.

DEFINITION 10. ***Fault set.*** *For $\mathcal{O}^+ = \mathcal{O} \cup Cl_{noise}$, $\mathcal{CL}^+ = \mathcal{CL} \cup \{Cl_{noise}\}$, and $\mathcal{C} = \{C_1, \ldots, C_k, C_\emptyset\}$, the set of objects mapped to a false class is*

$$\mathcal{F} = \{o \in \mathcal{O}^+ | \exists C_i : o \in C_i \wedge map(C_i) \neq Cl(o)\}$$

Obviously, a single point o can cause several faults. The set of clusters in which o causes a fault is

$$faultClu(o) = \{C_i \in \mathcal{C} | o \in C_i \wedge map(C_i) \neq Cl(o)\}$$

The penalty for a fault point depends on the connectivity to both its true class and the assigned class.

DEFINITION 11. ***Penalty.*** *Let o be an object from the fault set and $C_i \in faultClu(o)$. The penalty for o is*

$$pen(o, C_i) = con(o, Cl(o)) \cdot (1 - con(o, map(C_i)))$$

*The **overall penalty** for a fault point o w.r.t. all found clusters $C_i \in \mathcal{C}$ is defined as*

$$pen(o, \mathcal{C}) = \max_{C_i \in faultClu(o)} \{pen(o, C_i)\}$$

The single penalty function can handle all three fault types, and we detail on this in Sec. 5.1. To sustain comparability between points, we only count a fault point's most serious fault in the overall penalty.

Now we can define the CMM, which indicates how different a given clustering is from a given ground truth. CMM is a normalized sum of the penalties: if no fault occurs CMM will be 1 and 0 indicates maximal error.

DEFINITION 12. ***CMM.*** *Given an object set $\mathcal{O}^+ = \mathcal{O} \cup Cl_{noise}$, a ground truth $\mathcal{CL}^+ = \mathcal{CL} \cup \{Cl_{noise}\}$, a clustering $\mathcal{C} = \{C_1, \ldots, C_k, C_\emptyset\}$, and the fault set $\mathcal{F} \subseteq \mathcal{O}^+$, the **Cluster Mapping Measure** between $\mathcal{C}$ and $\mathcal{CL}^+$ is defined using the point weight $w(o)$ and the overall penalty from Def. 11 as*

$$CMM(\mathcal{C}, \mathcal{CL}) = 1 - \frac{\sum_{o \in \mathcal{F}} w(o) \cdot pen(o, \mathcal{C})}{\sum_{o \in \mathcal{F}} w(o) \cdot con(o, Cl(o))}$$

and if $\mathcal{F} = \emptyset$, then $CMM(\mathcal{C}, \mathcal{CL}) = 1$.

5. ANALYSIS & REFINEMENT

In this section we first analyze how CMM handles different fault types caused by evolving streams. In Sec. 5.2 we refine CMM such that it is robust against errors occurring due to the clustering model and that it can measure how well a clustering reflects the ground truth's main concepts. Sec. 5.3 introduces a structure analysis for data sets.

5.1 Missed, Misplaced, and Noise inclusion

In this section we discuss how the penalties differ for the three fault types and why these faults are correctly reflected.

An object $o \in \mathcal{CL}$ is missed if it is unassigned and is not a noise object, i.e. $o \in C_\emptyset \wedge o \notin CL_{noise}$. The set of these objects is called $\mathcal{F}_{missed}$. The connectivity of o to its class $Cl(o)$ gives an idea of how severe it is if an algorithm excludes o from a cluster mapped to $Cl(o)$. More precisely, if o is hardly connected to $Cl(o)$ then the exclusion of o is not as severe as an exclusion of a highly connected object. This is expressed by the first term in the penalty definition. The second term equals to 1 according to Def. 7, i.e.

$$o \in \mathcal{F}_{missed} \Rightarrow pen(o, C_\emptyset) = con(o, Cl(o)) \cdot \underbrace{(1 - con(o, \emptyset))}_{=1}$$

If a noise object o_{noise} is assigned to a cluster C_i, this noise inclusion is considered a fault, and the corresponding set is called $\mathcal{F}_{noise}$. We want to penalize noise inclusion according to the connection between o_{noise} and $map(C_i)$, i.e. if $con(o_{noise}, map(C_i))$ has a high value then we want to assign a low error and vice versa. This is expressed by the second term in the penalty definition, the first term is close to 1 due the random distribution of noise, i.e. $o \in \mathcal{F}_{noise} \Rightarrow$

$$pen(o, C_i) = \underbrace{con(o, Cl_{noise})}_{\approx 1 \text{ for noise}} \cdot (1 - con(o, map(C_i)))$$

For a misplacement error assume an object o from class $Cl(o)$ is assigned to cluster C_i and C_i is mapped to another class, i.e. $map(C_i) \neq Cl(o)$. Misplaced objects are summarized in $\mathcal{F}_{misplaced}$. In this case we want to consider both the connection of o to its true class $Cl(o)$ and its connection to the wrongly assigned class $map(C_i)$. More precisely, a high value for $con(o, Cl(o))$ yields a high error and a low value for $con(o, map(C_i))$ yields a high error as well. The penalty in CMM combines both aspects by multiplication.

A better understanding of clusterings and their evaluation can be achieved by constraining CMM (cf. Def. 12) to one of the subsets $\mathcal{F}_{missed}$, $\mathcal{F}_{noise}$, or $\mathcal{F}_{misplaced}$ (cf. Sec. 6).

5.2 Model error and concept detection

If one wants to evaluate the goodness of a given clustering algorithm, the corresponding approach, i.e. its underlying clustering model, is of importance. For example, given the data sets from Fig. 2 or 3, a spherical clustering algorithm can never reach a perfect result, since even the ground truth clusters contain objects from different classes yielding misplacement errors. How meaningful is a measure that tells you an algorithm has 70% performance if you don't know whether it's model even allows it to perform better?

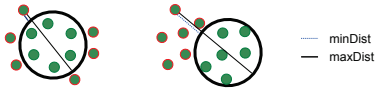
CMM shall measure how well an algorithm solved the clustering task with respect to its approach. To this end it is checked for each object, whether it causes an error in the corresponding ground truth clusters Cl_j° (cf. Def. 5). If an error occurs also in the ground truth, it is called an error by model and the object is added to the set $\mathcal{F}_{model}$. These errors by model are then excluded in the CMM computation. This way the ground truth always yields a perfect result and so can an algorithm.

Any stream clustering algorithm has to adhere to memory constraints for its internal representation, i.e. the total number of clusters or micro clusters is limited. Due to aging of objects, drifting clusters, and newly emerging clusters the algorithm constantly has to update its representation and decide on the position and size of its clusters. It is more important to reflect the main concepts of the current data distribution rather than spending limited resources to represent some concepts very detailed and missing other concepts in exchange. In other words, for two given clusterings with equally many missed objects, the one that better covers the concepts is preferable (cf. Fig. 4). To operationalize this aspect we decrease the penalty for missed points with respect to the severity of the fault. Formally, we redefine the overall penalty for this fault type. For an object $o \in \mathcal{F}_{missed}$ we test for all clusters C_i with $map(C_i) = Cl(o)$ the relative distance of o to the cluster boundary of C_i, i.e. $o \in \mathcal{F}_{missed} \Rightarrow$

$$pen(o, \mathcal{C}) = con(o, Cl(o)) \cdot \max_{\substack{C_i \in \mathcal{C} \\ map(C_i) = Cl(o)}} \{1 - e^{-\frac{minDist(o, C_i)}{maxDist(o, C_i)}}\}$$

The second term simplifies to $1 - e^{-\frac{d-r}{d+r}}$ for spherical clusters, where d is the distance of o to C_i's center and r its radius.

Figure 4: The concept is better recognized on the left.



5.3 Structure analysis

To analyze the structure of a given data set, we define the connectivity between two classes, which is based on the same concept as Def. 7. Intuitively, the connection from class Cl_i to class Cl_j is high, if several points that have a strong connection within Cl_i show a strong connection to Cl_j at the same time. The definition below uses the same locality parameter k as Def. 6.

DEFINITION 13. ***Class connectivity.*** *For a locality parameter k the connectivity from class Cl_i to class Cl_j is*

$$con_{Cl_i \to Cl_j} = \max_{\substack{KM \subseteq Cl_i \\ |KM| = k}} \left\{ \frac{1}{k} \sum_{p \in KM} con(p, Cl_i) \cdot con(p, Cl_j) \right\}.$$

The symmetric connectivity between two classes is then $con(Cl_i, Cl_j) = \min\{con_{Cl_i \to Cl_j}, con_{Cl_j \to Cl_i}\}$

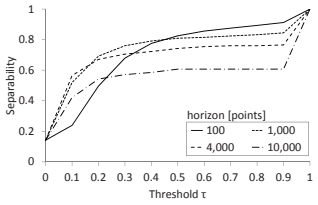
The above definition considers the k points within Cl_i that maximize the product of connectivity to both classes. The blue objects in Fig. 2 illustrate the connectivity between the two classes gray and black. In both cases the two classes are not strongly connected.

If two ground truth classes have a strong connectivity, the two classes are difficult to separate from each other. To avoid misplacement penalties between such classes, we merge classes who's connectivity according to Def. 13 exceeds a threshold τ and assign their points the same label. Analyzing the structure of a given ground truth $\mathcal{CL} = \{Cl_1, \ldots, Cl_l\}$ hence yields a reduced set of classes $\mathcal{CL}' = \{Cl'_1, \ldots, Cl'_h\}$ with $h \leq l$. As an indicator of how well the classes in a given ground truth $\mathcal{CL}^+$ are separable we define:

$$Separability(\mathcal{CL}^+) = h/l$$

In Fig. 5 we demonstrate for different horizons how the Separability is influenced by different τ values.

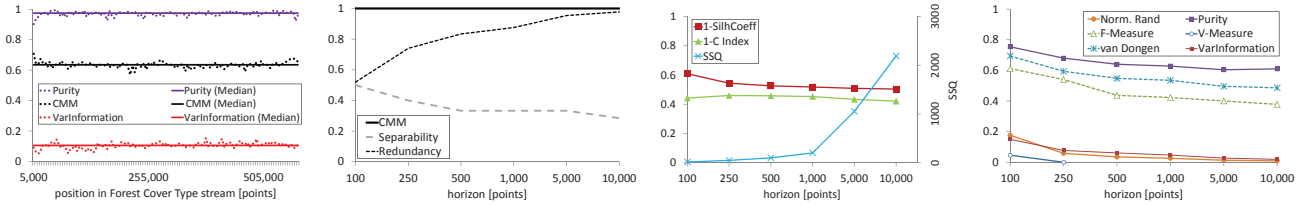
Figure 5: Evaluation of different τ values.



For the structure analysis in the experiments in Sec. 6, we use $\tau = 0.5$. To complete the structure analysis we define a second indicator, counting the percentage of objects represented redundantly, i.e. covered by several ground truth clusters Cl_i°:

$$Red(\mathcal{CL}^+) = |\{o \in \mathcal{O}^+ | o \in Cl_i^\circ \wedge o \in Cl_j^\circ \wedge i \neq j\}| / |\mathcal{O}^+|$$

This indicator allows for a more in depth analysis of the different measures' behavior in the empirical evaluation.



(a) CluStream clustering (b) CMM & structure analysis (c) internal measures (d) external measures

Figure 6: 10-dim. *Forest Cover Type* stream: (a): Illustration of the individual evaluations for a real stream clustering and the resulting medians. (b-d): Varying horizon on ground truth clusterings.

6. EXPERIMENTS

We evaluate CMM in comparison to evaluation measures introduced in Sec. 2. For repeatability, we integrated CMM into MOA (cf. Sec. 7). First of all, we describe our evaluation framework.

Stream generator. The generator's output is a stream of data points that have class labels and weights reflecting a point's age, depending on the used horizon. Since clusters move, a larger horizon yields more tail-like clusters; a small horizon yields more sphere-like clusters. Technically, the data points are obtained by generating clusters, which have a center and a maximal radius and which move through a unit cube ($[0,1]^d$) into a random direction. At the cube boundaries, the clusters bounce off. At specific intervals all cluster centers are shifted by 0.01 in their corresponding direction and points are equally drawn from each generating cluster. On average, the number of points in each cluster is equal. Our stream generator has the following parameters: the number of clusters moving through the dataspace, the number of generated points, the shift interval, the cluster radius, the dimensionality, and the noise degree.

Real world data streams. We use the 10 continuous attributes of *Forest Cover Type*[1], consisting of 581,012 objects and 7 classes, and we use the 34 continuous attributes of *Network Intrusion*[2], consisting of 4,898,431 objects and 5 classes. This variant was also used in [1, 8].

Cluster generator. We evaluate measures in specific clustering error scenarios, which are obtained by generating clusterings out of the synthetic stream that reflect a desired error level. We create cluster join, radius decrease, and cluster remove errors and the cluster boundaries are determined by a technique from [13]. The cluster join error simulates that two classes are covered by a single cluster by joining pairs of nearby, non-overlapping clusters; e.g., an error level of 0.4 indicates that all pairwise clusters are joint whose minimal boundary distance is below 40% of the minimum of their cluster radii. By this, we achieve a more realistic scenario without joining of far apart clusters. The radius decrease error states that the generated clusters have a radius that is smaller than the radius of the corresponding ground truth cluster. An error level of 1 states that the radius of the error cluster is 0; for an error level of 0 the two radii are equal. The cluster remove error denotes how many clusters are missed by a clustering; for an error level of 0.4, 40% of the clusters are removed from the error-free clustering.

Setup. We compare CMM and its variants (cf. Sec. 5.1) with internal and external measures from the literature (cf. Sec. 2). In selected experiments, we provide the structural measures Separability and Redundancy (cf. Sec. 5.3). All measures, besides the Sum of Squared distances (SSQ), are normalized to $[0,1]$. For some measures the best value corresponds to 0; we reverse them and indicate this by "1-" in the plots. The values of SSQ are plotted on a secondary y-axis. By using clusterings from the cluster generator, we ensure an evaluation unbiased by possible incorrect outcomes of stream clustering algorithms. Moreover, we evaluate real clusterings of CluStream [1] and DenStream [8] using CMM. The CMM parameter k was preliminary fixed to 2; the choice of k only has marginal influence on CMM effectiveness, as we will later illustrate. Synthetic data is generated with the following settings: the dimensionality is 2, the number of points is $200,000$, the cluster number is 6, the cluster radius is 0.075, the shift interval is 100 points, and noise is equally distributed and fixed at 10%.

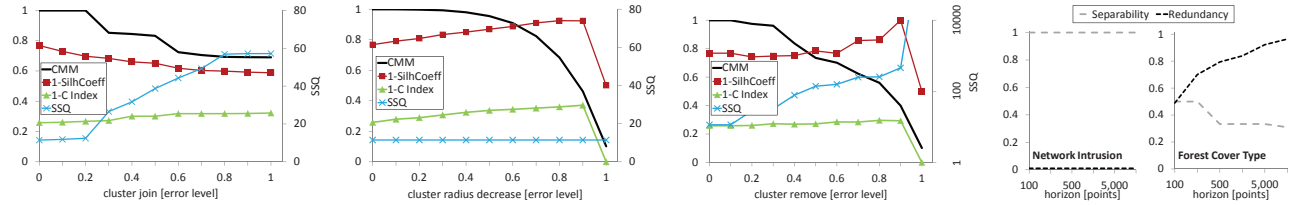
The horizon is defined by the number of included points, i.e. $h = |\mathcal{H}|$, and the evaluation frequency is the horizon. The values in all plots are the medians over all evaluations of the analyzed stream, as illustrated in Fig. 6(a) for three measures on CluStream clusterings of Forest Cover Type.

Varying horizon on real data. Fig. 6 shows experiments where we analyze the robustness of the measures under a varying horizon. We use error-free Forest Cover Type clusterings, i.e. the ground truth. By a varying horizon this ground truth changes; with greater horizons the cluster tails become longer, increasing the probability of overlapping clusters. This is confirmed by the two structural measures in Fig. 6(b): Redundancy rises while Separability (of clusters) drops. The comparison of CMM with the competing measures shows a clear picture: CMM (in Fig. 6(b)) achieves perfect quality that is stable over all settings. In contrast, both internal (Fig. 6(c)) and external measures (Fig. 6(d)) degrade with increasing horizons, showing that they cannot cope with higher redundancy and the worsened separability that occurs with larger horizons. More surprisingly, however, is that they also deliver low qualities for small horizons, indicating their general inability to express that the analyzed clusterings are error-free.

Error level. We analyze how good the measures can reflect the errors in generated clusterings.

We start with the internal measures in Fig. 7, where each graph corresponds to one error type. Comparing an external measure like CMM with internal measures is inherently unfair; however, since internal measures are often used in practice, we think a general overview of their performance in error scenarios is of interest. From the three plots, we can conclude the following: The C Index shows nearly no reaction; only extreme cases ($error \to 1$) for radius decrease and cluster remove errors are indicated through quality drops. The Silhouette Coefficient reflects the cluster join error, but can-

[1]http://archive.ics.uci.edu/ml/datasets/Covertype
[2]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html



(a) Cluster Join Error (b) Radius Decrease Error (c) Cluster Remove Error (d) Structure analysis

Figure 7: (a-c): CMM & internal measures under different error types with $h = 5,000$; (d): Structure analysis.

not reflect the other errors, where the corresponding values even rise. The SSQ, which is used in many stream clustering publications [1, 9], shows a good reaction on the cluster join and the cluster remove error; however, if we look at the corresponding maximal values (60 and 10,000) a practical use is questionable due to its unnormalized nature. CMM shows a very good performance on all scenarios, and a deeper analysis will be included in the next paragraph.

In Fig. 9 we compare CMM with the external measures using horizons of 5,000 and 10,000. We start with cluster join errors. Note that due to the technique used for creating this error (only nearby clusters are joined), the error level 1 does not indicate that all clusters are joined into a single cluster. Fig. 9(a) and 9(b) show that CMM reflects this error well, i.e. with an increasing number of joined clusters the quality constantly drops. We expect this to be caused by misplaced points, i.e. points that are assigned to wrong clusters, which is confirmed by $\text{CMM}_{misplaced}$. CMM_{noise} illustrates that when clusters are joined the resulting clusters tend to absorb noise points due to their larger sizes. The external measures in Fig. 9(c) also reflect the errors. However, a general problem of these measures are error-free settings (error=0) in which they show lower qualities (0.8-0.6). We suspect this to be caused by cluster overlap and it worsens with a larger horizon, as illustrated in Fig. 9(d). Considering that these measures are normalized to $[0, 1]$, this is a severe problem. In contrast, CMM shows maximal quality in error-free settings for both horizons and this also holds for the other error types in Fig. 9. The cluster radius decrease error is analyzed for CMM in Fig. 9(e) and 9(f). This error mostly results in missed points, which is shown by CMM_{missed} and thus by CMM. Here we also demonstrate the influence of the CMM refinement that better measures how well the main concepts of the data are reflected by a clustering (cf. Sec. 5.2). The figure shows that at lower error levels, CMM does not decrease as rapidly as the variants without refinement (green lines). This is intended, since the main concepts of the data are well represented at this stage and only close points are missed. From the six measures in Fig. 9(g), four can basically reflect the error; however, for lower error levels (≤ 0.4) three of the measures go up and the negative effects of larger horizons on all competing measures are again obvious in Fig. 9(h). Especially the bad performance of Purity is mentionable, a measure which is the main evaluation method in many stream clustering publications [8, 19]. The cluster remove error is analyzed in Fig. 9(i)-9(l). This error corresponds to removal of complete clusters and thus concepts; accordingly and in contrast to the radius decrease error this loss of concepts should be reflected by a constant decrease of the measured quality. This is achieved by CMM. A comparison to its variant without refinement shows that for this error type the refinement has no undesired effects; this refinement only affects the measurement of radius decrease errors. The external measures show a similar behavior as for the radius decrease error: only three reflect the error at all and their expressiveness strongly decreases for larger horizons.

Summarized, CMM can precisely reflect all errors. Most of the internal measures cannot reflect the errors at all, while some of the external measures reflect the errors but have serious problems in settings without errors or larger horizons.

CMM locality parameter k. In Fig. 8 we analyze the influence of k (cf. Sec. 4.1). We simulate a more realistic evaluation scenario by using synthetic clusterings with embedded errors (error level=0.5). For each error type, we tried 10 different ks ($k \in \{1, .., 10\}$) with 4 different horizons. On the left we plot the standard deviation of the CMM results w.r.t. the choice of k: CMM values for two of the error types are influenced by k for very small horizons; however, considering the value range of the CMM and the small standard deviations (< 0.009), these effects are negligible. This is also illustrated on the right, where we plot the exact CMM values with a horizon of $10,000$ for the individual ks on the range $[0.8, 0.98]$. Both figures show that k has only marginal influence for very small horizons and does not affect CMM for larger horizons.

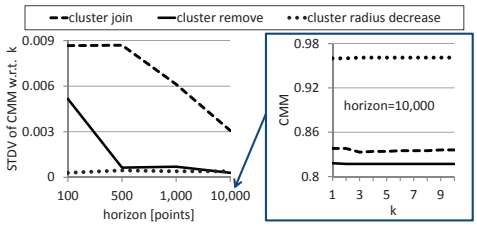
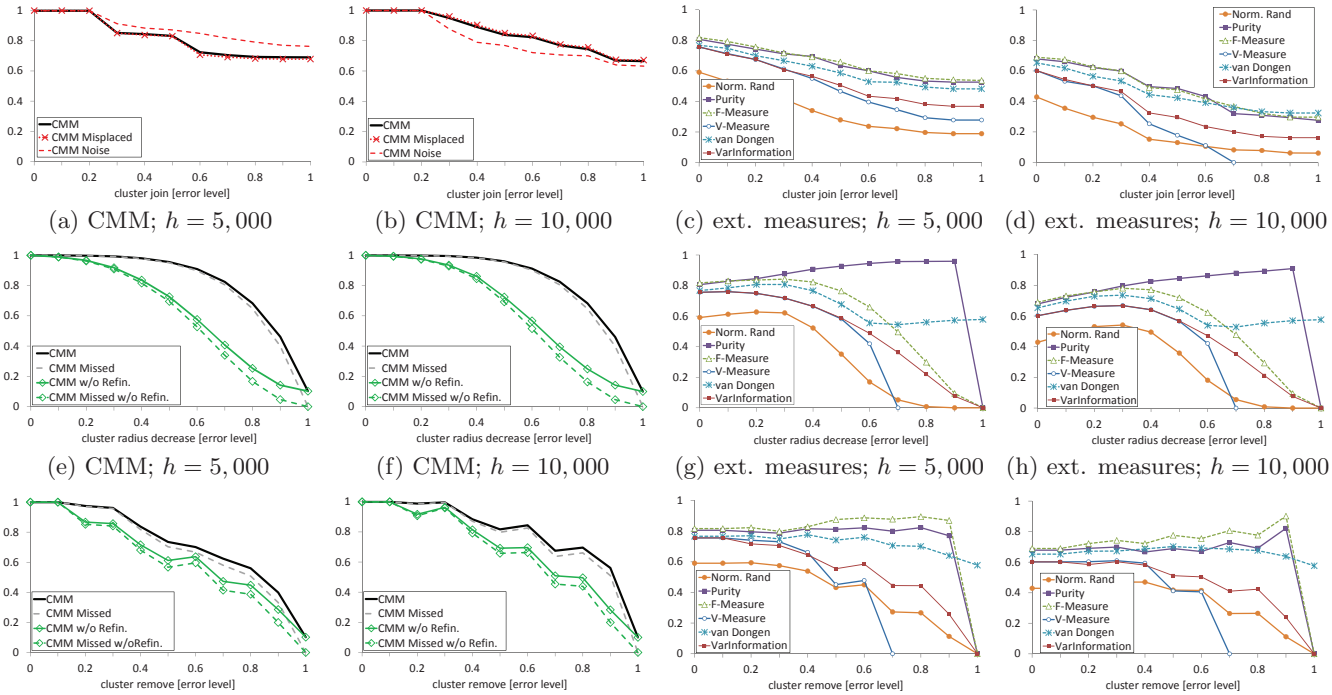


Figure 8: Marginal effects (std. deviation < 0.009) of varying k.

Real data and stream clustering algorithms. Finally, we analyze the measures in real application scenarios, in which real world data streams are clustered by actual stream clustering algorithms, as in this case CluStream [1] and DenStream [8]. In Fig. 10, we distinguish between macro clusterings, i.e. the output of the algorithms, and micro clusterings, i.e. the underlying internal representations constantly adapted by the algorithms. The macro clusterings are produced from the micro clusterings by an offline component. Comparability is ensured by using CluStream's offline component in CluStream and DenStream. We apply an horizon of 1,000. A structure analysis of the real world streams is shown in Fig. 7(d): Forest Cover Type has high redundancy and low separability; Network Intrusion has no redundancy and high separability, making the clusters easier to find.

We start with Network Intrusion. The results obtained by DenStream for the macro clustering in Fig 10(a) confirm

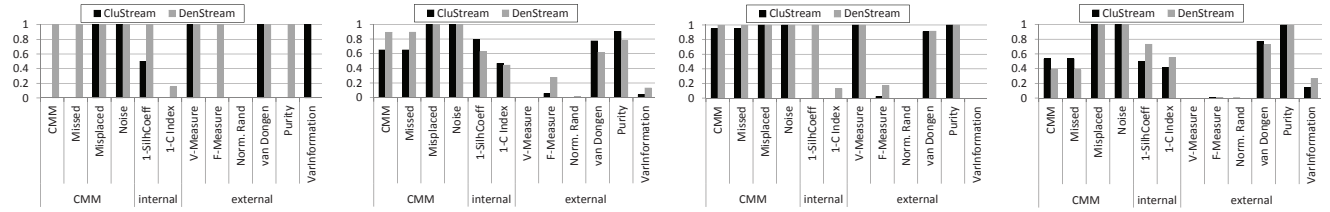


(a) CMM; $h = 5,000$ (b) CMM; $h = 10,000$ (c) ext. measures; $h = 5,000$ (d) ext. measures; $h = 10,000$

(e) CMM; $h = 5,000$ (f) CMM; $h = 10,000$ (g) ext. measures; $h = 5,000$ (h) ext. measures; $h = 10,000$

(i) CMM; $h = 5,000$ (j) CMM; $h = 10,000$ (k) ext. measures; $h = 5,000$ (l) ext. measures; $h = 10,000$

Figure 9: Evaluating CMM and external measures under different error types.



(a) Macro clusterings of *Network Intrusion* data. (b) Macro clusterings of *Forest Cover Type* data. (c) Micro clusterings of *Network Intrusion* data. (d) Micro clusterings of *Forest Cover Type* data.

Figure 10: Evaluation of *CluStream* and *DenStream* on real world data streams.

the conclusions from the structural analysis: nearly all evaluation measures indicate that high quality clusterings are generated. For CluStream, however, we can make a surprising conclusion. CMM and most of the other measures indicate that the found clusterings are of low quality. This is mostly caused by missed points, as CMM_{Missed} shows. In contrast, the evaluation of micro clusterings used by CluStream in Fig. 10(c) shows another picture: the micro clusterings are of high quality. We can conclude that CluStream correctly identifies the concepts in the data, but the used offline method that generates the macro clusters from the micro clusters is ineffective for this data type.

For Forest Cover Type in Fig. 10(b) and 10(d), we can conclude by CMM_{Missed} that DenStream covers nearly all of the concepts in the macro clusterings. CluStream, however, seems to miss more concepts of the data. For this dataset, the macro clusterings of both CluStream and DenStream are of higher quality than the micro clusterings.

Summarized, CMM is an effective measure for evaluation on real data streams and clusterings generated by stream clustering algorithms, and its variants help in understanding why an analyzed clustering is of a specific quality.

7. CONCLUSION

Evaluation of clustering results on static data sets received a lot of research attention for several decades. Many recent publications deal with the important task of clustering on evolving data streams. However, despite the amount of work on evaluation measures and on stream clustering algorithms, no effort has been made to meet the requirements of both tasks. In this paper we proposed a novel and effective evaluation measure for clustering on evolving data streams called CMM. It is the first measure that takes the important properties of the streaming context into account. It is based on a novel mapping component that can handle emerging and disappearing clusters correctly. We included CMM into the open source MOA framework [29] and showed in extensive experiments on real and synthetic data that it is a robust measure precisely reflecting errors in data stream scenarios.

Acknowledgments. This work has been supported by the UMIC Research Centre, RWTH Aachen University, Germany. Special thanks go to Marc Wichterich for his valuable comments and thorough reviewing of the paper.

8. REFERENCES

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *VLDB*, pages 81–92, 2003.

[2] F. B. Baker and L. J. Hubert. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association (JASA)*, 70(349):31–38, 1975.

[3] D. Barbará and P. Chen. Using the fractal dimension to cluster datasets. In *ACM SIGKDD*, pages 260–264, 2000.

[4] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *ACM SIGKDD*, pages 139–148, 2009.

[5] A. Bifet, G. Holmes, B. Pfahringer, P. Kranen, H. Kremer, T. Jansen, and T. Seidl. MOA: Massive online analysis, a framework for stream classification and clustering. In *JMLR*, 2010.

[6] M. Bouguessa, S. Wang, and H. Sun. An objective approach to cluster validation. *Pattern Recognition Letters*, 27(13):1419–1430, 2006.

[7] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. R. Dougherty. Model-based evaluation of clustering validation measures. *Pattern Recognition*, 40(3):807–824, 2007.

[8] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *SIAM SDM*, pages 328–339, 2006.

[9] Y. Chen and L. Tu. Density-based clustering for real-time stream data. In *ACM SIGKDD*, pages 133–142, 2007.

[10] T. Cover and J. Thomas. *Elements of Information Theory (2nd Edition)*. Wiley-Interscience, 2006.

[11] J. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[12] E. Folkes and C. Mallows. A method for comparing two hierarchical clusterings. *JASA*, 78:553–569, 1983.

[13] B. Gartner. Fast and robust smallest enclosing balls. In *ESA*, pages 325–338. Springer, 1999.

[14] M. Halkidi and M. Vazirgiannis. A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, 29(6):773–786, 2008.

[15] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[16] J. A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[17] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[18] L. J. Hubert and J. R. Levin. A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 83(6):1072–1080, 1976.

[19] A. Jain, Z. Zhang, and E. Y. Chang. Adaptive non-linear clustering in data streams. In *ACM CIKM*, pages 122–131, 2006.

[20] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM CS*, 31(3):264–323, 1999.

[22] L. Kaufmann and P. Rousseeuw. *Finding Groups in Data: an Introduct. to Cluster Analysis*. Wiley, 1990.

[23] P. Kranen, I. Assent, C. Baldauf, and T. Seidl. Self-adaptive anytime stream clustering. In *IEEE ICDM*, pages 249–258, 2009.

[24] P. Kranen, H. Kremer, T. Jansen, T. Seidl, A. Bifet, G. Holmes, and B. Pfahringer. Clustering performance on evolving data streams: Assessing algorithms and evaluation measures within MOA. In *IEEE ICDMW*, pages 1400–1403, 2010.

[25] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *IEEE ICDM*, pages 911–916, 2010.

[26] M. Meila. Comparing clusterings: an axiomatic view. In *ICML*, pages 577–584, 2005.

[27] G. W. Milligan. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 45(3):325–342, 1980.

[28] G. W. Milligan. A monte carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, 46(2):187–199, 1981.

[29] MOA project. http://moa.cs.waikato.ac.nz.

[30] S. G. Mojaveri, E. Mirzaeian, Z. Bornaee, and S. Ayat. New approach in data stream association rule mining based on graph structure. In *IEEE ICDM*, pages 158–164, 2010.

[31] L. O'Callaghan, A. Meyerson, R. Motwani, N. Mishra, and S. Guha. Streaming-data algorithms for high-quality clustering. In *ICDE*, pages 685–694, 2002.

[32] W. Rand. Objective criteria for the evaluation of clustering methods. *JASA*, 66:846–850, 1971.

[33] C. Rijsbergen. *Information Retrieval (2nd Edition)*. Butterworths, London, 1979.

[34] F. J. Rohlf. Methods for comparing classifications. *Annual Review of Ecology and Sys.*, 5:101–113, 1974.

[35] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, pages 410–420, 2007.

[36] V. Roth, M. L. Braun, T. Lange, and J. M. Buhmann. Stability-based model order selection in clustering with applications to gene expression data. In *ICANN*, pages 633–640. Springer, 2002.

[37] S. Saitta, B. Raphael, and I. F. C. Smith. A comprehensive validity index for clustering. *Intell. Data Anal. (IDA)*, 12(6):529–548, 2008.

[38] M. J. Song and L. Zhang. Comparison of cluster representations from partial second- to full fourth-order cross moments for data stream clustering. In *IEEE ICDM*, pages 560–569, 2008.

[39] S. Van Dongen. Performance criteria for graph clustering and markov cluster experiments. *Report-Information systems*, (12):1–36, 2000.

[40] L. Wang, U. T. V. Nguyen, J. C. Bezdek, C. Leckie, and K. Ramamohanarao. iVAT and aVAT: Enhanced visual analysis for cluster tendency assessment. In *PAKDD (1)*, pages 16–27. Springer, 2010.

[41] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *ACM SIGKDD*, pages 877–886, 2009.

[42] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

[43] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *ML*, 55(3):311–331, 2004.