Clustering Performance on Evolving Data Streams: Assessing Algorithms and Evaluation Measures within MOA

Philipp Kranen*, Hardy Kremer*, Timm Jansen*, Thomas Seidl*, Albert Bifet†, Geoff Holmes† and Bernhard Pfahringer‡

*Data Management and Data Exploration Group, RWTH Aachen University, Germany
†University of Waikato Hamilton, New Zealand
‡University of Waikato Hamilton, New Zealand

Abstract—In today’s applications, evolving data streams are ubiquitous. Stream clustering algorithms were introduced to gain useful knowledge from these streams in real-time. The quality of the obtained clusterings, i.e., how good they reflect the data, can be assessed by evaluation measures. A multitude of stream clustering algorithms and evaluation measures for clusterings were introduced in the literature; however, until now there is no general tool for a direct comparison of the different algorithms or the evaluation measures. In our demo, we present a novel experimental framework for both tasks. It offers the means for extensive evaluation and visualization and is an extension of the Massive Online Analysis (MOA) software environment released under the GNU GPL License.

Keywords—data streams, clustering, evaluation measures

I. INTRODUCTION

Data streams are ubiquitous nowadays and a multitude of algorithms exist for stream learning scenarios, e.g., stream classification or stream clustering. In most publications, newly proposed algorithms are only compared to a small subset or even none of the competing solutions, making the assessment of their actual effectiveness tough. Moreover, the majority of experimental evaluations use only small amounts of data. In the context of data streams this is disappointing, because to be truly useful the algorithms need to be capable of handling very large (potentially infinite) streams of examples. Demonstrating systems only on small amounts of data does not build a convincing case for capacity to solve more demanding data stream applications.

In traditional batch learning scenarios, evaluation frameworks were introduced to cope with the comparison issue. One of these frameworks is the well-known WEKA Data Mining Software that supports adding new algorithms and evaluation measures in a plug-and-play fashion [1], [2]. As data stream learning is a relatively new field, the evaluation practices are not nearly as well researched and established as they are in the traditional batch setting. For this purpose, a framework for stream learning evaluation was recently introduced, called Massive Online Analysis (MOA) [3], that builds on the work in WEKA. So far, however, MOA only considers stream classification algorithms. Accordingly, no stream clustering evaluation tool exists that offers a suite of implemented stream clustering algorithms and evaluation measures, although stream clustering is an active field of research with many recent publications.

Besides comparing new algorithms to the state of the art, the choice of the evaluation measures is a second key issue for clustering performance on evolving data streams. Most often traditional measures are employed that do not reflect the errors that are specific to evolving data streams, e.g., through moving or merging clusters. Therefore, our goal is to build an experimental stream clustering system able to evaluate state-of-the-art methods both regarding clustering algorithms and evaluation measures.

II. FEATURES AND SYSTEM ARCHITECTURE

In this section we briefly describe the usage and configuration of our system as well as how to extend the framework. A detailed description will be available in the manual and is beyond the scope of this demo paper.

Our goal is to build an experimental framework for clustering data streams similar to the WEKA framework, making it easy for researchers to run experimental data stream benchmarks. The MOA framework offers such possibilities for classification algorithms on data streams. The new features of our MOA extension to stream clustering are:

• data generators for evolving streams (including events like novelty, merging clusters, etc. [4]),
• an extensible set of stream clustering algorithms,
• an extensible set of evaluation measures,
• methods to assess evaluation measures under specific error scenarios
• visualization tools for analyzing results and comparing different settings.

From the existing MOA framework we inherit data generators that are most commonly found in the literature. MOA streams can be built using synthetic generators, reading ARFF files, joining several streams, or filtering streams. They allow the simulation of a potentially infinite sequence of data and include concept drift simulation for classification tasks [5]. For stream clustering we added new data generators that support the simulation of cluster evolution events such as merging or disappearing of clusters [4]. We
provide more detail on the usage of events in Section II. The contained stream clustering algorithms and evaluation measures are treated in Sections III and IV respectively along with a description of how to assess either of these.

Both architecture and usage of our stream clustering framework follow the same straightforward workflow concept (cf. Figure 1): first a data feed is chosen and configured, then a stream clustering algorithm and its settings are fixed, and last a set of evaluation measures is selected.

Our framework can be easily extended in all the three first parts of the workflow described above, i.e. new data feeds or generators can be added as well as additional algorithms or evaluation measures. We will exemplarily sketch the integration of a new algorithm. Details on all extension points and interfaces are available on our website (cf. Section V). Classes that implement the interface Clusterer.java are added to the framework via reflections on start up. The three main methods of this interface are:

- void resetLearningImpl(): a method for initializing a clusterer learner
- void trainOnInstanceImpl(Instance): a method to train a new instance
- Clustering getClusteringResult(): a method to obtain the current clustering result for evaluation or visualization

After the evaluation process is started, several options for analyzing the outputs are given: a) the stream can be stopped and the current (micro) clustering result can be passed as a data set to the WEKA explorer for further analysis or mining; b) the evaluation measures, which are taken at configurable time intervals, can be stored as a .csv file to obtain graphs and charts offline using a program of choice; c) last but not least both the clustering results and the corresponding measures can be visualized online within our framework (cf. Figure 3). Our framework allows the simultaneous configuration and evaluation of two different setups for direct comparison, e.g. of two different algorithms on the same stream or the same algorithm on streams with different noise levels etc.

III. ASSESSING ALGORITHMS

MOA contains several stream clustering methods such as StreamKM++ [6], CluStream [7], ClusTree [8], DenStream [9], D-Stream [10], CobWeb [11] and others. MOA contains measures for analyzing the performance of the clustering models generated from both online and offline components. The available measures evaluate both the correct assignment of examples [12] and the internal structure of the resulting clustering [13]. The visualization component (cf. Figure 3) allows to visualize the stream as well as the clustering results, choose dimensions for multi dimensional settings, and compare experiments with different settings in parallel (cf. Section II).

Figure 2 shows a screenshot of the configuration dialog for our RBF data generator with events. Generally the dimensionality, number, and size of clusters can be set as well as the drift speed, decay horizon (aging), noise rate, etc. Events constitute changes in the underlying data model such as growing of clusters, merging of clusters or creation of new clusters [4]. Using the event frequency and the individual event weights, one can study the behaviour and performance of different approaches on various settings. Finally, the settings for the data generators can be stored and loaded, which offers the opportunity of sharing settings and thereby providing benchmark streaming data sets for repeatability and comparison.

Figure 3 shows a screenshot of our visualization tab. For this screenshot two different settings of the CluStream algorithm [7] were compared on the same stream setting (including merge/split events every 50000 examples) and four measures were chosen for online evaluation (F1, Precision, Recall and SSQ). The upper part of the GUI offers options to pause and resume the stream, adjust the visualization speed, choose the dimensions for x and y as well as the components to be displayed (points, micro- and macro clustering and ground truth). The lower part of the GUI displays the measured values for both settings as numbers (left side, including mean values) and the currently selected measure as a plot over the arrived examples (right, F1 measure in this example). For the given setting one can see a clear drop in the performance after the split event at roughly 160000 examples (event details are shown when choosing the corresponding vertical line in the plot). While this holds for both settings, the left configuration (red, CluStream with 100 micro clusters) is constantly outperformed by the right configuration (blue, CluStream with 20 micro clusters). A video containing an online demo of our system can be found at our website along with more screenshot and explanations (cf. Section V).

IV. ASSESSING EVALUATION MEASURES

The results of the evaluation of stream clustering algorithms highly depend on the employed evaluation measures. These measures can be categorized into structural measures,
called internal, and ground truth based measures, called external. A list of evaluation measures from the literature is shown in Table I. An extensive study of thirty internal measures can be found in [13] and recent studies for external measures in [14] or [12].

<table>
<thead>
<tr>
<th>Internal measures</th>
<th>External measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>Rand statistic</td>
</tr>
<tr>
<td>C Index</td>
<td>Jaccard coefficient</td>
</tr>
<tr>
<td>Point-Biserial</td>
<td>Folkes and Mallow Index</td>
</tr>
<tr>
<td>Log Likelihood</td>
<td>Hubert Γ statistics</td>
</tr>
<tr>
<td>Dunn’s Index</td>
<td>Minkowski score</td>
</tr>
<tr>
<td>Tau</td>
<td>Purity</td>
</tr>
<tr>
<td>Tau A</td>
<td>van Dongen criterion</td>
</tr>
<tr>
<td>Tau C</td>
<td>V-measure</td>
</tr>
<tr>
<td>Somer’s Gamma</td>
<td>Completeness</td>
</tr>
<tr>
<td>Ratio of Repetition</td>
<td>Homogeneity</td>
</tr>
<tr>
<td>Modified Ratio of Repetition</td>
<td>Variation of information</td>
</tr>
<tr>
<td>Adjusted Ratio of Clustering</td>
<td>Mutual information</td>
</tr>
<tr>
<td>Fagan’s Index</td>
<td>Class-based entropy</td>
</tr>
<tr>
<td>Deviation Index</td>
<td>Cluster-based entropy</td>
</tr>
<tr>
<td>Z-Score Index</td>
<td>Precision</td>
</tr>
<tr>
<td>D Index</td>
<td>Recall</td>
</tr>
<tr>
<td>Silhouette coefficient</td>
<td>F-measure</td>
</tr>
</tbody>
</table>

Table I
INTERNAL AND EXTERNAL CLUSTERING EVALUATION MEASURES.

Our demo provides the means to assess the performance of such evaluation measures by testing them in specific clustering error scenarios. We obtain these scenarios by generating clusterings out of our synthetic stream (cf. Section III) that reflect a desired error level. Then, we assess the performance by testing whether the obtained qualities of the evaluation measures reflect the error in these generated clusterings. We can create cluster center position errors, called position offset errors, and radius errors. For position offset errors, the cluster centers are shifted away from their ground truth position. The maximal error level of 1 indicates that the ground truth cluster and the error cluster are positioned next to each other. There are two types of radius errors: the radius decrease error indicates that the generated error clusters have a radius that is smaller than the radius of the corresponding ground truth cluster. The maximal 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 radius increase error is realized analogously: an error level of 1 indicates that the radius of the error cluster has doubled. Moreover, clustering evaluation measures are very sensitive to the overlap of clusters in the analyzed clusterings. The overlap is highly dependent on the used aging of the clustered points; looking at a larger history of data points results in more tail-like clusters, which in turn yields a higher overlap. In our demo, we can analyze the effects of different aging scenarios and measure the occurring overlap for detailed analysis.

Examples of the application of our demo can be seen in Fig. 4, where we evaluated some of the measures from Table 4 for the cluster radius decrease error. We can conclude some interesting facts from these plots: Fig. 4(a) indicates that the general performance of internal measure is poor for this error scenario, i.e. the measures do not reflect the increasing error in the clusterings. For the external measures we distinguish between two overlap scenarios, i.e. an overlap of 0% in Fig. 4(b) and an overlap of 40% in Fig. 4(c). Generally speaking, the external measures do reflect the errors, i.e. with increasing errors the values of the measures drop; however, with a higher overlap (Fig. 4(c)), the measures start with lower qualities for the error-free clusterings.

V. DEMONSTRATION PLAN

We will showcase our prototype system and demonstrate its usage, user interaction, fast methods, and scalable system design through scenarios on several data streams.

We would like to discuss with experts and practitioners to get new input and ideas for further features and extensions of our framework. Particularly, we would also like to focus on giving insight into workings and drawbacks of different approaches, for example in which situation does an algorithm or an evaluation measure fail etc.
We think the audience will benefit from knowing that a clustering system for massive data streams will be available with the following characteristics:

- Benchmark settings for streaming data through stored, shared, and repeatable settings for the various data feeds and noise options, both synthetic and real
- Set of implemented algorithms for comparison to approaches from the literature
- Set of evaluation measures and methods to assess their performance on evolving data streams
- Open source tool and framework for research and teaching similar to WEKA

MOA sources can be found at [3] along with a tutorial, an API reference, a user manual, and a manual about mining data streams. Several examples of how the software can be used are available. Additional material regarding the extension of MOA to stream clustering can be found at [http://dme.rwth-aachen.de/moa-datastream/](http://dme.rwth-aachen.de/moa-datastream/)

The material includes a live video of the software as well as screenshots and explanations for the most important interfaces that are needed for extending our framework through novel data feeds, algorithms or measures.

VI. Conclusions

Our goal is to build an experimental framework for clustering data streams similar to the WEKA framework, so that it will be easy for researchers to run experimental data stream clustering benchmarks. Our stream clustering framework provides a set of data generators, algorithms and evaluation measures. Besides insights into workings and drawbacks of different algorithms and measures our framework allows the creation of benchmark streaming data sets through stored, shared and repeatable settings for the data feeds. The proposed demonstration builds upon the MOA framework, the sources are publicly available and are released under the GNU GPL license. Videos, screenshots and the most important interfaces for extending the framework can be found on our website along with a short explanation.

### REFERENCES


