Benchmarking Stream Clustering Algorithms within the MOA Framework

Philipp Kranen, Hardy Kremer, Timm Jansen, Thomas Seidl
Data Management and Data Exploration Group, RWTH Aachen University, Germany
{kranen, kremer, jansen, seidl}@cs.rwth-aachen.de

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

ABSTRACT
In today’s applications, massive, evolving data streams are ubiquitous. To gain useful information from this data, real time clustering analysis for streams is needed. A multitude of stream clustering algorithms were introduced. However, assessing the effectiveness of such an algorithm is challenging, because up to now there is no tool that allows a direct comparison of these algorithms. We present a novel clustering evaluation framework for data streams. It is an extension of Massive Online Analysis (MOA), a software environment for implementation and evaluation of algorithms for online learning from evolving data streams. Our stream clustering algorithm evaluation framework includes a collection of online clustering methods and offers tools for extensive evaluation and visualization. Moreover, it allows for bidirectional interaction with WEKA, since it uses the same internal data structures. Our framework is designed for extensibility, allowing straightforward adding of more algorithms, evaluation measures, and data feeds. It is released under the GNU GPL license.

1. INTRODUCTION
In data stream scenarios data arrives at high speed strictly constraining processing algorithms in space and time. To adhere to these constraints, specific requirements have to be fulfilled by the stream processing algorithms that are different from traditional batch processing settings. The most significant requirements are the following:

1. Process an example at a time, and inspect it only once (at most).
2. Work in a limited amount of time.
3. Use a limited amount of memory.
4. Be ready to output a model at any time.

Stream learning algorithms are an important type of stream processing algorithms: In a repeated cycle, the learned model is constantly updated to reflect the incoming examples from the stream. They do so without exceeding their memory and time bounds. After processing an incoming example, the algorithms are always able to output a model. Typical learning tasks in stream scenarios are classification, outlier analysis, and clustering.

Since a multitude of algorithms exist for stream learning scenarios, a thorough comparison by experimental evaluation is crucial. 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 [9].

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 [8, 11, 12]. 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.

Our goal is to build an experimental clustering system able to evaluate several state-of-the-art methods with large data streams; therefore, in this demonstration we present a novel framework for stream clustering, extending MOA. Our framework permits evaluation of data stream clustering algorithms on large streams, in the order of tens of millions of examples where possible, and under explicit memory limits. Any less than this does not actually test algorithms in a realistically challenging setting.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXX-XX-X/XX/XX ...$10.00.
In the next sections we present the features of our new framework for stream clustering evaluation, the architecture, and the advantages of our system.

2. FEATURES AND OBJECTIVES

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 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, merge, etc. [13]),
- an extensible set of stream clustering algorithms,
- evaluation measures for stream clustering,
- 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 [4]. The following generators are currently available under [3]:

- real datasets, datasets with restricted dimensions, dataset merge, leave out classes to simulate novelty,
- synthetic data (RBF, grid based, spline based),
- different types and settings for noise to be added.

For stream clustering we added new data generators that support the simulation of cluster evolution events such as merging or disappearing of clusters [13]. We provide more detail on the usage of events in Section 3.

MOA contains several stream clustering methods such as:

- StreamKM++ [1]: It computes a small weighted sample of the data stream and it uses the k-means++ algorithm as a randomized seeding technique to choose the first values for the clusters. To compute the small sample, it employs coreset constructions using a core-set tree for speed up.
- CluStream [2]: It maintains statistical information about the data using micro-clusters. These micro-clusters are temporal extensions of cluster feature vectors. The micro-clusters are stored at snapshots in time following a pyramidal pattern. This pattern allows to recall summary statistics from different time horizons.
- ClusTree [10]: It is a parameter free algorithm automatically adapting to the speed of the stream and it is capable of detecting concept drift, novelty, and outliers in the stream. It uses a compact and self-adaptive index structure for maintaining stream summaries.
- Den-Stream [5]: It uses dense micro-clusters (named core-micro-cluster) to summarize clusters. To maintain and distinguish the potential clusters and outliers, this method presents core-micro-cluster and outlier micro-cluster structures.

MOA contains measures for analyzing the performance of the clustering models generated. It contains measures commonly used in the literature as well as novel evaluation measures to compare and evaluate both online and offline components. The available measures evaluate both the correct assignment of examples [6] and the compactness of the resulting clustering. 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 3).

Beside providing an evaluation framework, the second key objective is the extensibility of the benchmark suite regarding the set of implemented algorithms as well as the available data feeds and evaluation measures. Figure 1 illustrates the available extension points, technical details are given in the next section.

3. SYSTEM ARCHITECTURE

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

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 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.

Figure 2 shows a screenshot of the configuration dialog for our RBF data generator with events. Generally the di-
Classes that implement the interface Clusterer.java and interfaces are available on our website (cf. Section 4). 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. Additional material regarding the extension of MOA to stream clustering can be found at http://dme.rwth-aachen.de/moa-datastream/

We think the audience will benefit from knowing that a clustering system for massive data streams will be available with the following characteristics:

- benchmark streaming data sets 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
- 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. 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.
5. 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 approaches 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.

6. ACKNOWLEDGMENTS

This work has been supported by the UMIC Research Centre, RWTH Aachen University, Germany.

7. REFERENCES


Figure 3: Visualization tab of the clustering MOA graphical user interface.