Stream Data Mining Using the MOA Framework

Philipp Kranen[1], Hardy Kremer[1], Timm Jansen[1], Thomas Seidl[1],
Albert Bifet[2], Geoff Holmes[2], Bernhard Pfahringer[2], and Jesse Read[2]

[1] Data Management and Exploration Group, RWTH Aachen University, Germany
{kranen,kremer,jansen,seidl}@cs.rwth-aachen.de
[2] Department of Computer Science, University of Waikato, Hamilton, New Zealand
{abifet,geoff,bernhard,jmr30}@cs.waikato.ac.nz

Abstract. **M**assive **O**nline **A**nalysis (MOA) is a software framework that provides algorithms and evaluation methods for mining tasks on evolving data streams. In addition to supervised and unsupervised learning, MOA has recently been extended to support multi-label classification and graph mining. In this demonstrator we describe the main features of MOA and present the newly added methods for outlier detection on streaming data. Algorithms can be compared to established baseline methods such as LOF and ABOD using standard ranking measures including Spearman rank coefficient and the AUC measure. MOA is an open source project and videos as well as tutorials are publicly available on the MOA homepage.

1 Introduction

Data streams are ubiquitous, ranging from sensor data to web content and click stream data. Consequently there is a rich and growing body of literature on stream data mining. For traditional mining tasks on static data, several established software frameworks such as WEKA (Waikato Environment for Knowledge Analysis) provide mining algorithms and evaluation methods from the research literature. Such environments allow for both choosing an algorithm for a given application as well as comparing new approaches against the state of the art.

MOA [4] is a software environment for implementing algorithms and running experiments for online learning from evolving data streams. MOA is designed to deal with the challenging problems of scaling up the implementation of state of the art algorithms to real world dataset sizes and of making algorithms comparable in benchmark streaming settings. MOA initially contained algorithms for stream classification and was extended over the last years to support clustering, multi-label classification and graph mining on evolving data streams. In this paper we describe the newly added methods for outlier detection on data streams in comparison to established baseline methods.

Only two other open-source data streaming packages exist: VFML and a RapidMiner plugin. The VFML (Very Fast Machine Learning) [5] toolkit was the first open-source framework for mining high-speed data streams and very large data sets. It was developed until 2003. VFML is written mainly in standard C, and contains tools for learning decision trees (VFDT and CVFDT),

S.-g. Lee et al. (Eds.): DASFAA 2012, Part II, LNCS 7239, pp. 309–313, 2012.

for learning Bayesian networks, and for clustering. The data stream plugin (formerly: concept drift plugin) [6] for RapidMiner (formerly: YALE (Yet Another Learning Environment)), is an extension to RapidMiner implementing operators for handling real and simulated concept drift in evolving streams.

MOA is built on experience with both WEKA and VFML. The main advantage of MOA is that it provides many of the recently developed data stream algorithms, including learners for multi-label classification, graph mining and outlier detection. Generally, it is straightforward to use or to extend MOA.

2 The MOA Framework

In the following we first describe the general architecture, usage and features of MOA. In Section 2.1 we discuss the new components for outlier detection and Section 2.2 provides information on documentation and tutorials.

Architecture, extension points and workflow follow the design depicted in Figure 1. The three components *data feed*, *algorithm* and *evaluation method* have to be chosen and parameterized in order to run an experiment. For each component a simple interface is available in MOA which can be used to include new components. MOA is written in Java, allowing portability to many platforms and usage of the well-developed support libraries. MOA can be used from the command line, as a library, or via the graphical user interface (cf. Figure 2).

Data Streams. MOA streams can be build using generators, reading ARFF files, joining several streams, or filtering streams. Most of the data generators commonly found in the literature are provided: Random Tree Generator, SEA Concepts Generator, STAGGER Concepts Generator, Rotating Hyperplane, Random RBF Generator, LED Generator, Waveform Generator, and Function Generator. Settings can be stored to generate benchmark data sets.

Classification. MOA contains a range of classification methods such as: Naive Bayes, Stochastic Gradient Descent, Perceptron, Hoeffding Tree, Adaptive Hoeffding Tree, Boosting, Bagging, and Leveraging Bagging.

Clustering. For clustering MOA contains several stream clustering methods, such as StreamKM++, CluStream, ClusTree, Den-Stream, CobWeb, as well as a large set of evaluation methods including the recent Cluster Mapping Measure (CMM) [7]. Dynamic visualization of cluster evolution is available, as depicted in Figure 2.

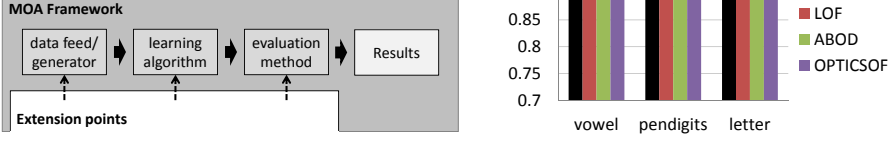

Fig. 1. Left: Architecture and workflow of MOA. Right: outlier detection results.

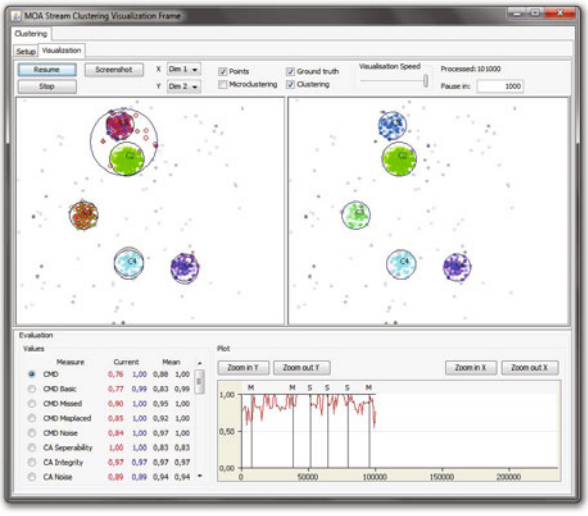

Fig. 2. Visualization tab of the clustering MOA graphical user interface

Multi-label Classification. Multi-label classification has seen considerable development in recent years, but so far most of the work has been carried out in the context of batch learning. MOA implements multi-label stream generators and several state of the art methods: ECC Ensembles of classifier-chains, EPS Ensembles of Pruning Sets, Multi-label Hoeffding Trees, and multi-label adaptive bagging methods.

Graph Mining. MOA also contains a framework for studying graph pattern mining on time-varying streams [3]. All methods work on coresets of closed subgraphs, compressed representations of graph sets. The methods maintain these sets in a batch-incremental manner, but use different approaches to address potential concept drift. MOA implements IncGraphMiner, WinGraphMiner and AdaGraphMiner.

2.1 Outlier Detection on Data Streams Using MOA

Outlier detection is an important task in stream data mining. Applications range from fault detection in network or transaction data to event or error detection in sensor networks and remote monitoring. Several approaches have been proposed in the literature, which make use of different paradigms. In [2] a solution using a hierarchy of clusterings is proposed, other solutions follow density based or distance based approaches. We added an `OutlierDetector` interface to MOA, which allows for easy inclusion of new or additional methods.

To evaluate the performance of these approaches it is essential to have a comparison to established methods such as LOF or ABOD. We use the outlier algorithms implemented in the ELKI open source framework[1] [1] and make them available in the MOA GUI. These methods can be seen as a baseline since they do not impose any time restrictions on themselves and assume random access to the data. The ELKI algorithms are run on a user defined history of point, where the granularity of the evaluation and the length of the history can be parameterized. As evaluation measures MOA currently provides three standard ranking measures, namely Spearman rank coefficient (cf. Fig. 1), Kendall's Tau and AUC (Area Under the ROC Curve) for outlier detection on data streams.

2.2 Website, Tutorials, and Documentation

MOA can be found at `http://moa.cs.waikato.ac.nz/`. The website includes a video and tutorials, an API reference, a user manual, and a general manual about mining data streams. Several examples of how the software can be used are available. We are currently working on extending the framework to include data stream regression, and frequent pattern learning.

3 Demo Plan and Conclusions

In this demonstrator we focus on presenting the newly added algorithms for outlier detection and the corresponding evaluation on data streams. For researchers MOA yields insights into advantages and disadvantages of different approaches and allows for the creation of benchmark streaming data sets through stored, shared and repeatable settings for the data feeds. Practitioners can use the framework to easily compare algorithms and apply them to real world data sets and settings. Besides providing algorithms and measures for evaluation and comparison, MOA is easily extensible with new contributions and allows for the creation of benchmark scenarios.

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

References

1. Achtert, E., Kriegel, H.-P., Reichert, L., Schubert, E., Wojdanowski, R., Zimek, A.: Visual Evaluation of Outlier Detection Models. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5982, pp. 396–399. Springer, Heidelberg (2010)
2. Assent, I., Kranen, P., Baldauf, C., Seidl, T.: Anyout: Anytime Outlier Detection on Streaming Data. In: Lee, S.-G., et al. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 228–242. Springer, Heidelberg (2012)

[1] ELKI: `http://elki.dbs.ifi.lmu.de/` - MOA: `http://moa.cs.waikato.ac.nz/`

3. Bifet, A., Holmes, G., Pfahringer, B., Gavaldà, R.: Mining frequent closed graphs on evolving data streams. In: 17th ACM SIGKDD, pp. 591–599 (2011)
4. Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: Moa: Massive online analysis, a framework for stream classification and clustering. Journal of Machine Learning Research - Proceedings Track 11, 44–50 (2010)
5. Hulten, G., Domingos, P.: VFML – a toolkit for mining high-speed time-changing data streams (2003)
6. Klinkenberg, R.: Rapidminer data stream plugin. RapidMiner (2010), `http://www-ai.cs.uni-dortmund.de/auto?self=eit184kc`
7. Kremer, H., Kranen, P., Jansen, T., Seidl, T., Bifet, A., Holmes, G., Pfahringer, B.: An effective evaluation measure for clustering on evolving data stream. In: 17th ACM SIGKDD, pp. 868–876 (2011)