International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967
Volume 2 , Issue 1(Jan-Feb 2015), PP47-50

Clustering Performance on Evolving Data Streams

Shruti Bawage[1]
[1](M.Tech (CSE) MIT JNTU Hyderabad , bawageshruti17@gmail.com, ,*India)*

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 clustering's, i.e. how good they reflect the data, can be assessed by evaluation measures. A multitude of stream clustering algorithms and evaluation measures for clustering's 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.

Keywords :- RBM, DBN, DBM, SVM, k-NN

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. 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. Our extension of MOA to stream clustering offers the following new features:

1.data generators for evolving streams

2.an extensible set of stream clustering algorithms,

3. an extensible set of evaluation measures,

4. methods to assess evaluation measures under specific error scenarios

5.visualization tools for analyzing results and comparing different settings.

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.

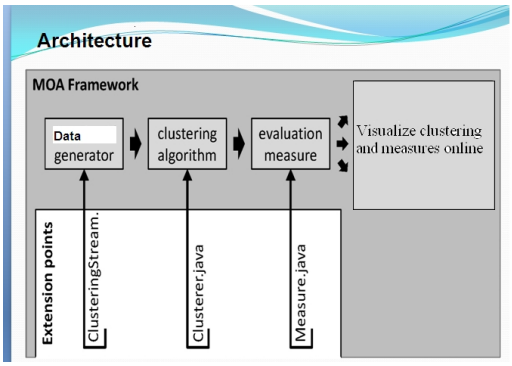

Fig 1: Extension points of the MOA stream clustering framework.

After the evaluation process is started, several options for analyzing the outputs are given 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.

III .ASSESSING ALGORITHMS

MOA contains several stream clustering methods such as StreamKM++ [6], CluStream [7], ClusTree [8], Den- Stream [7], D-Stream [6], CobWeb [3] 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 [2] and the internal structure of the resulting clustering [6]. The visualization component (cf. Figure 3) allows to visualize the stream as well as the clustering results, choose dimensions for multidimensional settings, and compare experiments with different settings in parallel. 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. 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 50,000 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, 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 160,000 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

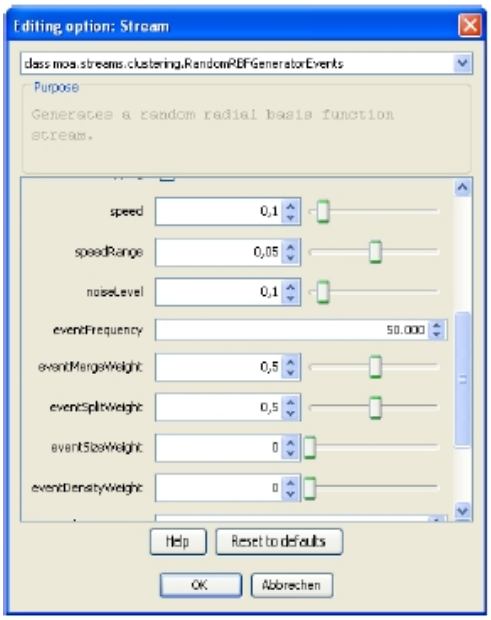

Fig 2 :Option dialog for the RBF data generator.

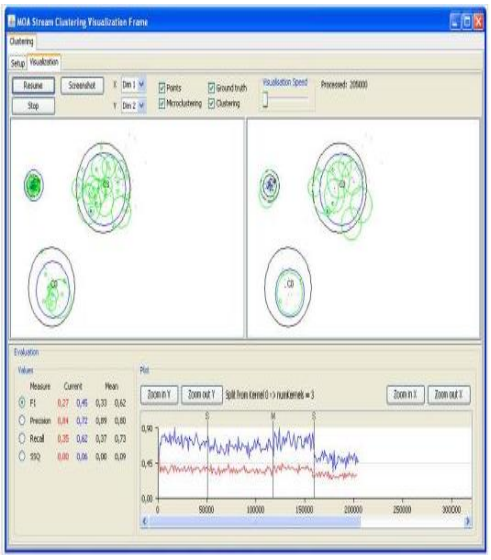

Fig 3. Visualization tab of the clustering MOA graphical user interface.

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 clustering's 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 clustering's. 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.

IV. CONCLUSION

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 the creation of benchmark streaming data sets through stored, shared and repeatable settings for the data feeds.

REFERENCES

[1] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," SIGKDD Explorations 11(1):10–18, 2009.
[2] E. M¨uller, I. Assent, S. G¨unnemann, T. Jansen, and T. Seidl, "OpenSubspace: An open source framework for evaluation and exploration of subspace clustering algorithms in weka," in OSDM in conjunction with PAKDD, 2009, pp. 2–13.
[3] 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 Journal of Machine Learning Research (JMLR), 2010.
[4] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult, "MONIC: modeling and monitoring cluster transitions," in ACM KDD, 2006, pp. 706–711.
[5] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavald`a, "New ensemble methods for evolving data streams." in ACM KDD, 2009, pp. 139–148.
[6] M. R. Ackermann, C. Lammersen, M. M¨artens, C. Raupach, C. Sohler, and K. Swierkot, "StreamKM++: A clustering algorithm for data streams," in SIAM ALENEX, 2010.
[7] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in VLDB, 2003.
[8] P. Kranen, I. Assent, C. Baldauf, and T. Seidl, "Self-adaptive anytime stream clustering," in ICDM, 2009, pp. 249–258.
[9] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in SDM, 2006.