Algorithm Recommendation for Data Streams

Jáder M. C. de Sá[1] André L. D. Rossi[2] Gustavo E. A. P. A. Batista[3] Luís P. F. Garcia[1]

[1]University of Brasilia (UnB) [2]São Paulo State University (UNESP)

[3]University of New South Wales (UNSW)

Introduction

In the last decades, many companies have taken advantage of knowledge discovery to identify valuable information in massive volumes of data generated at high frequency. Machine learning techniques can be employed for knowledge discovery since they can extract patterns from data and induce models to predict future events. However, dynamic and evolving environments usually generate non-stationary data streams. Hence, models trained in these scenarios may perish over time due to seasonality or concept drift. Periodic retraining can help, but a fixed hypothesis space may no longer be appropriate. An alternative solution is to use meta-learning for regular algorithm selection in time-changing environments, choosing the bias that best suits the current data. In this work, we present an enhanced framework for data stream algorithm selection based on MetaStream.

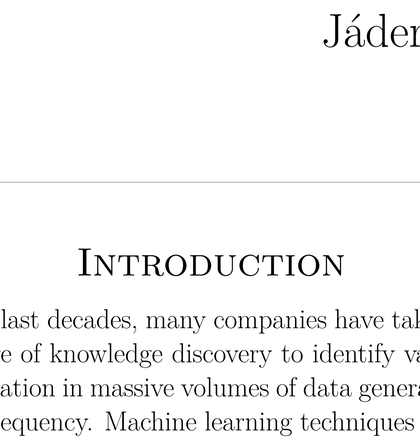

Figure 1:Online phase of MetaStream.

Our approach uses meta-learning and incremental learning to actively select the best algorithm for the current concept in a time-changing environment. Different from previous work, we use a rich set of state-of-the-art meta-features, and an incremental learning approach in the meta-level based on LightGBM. The results show that this new strategy can improve the recommendation accuracy of the best algorithm in time-changing data.

Methods

The MetaStream framework based on Rossi et al. (2014) [1], performes a continuous selection of algorithms for the current stream of data. Initially, in the offline stage, it performs hyperparameter tuning, validation and training data generation with a small initial sample of data. Then, the online phase acts in the dynamic environment recommending an algorithm for a given window of the data.

The offline phase starts after a given initial amount of training data has arrived. With this batch of data, Metastream induces the base-algorithms through k-fold cross-validation for hyperparameter tunning. With the same initial data, the algorithm continues by swiping a sliding window through the data, as shown in Figure 2. In this setting, Metastream induces base models using the base-level algorithms and extracts meta-features x^m for each window ω_{bi}. The best performing model becomes the label of meta-example (y^m).

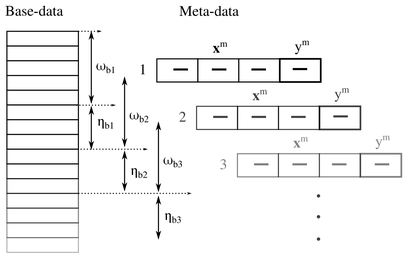

Figure 2:Meta-feature extraction from ω_b and label obtaining from η_b windows.

In the online phase, the algorithm receives a continuous stream of data. At first, it gets a feature vector $\boldsymbol{x}_b = (x_1, ..., x_p)$, and with some delay, the target attribute $y_b \in \{1, 2, .., k\}$ for classification, where k is the number of classes.

It has a window of fixed size ω_b that is used to induce the model and a window of fixed size η_b where the model induced on ω_b is evaluated. When Metastream processes all examples in η_b, it shifts the ω_b and η_b windows η_b instances to the right. Afterwards, Metastream induces a new model for this window.

Results

Figures 3 and 4 show the cumulative gain score for MetaStream, which is the difference between the recommended algorithm accuracy and the algorithm that performed better in the offline dataset (Default method) accuracy. The filled area is the cumulative sum of those score differences over time while the colours orange and blue represent incremental and non-incremental algorithms, respectively. Each black dot represents the score gain for time t.

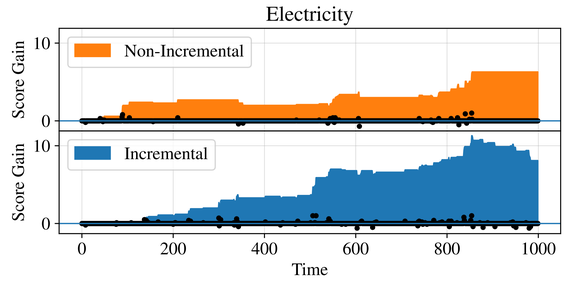

Figure 3:Cumulative score gain over time for Electricity.

In these figures, we observe a positive gain for the Electricity and PowerSupply datasets considering the recommendations of both strategies. However, the incremental strategy presents better performance for Electricity as opposed to PowerSupply, where the non-incremental is slightly better.

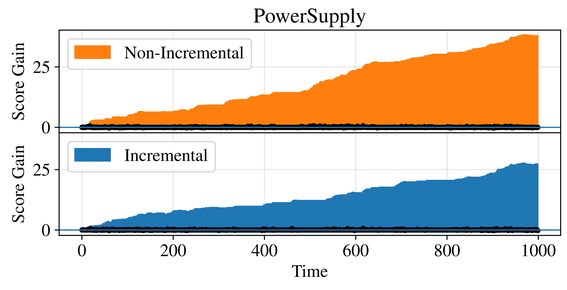

Figure 4:Cumulative score gain over time for PowerSupply.

Conclusion

We enhance MetaStream by extending the meta-features to modern and more informative ones, by including the incremental learning in the MtL level and by proposing LightGBM as meta-classifier. Although both strategies performed similarly, the incremental one had a significant lesser consumption of memory and processing time. The experimental results showed that the meta-classifier can consistently recommend the best algorithm for a given window in the data stream, leading to an increased gain of performance over time.

References

[1] André L. D. Rossi, André C. P. L. F. de Carvalho, Carlos Soares, and Bruno Feres de Souza. MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data. *Neurocomputing*, 127:52–64, 2014.

Acknowledgements

The fourth author would like to thank FAPDF for the financial support (grant 40/2020). We also would like to thank E. Alcobaça and F. Siqueira, developers of the pymfe package, for implementing additional features in the package which help us in this study.