Towards Automated Configuration of Stream Clustering Algorithms

Matthias Carnein[1], Heike Trautmann[1], Albert Bifet[2], Bernhard Pfahringer[2]

[1]Information Systems and Statistics Group,
University of Münster, Münster, Germany
{carnein,trautmann}@wi.uni-muenster.de

[2]Department of Computer Science,
University of Waikato, Hamilton, New Zealand
{abifet,bernhard}@waikato.ac.nz

Introduction

Automated algorithm configuration tries to automatically find the **best parameter settings**. Unfortunately, none of the existing approaches are directly applicable to streaming applications.
We present the first approach for automated configuration of stream clustering algorithms using an **ensemble of configurations**.

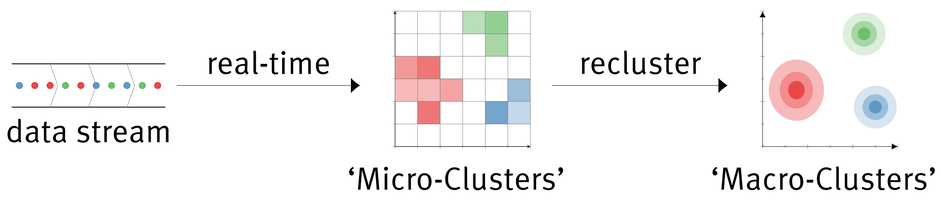

Fig. 1: Stream clustering finds clusters over time [3, 4].

Automated Algorithm Configuration

Popular approaches for algorithm configuration are `SMAC` [6] or `irace` [7]. However, both require multiple evaluations and stationary data which is **infeasible for data streams**.

First ideas for streaming data can be found in the algorithm *selection* and stream *classification* literature: `BLAST` [8], trains an ensemble of algorithms in parallel. Periodically, it selects the **best performing algorithm** of the last window as the *active* classifier.

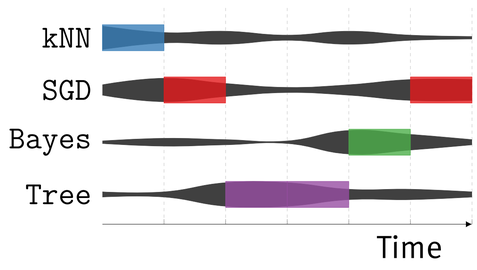

Fig. 2: `BLAST` [8] is an ensemble approach which uses the **B**est algorithm of the **LAST** window.

Automated Configuration for Stream Clustering

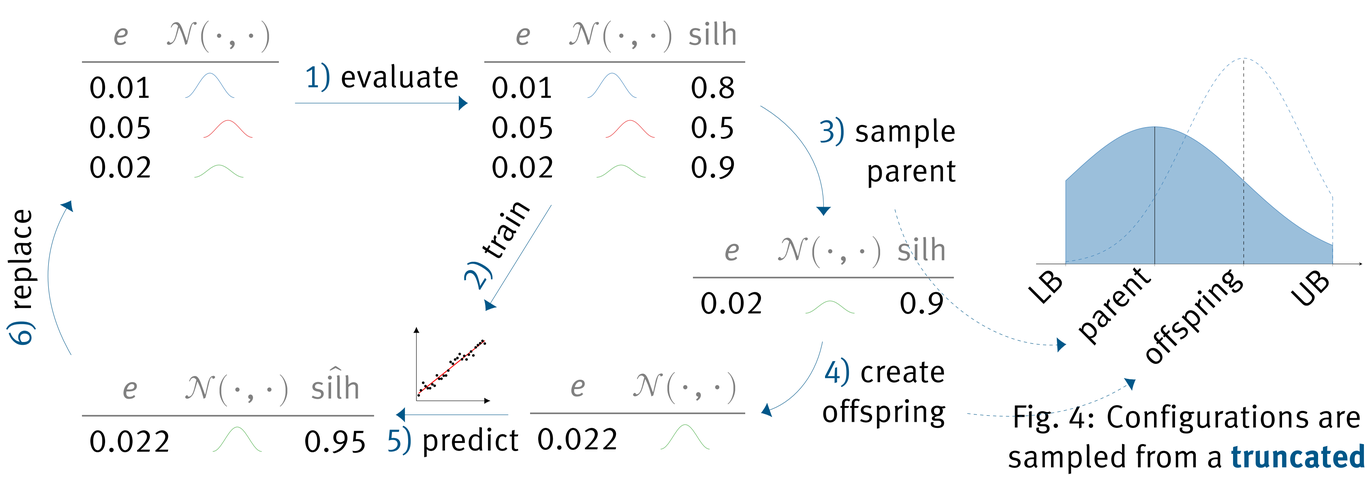

Fig. 3: `confStream` maintains, adapts and improves an ensemble of different configurations over time.

Fig. 4: Configurations are sampled from a **truncated normal distribution** that is biased towards more promising solutions [7].

We propose `confStream`, an ensemble-based approach for algorithm configuration of stream clustering algorithms. `confStream` uses a given **starting configuration** and processes the stream in windows:

1. After every window, the **clustering quality** of configurations in the ensemble is evaluated using a quality metric such as the Silhouette Width.
2. A **regression model** is trained based on the parameter value and its performance to learn how well certain configurations perform [5].
3. To **generate new configurations**, one configuration is sampled from the ensemble, proportionally to its performance.
4. New parameter values are drawn from a **truncated normal distribution** which is biased towards better solutions by reducing the standard deviation.
5. If its predicted performance is better, a random configuration in the ensemble is **replaced**, proportionally to its performance.

Experiments

We implemented `confStream` as a clustering algorithm for the **MOA framework** [1]. We evaluate cluster quality over time using the Silhouette Width every 1000 data points. We use an ensemble size of 25, generate 10 new configurations per iteration and evaluate the micro-clusters.

`confStream`	vs.	`DenStream` [2]
optimised ε		default configuration
distance threshold		$\varepsilon = 0.02, \beta = 0.2, \mu = 1$

Results:
`confStream` **considerably improves the clustering performance**. It adapts to changes better and finds solutions where the default configuration does not. While training an ensemble is slower, learning can be parallelised for real-time performance.

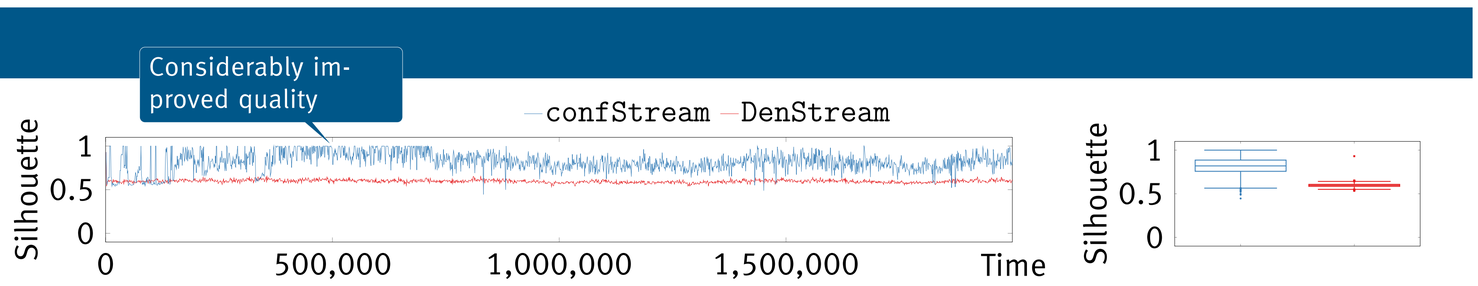

Fig. 5: `Random RBF` stream, $d = 2$, $n = 2,219,803$, Artificially generated

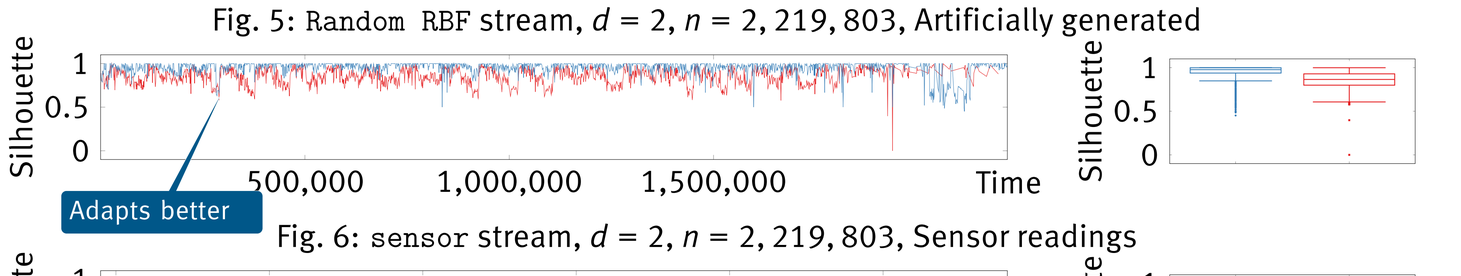

Fig. 6: `sensor` stream, $d = 2$, $n = 2,219,803$, Sensor readings

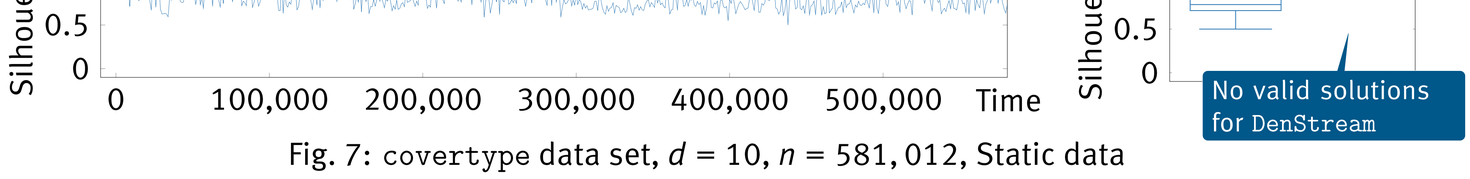

Fig. 7: `covertype` data set, $d = 10$, $n = 581,012$, Static data

Conclusion & Outlook

We explored the possibility of **automated algorithm configuration for stream clustering**. By training an ensemble of algorithms and deriving new configurations from promising solutions, we are able to find much better configurations over time.
In future work, we will extend our approach to **multiple parameters**, which can be of different types, such as **categorical or integer**. We also aim to include different algorithms into the ensemble resulting in **per-instance algorithm selection and configuration**.

Poster, Paper and Implementation available at:
https://www.carnein.com/confStream

References

[1] A. Bifet, G. Holmes, R. Kirkby and B. Pfahringer. 'MOA: Massive Online Analysis'. In: *Journal of Machine Learning Research* 11 (2010).

[2] F. Cao, M. Ester, W. Qian and A. Zhou. 'Density-based clustering over an evolving data stream with noise'. In: *Conference on Data Mining (SIAM '06)*. 2006.

[3] M. Carnein, D. Assenmacher and H. Trautmann. 'An Empirical Comparison of Stream Clustering Algorithms'. In: *Proceedings of the ACM International Conference on Computing Frontiers (CF '17)*. 2017.

[4] M. Carnein and H. Trautmann. 'Optimizing Data Stream Representation: An Extensive Survey on Stream Clustering Algorithms'. In: *Business & Information Systems Engineering (BISE)* 61 (3 2019).

[5] H. M. Gomes, J. P. Barddal, L. E. B. Ferreira and A. Bifet. 'Adaptive random forests for data stream regression'. In: *26th European Symposium on Artificial Neural Networks (ESANN '18)*. 2018.

[6] F. Hutter, H. H. Hoos and K. Leyton-Brown. 'Sequential Model-Based Optimization for General Algorithm Configuration'. In: *Proceedings of LION-5*. 2011.

[7] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle and M. Birattari. 'The irace package: Iterated Racing for Automatic Algorithm Configuration'. In: *Operations Research Perspectives* 3 (2016).

[8] J. N. van Rijn, G. Holmes, B. Pfahringer and J. Vanschoren. 'Having a Blast: Meta-Learning and Heterogeneous Ensembles for Data Streams'. In: *2015 IEEE International Conference on Data Mining*. 2015.