Towards Automated Configuration of Stream Clustering Algorithms

Matthias Carnein¹, Heike Trautmann¹, Albert Bifet², Bernhard Pfahringer²

¹Information Systems and Statistics Group, University of Münster, Münster, Germany (carnein,trautmann)@wi.uni-muenster.de
²Department 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.

**Automated Configuration for Stream Clustering**

1) train
2) sample
3) predict
4) create offspring

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

We propose confStream, an ensemble-based approach for algorithm configuration of stream clustering algorithms. confStream uses a 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 clustering 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]

<table>
<thead>
<tr>
<th>Optimised $\epsilon$</th>
<th>Default configuration</th>
<th>$\epsilon = 0.02$, $\beta = 0.2$, $\mu = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>$\epsilon$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$\epsilon$ (silhouette)</td>
<td>$\epsilon$ (silhouette)</td>
<td>$\epsilon$ (silhouette)</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>

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.

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.

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]

<table>
<thead>
<tr>
<th>Optimised $\epsilon$</th>
<th>Default configuration</th>
<th>$\epsilon = 0.02$, $\beta = 0.2$, $\mu = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>$\epsilon$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$\epsilon$ (silhouette)</td>
<td>$\epsilon$ (silhouette)</td>
<td>$\epsilon$ (silhouette)</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>

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.

References