A Java Toolbox for Analysis of Massive Data Streams using Probabilistic Graphical Models

Masegosa, Andres; Martinez, Ana M.; Ramos-López, Darío; Langseth, Helge; Nielsen, Thomas Dyhre; Salmerón, Antonio; Cabanas de Paz, Rafael; Madsen, Anders Læsø

Publication date:
2016

Link to publication from Aalborg University

Citation for published version (APA):
A Java Toolbox for Analysis of Massive Data Streams using Probabilistic Graphical Models

Andrés R. Masegosa\textsuperscript{1}, Ana M. Martínez\textsuperscript{2}, Darío Ramos-Lopez\textsuperscript{3}, Helge Langseth\textsuperscript{1}, Thomas D. Nielsen\textsuperscript{2}, Antonio Salmerón\textsuperscript{3}, Rafael Cabañas\textsuperscript{2} & Anders L. Madsen\textsuperscript{2,4}

\textsuperscript{1}Department of Computer and Information Science, NTNU, Norway \textsuperscript{2}Department of Computer Science, Aalborg University, Denmark \textsuperscript{3}Department of Mathematics, University of Almería, Spain \textsuperscript{4}Hugin Expert A/S, Aalborg, Denmark

Presentation

Data mining frameworks

- PGMs
- AMIDST
- Sta4onary data sets
- Data streams
- MLlib|Apache Spark/Flink
- MOA
- Elvira
- Infer.net
- Hugin
- Weka
- R
- Libs
- Matlab
- Apache SAMOA
- Vowpal Wabbit

Academic and Industrial partners

Description

- **Analysis of big data streams**: A complete collection of algorithms for inference and learning of both static and dynamic Bayesian networks from streaming data. Existing software systems for PGMs only focus on stationary datasets.
- **Distributed parallel algorithms**: AMIDST provides parallel multi-core and distributed implementations of Bayesian parameter learning, using streaming variational Bayes and variational message passing.

Main Features

- Java 8 based
- Latent variable models
- Integration

Code example

```java
// We can store the SVB object
System.out.println(bnModel.toString());
// We print the model
BayesianNetwork bnModel = parameterLearningAlgorithm.getLearntBayesianNetwork();
// And we get the model
parameterLearningAlgorithm.runLearning();
// We perform the learning
parameterLearningAlgorithm.setDataStream(data);
// We can activate the output
parameterLearningAlgorithm.setDAG(DAGGenerator.getHiddenNaiveBayesStructure(data).
// We fix the DAG structure
parameterLearningAlgorithm.setWindowsSize(100);
// We can open the data stream using the static class DataStreamLoader
DataStream<DataInstance> data = DataStreamLoader.openFromFile("datasets/simulated/
// We create a SVB object
WasteIncineratorSample.arff");
```

Use-case: Risk prediction in credit operations

And much more...

AMIDST project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209.