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. Masegosa¹, Ana M. Martínez², Darío Ramos-López³, Helge Langseth¹,
Thomas D. Nielsen², Antonio Salmerón³, Rafael Cabañas² & Anders L. Madsen²,⁴

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

Presentation

Data mining frameworks

- Stationary data sets
  - Weka
  - R Libs
  - Matlab
  - Elvira
  - Apache
  - Inference
  - Spark/Flink

- Dynamic data sets
  - AMIDST
  - MOA
  - Elvira
  - Apache SamOA
  - Hugin
  - WEKA
  - R Libs

- Data streams
  - MLlib
  - Apache SAMOA
  - Vowpal Wabbit

Academic and Industrial partners

- DAIMLER
- NTNU
- UNIVERSIDAD DE ALMERÍA
- DAVID
- HUGIN

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

```
// Learn hidden naive Bayes model from data stream
DataStream<DataInstance> data = DataStreamLoader.openFromFile("datasets/simulated/WasteIncineratorSample.arff");

// We can open the data stream using the static class DataStreamLoader
ParameterLearningAlgorithm data = new ParameterLearningAlgorithm();

// We can set the output
parameterLearningAlgorithm.setOutput(true);

// We fix the window size
parameterLearningAlgorithm.setWindowsSize(100);

// We fix the structure
parameterLearningAlgorithm.setStructure("MarkovBlanketClassifiers", 2);

// We fix the size of the window
parameterLearningAlgorithm.setWindowSize(100);

// We can retrieve the output
System.out.println(bnModel.toString());

// We can create a new object
parameterLearningAlgorithm.runLearning();

// We can create a new object
parameterLearningAlgorithm.setDataStream(data);
```

Use-case: Risk prediction in credit operations

- Concept drift
- Correlated with Unemployment Rate
- 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.