

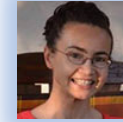
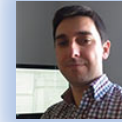
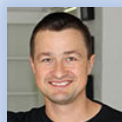
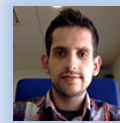
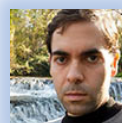
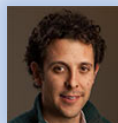
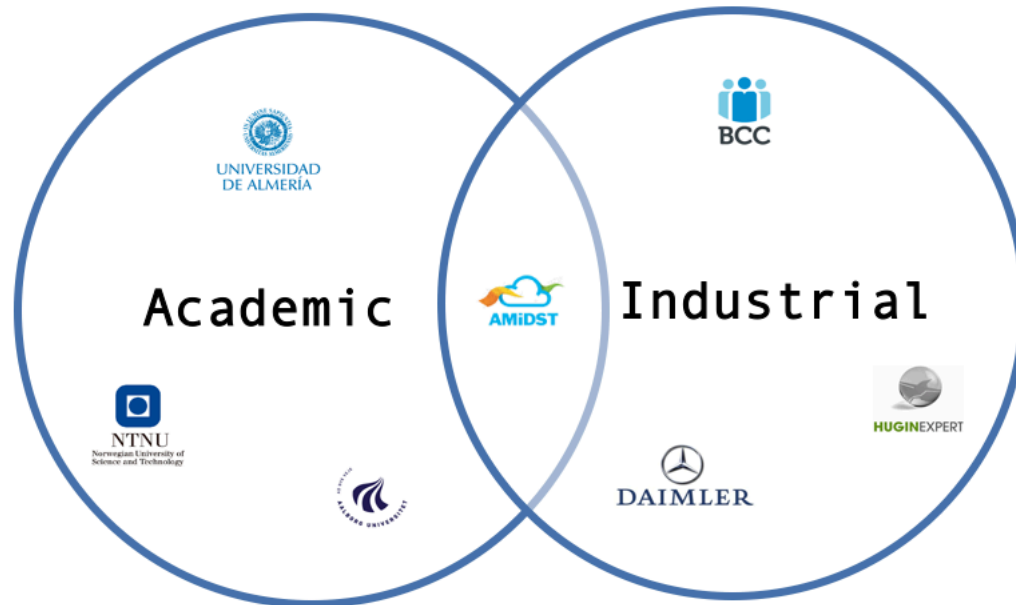
AMiDST TOOLBOX

A Java Toolbox for Scalable Probabilistic Machine Learning

Andrés R. Masegosa

Norwegian University of
Science and Technology
andres.masegosa@idi.ntnu.no

Who are we?

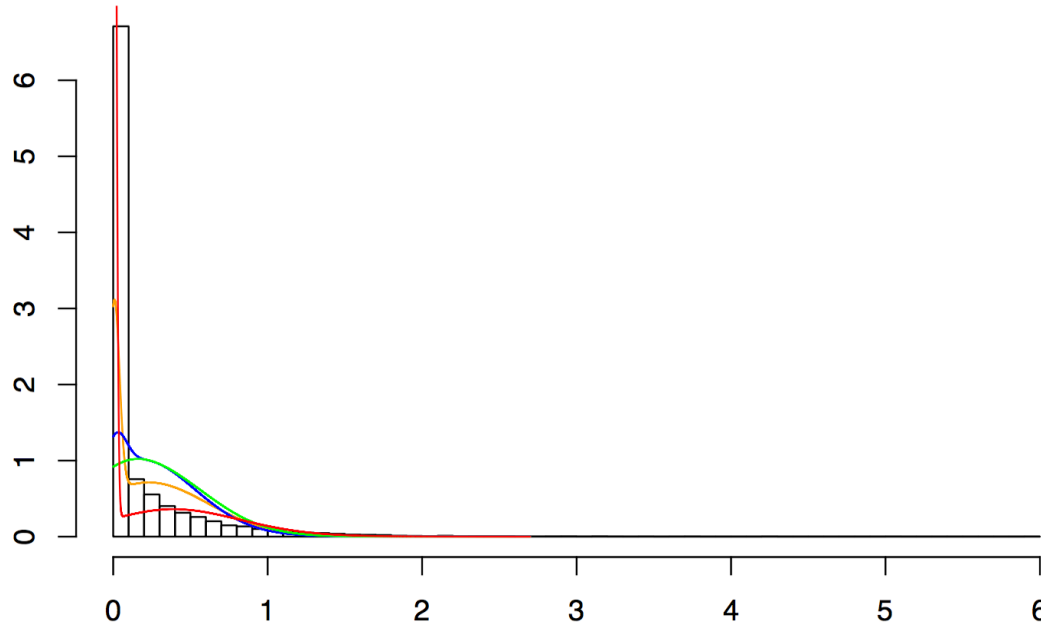


Running Use Case



Predicting Defaulting Clients

Predicts probability a customer will default within 2 years



- Daily data for millions of clients
- Tons of missing data.
- Odd distributions.

Toolbox presentation





Probabilistic machine learning

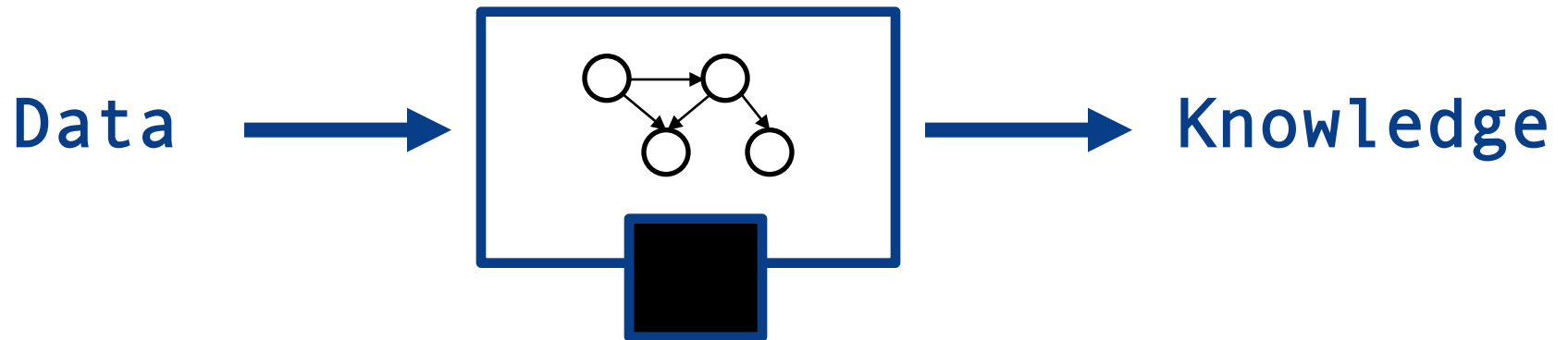
Model your problem using a flexible probabilistic language based on graphical models. Then, fit it with data using a Bayesian approach to handle modelling uncertainty.

Multi-core and distributed processing

AMIDST provides tailored parallel and distributed implementations of Bayesian parameter learning (and probabilistic inference) for batch and streaming data. This processing is based on flexible and scalable message passing algorithms.

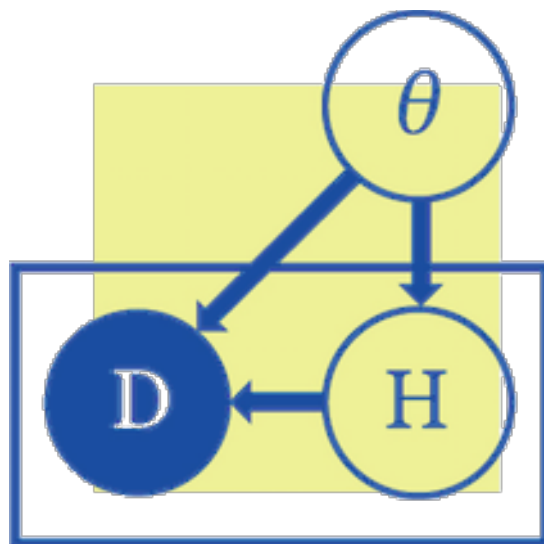


Openbox Models



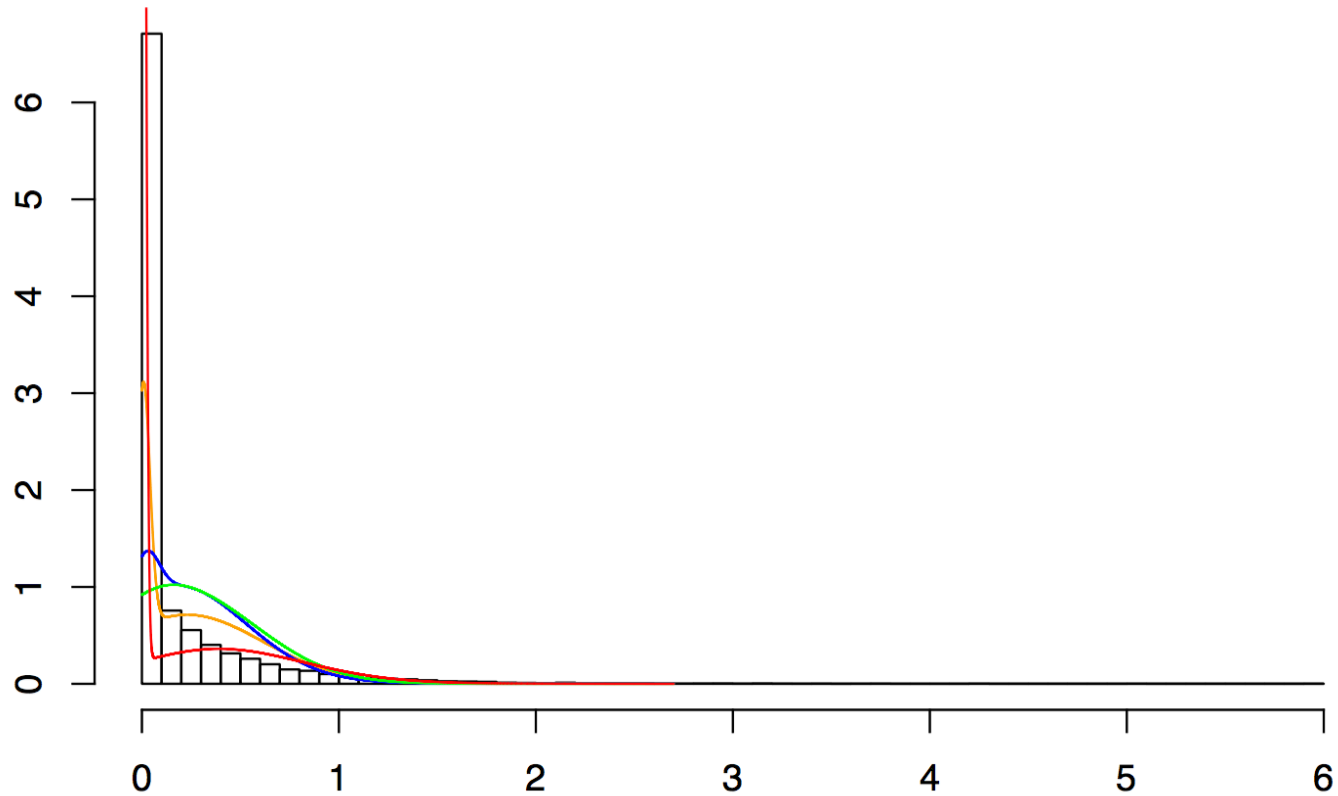
Blackbox Inference Engine
(Powered by Flink)

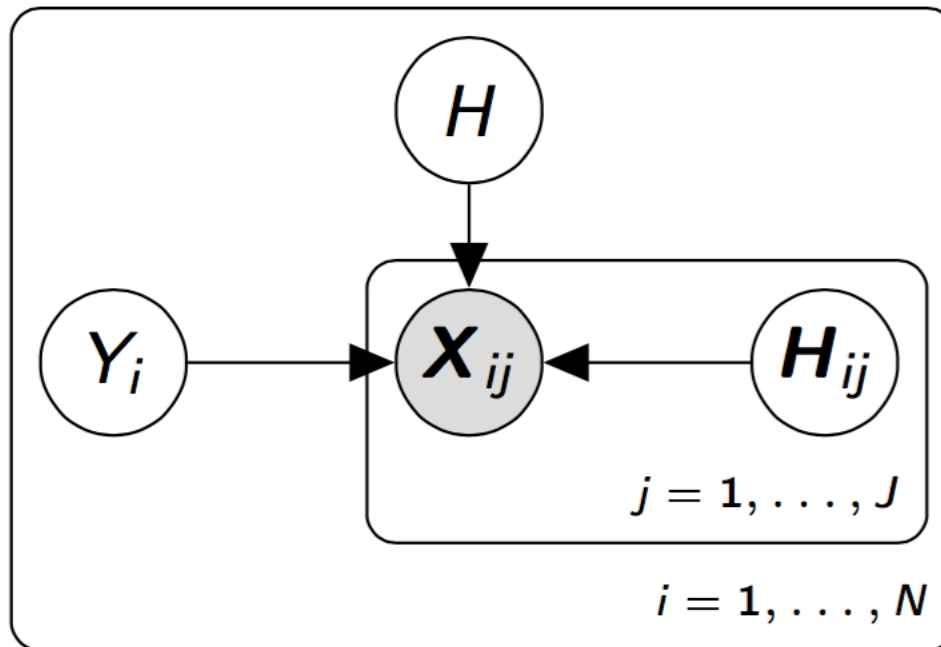
Main Features



Probabilistic graphical models (PGMs)

Specify your model using probabilistic graphical models with latent variables and temporal dependencies





Custom Gaussian Mixture Model

H_{ij} defines local mixture

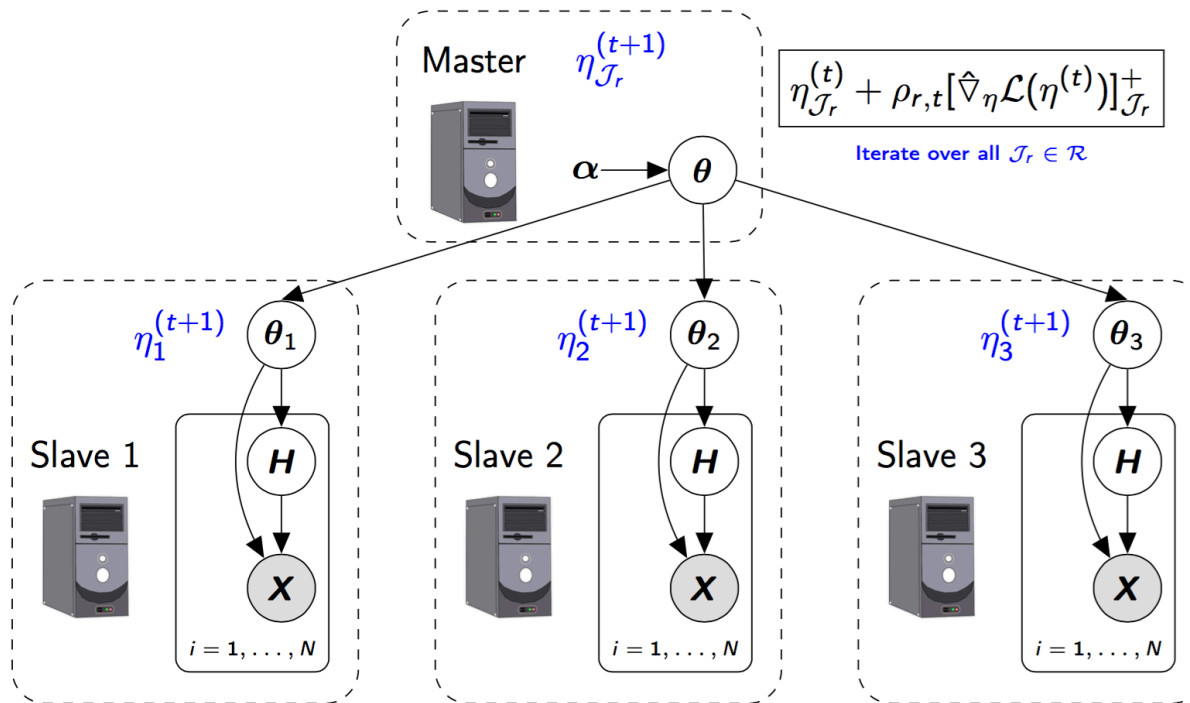
H_i defines a global mixture.

```
//Set-up Flink session.  
final ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();  
  
//Load the data stream  
String filename = "hdfs://dataFlink_month0.arff";  
DataFlink<DataInstance> data =  
    DataFlinkLoader.loadDataFromFolder(env, filename, false);  
  
//Build the model  
Model model = new CustomGaussianMixture(data.getAttributes());
```

$$P(\theta | \mathbf{D})$$

Scalable Learning

Perform Bayesian inference on your probabilistic models with powerful approximate and scalable algorithms.



d-VMP Algorithm

A state-of-the-art distributed variational message passing algorithm.


```
//Set-up Flink session.
final ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();

//Load the data stream
String filename = "hdfs://dataFlink_month0.arff";
DataFlink<DataInstance> data =
    DataFlinkLoader.loadDataFromFolder(env, filename, false);

//Build the model
Model model = new CustomGaussianMixture(data.getAttributes());

//Learn the model
model.updateModel(data);
```





01011100

Data Streams

Update your models when new data is available. This makes our toolbox appropriate for learning from data streams.

```
//Set-up Flink session.
final ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();

//Load the data stream
String filename = "hdfs://dataFlink_month0.arff";
DataFlink<DataInstance> data =
    DataFlinkLoader.loadDataFromFolder(env, filename, false);

//Build the model
Model model = new CustomGaussianMixture(data.getAttributes());

//Learn the model
model.updateModel(data);

//Update your model
for(int i=1; i<12; i++) {
    filename = "dataFlink_month"+i+".arff";
    data = DataFlinkLoader.loadDataFromFolder(env, filename, false);
    model.updateModel(data);
}
```





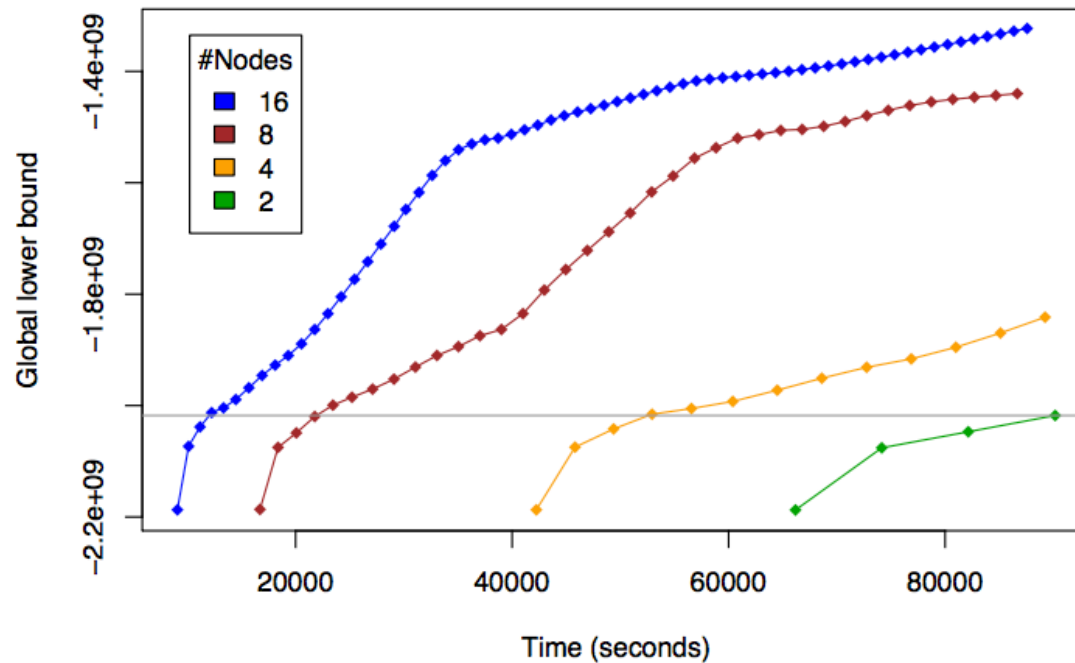
Predicting Defaulting Clients

- Old BCC's models based on logistic regression got an AUC of 0.816
- AMIDST's models gets an AUC of 0.952.



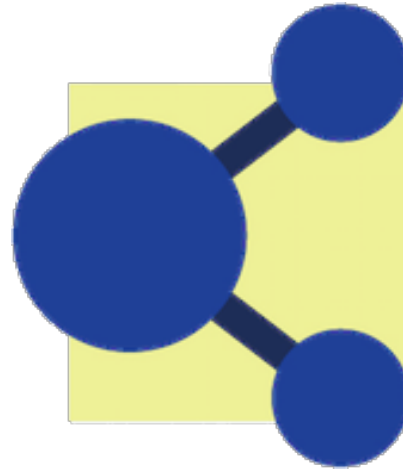
Large-Scale Data

Use your defined models to process massive data sets in a distributed computer cluster using Flink and Spark.



One billion node probabilistic model

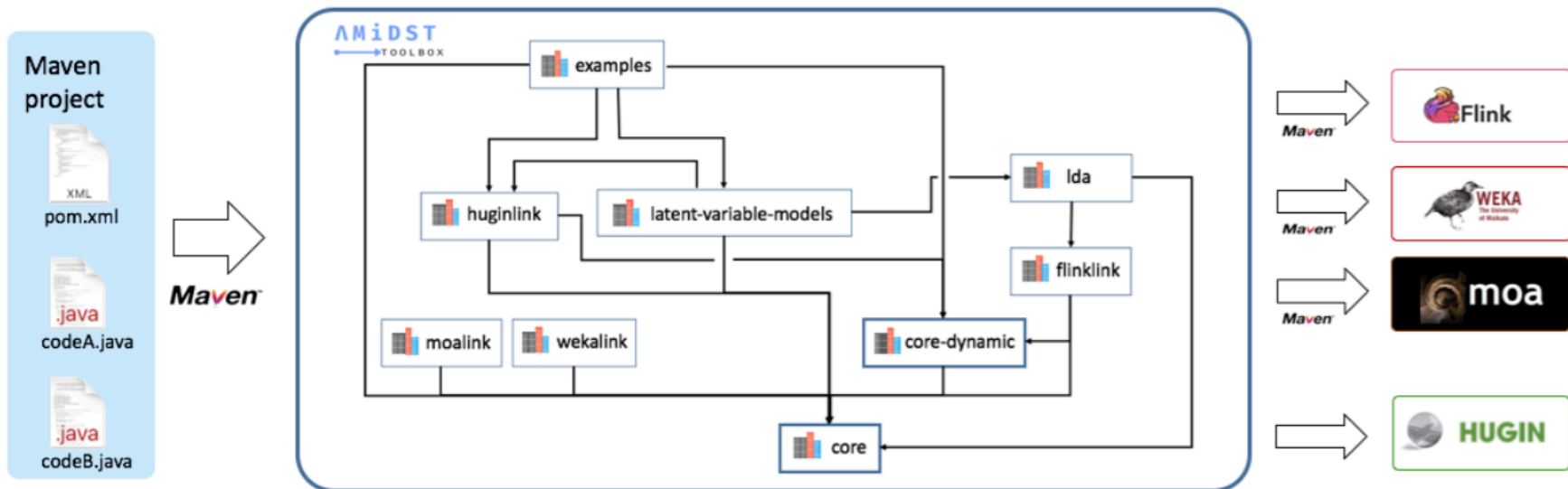
Experiment on an Flink cluster with 16 nodes on AWS.



Modular Design

The AMIDST Toolbox has been designed following a modular structure. This makes easier:

- The maintenance and enhancement of the software
- The integration with external software: HUGIN, MOA, Weka, R.

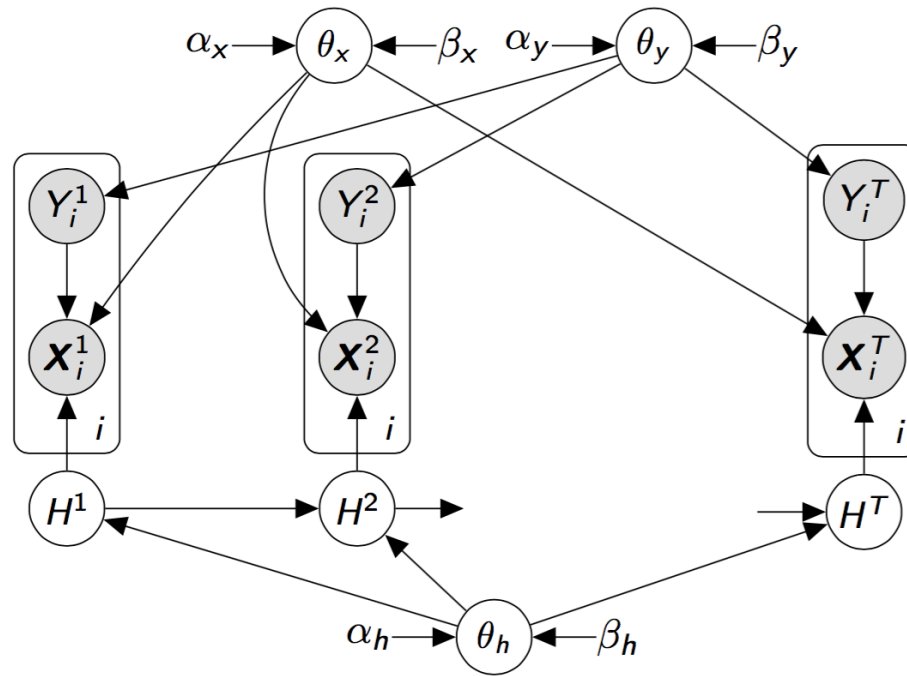


Running Use Case II



Tracking Concept Drift

Detects changes in customer profiles during Spanish financial crisis



Hidden Variables are used to capture changes in customer profile

```
//Set-up Flink session.
final ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();

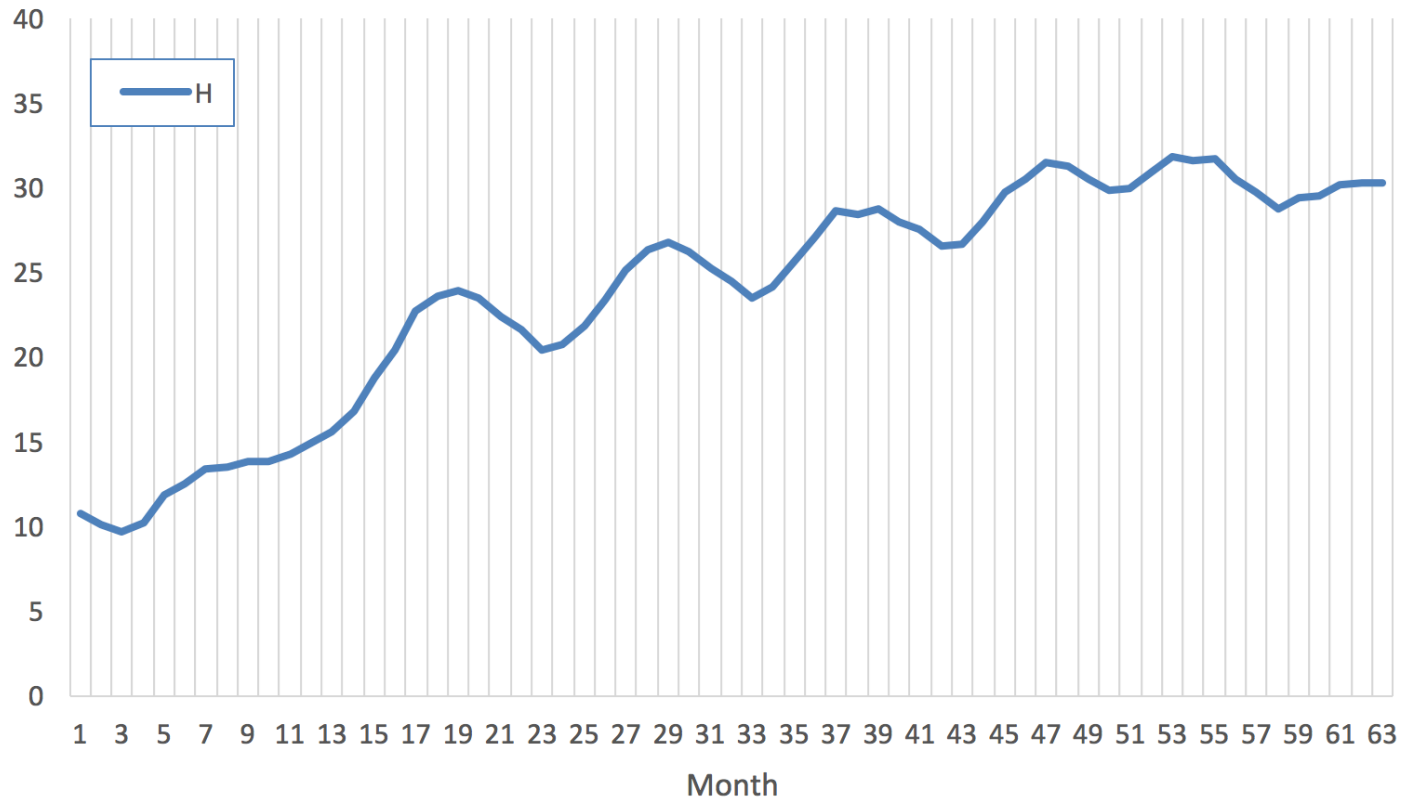
//Load the data stream
String filename = "hdfs://dataFlink_month0.arff";
DataFlink<DataInstance> data =
    DataFlinkLoader.loadDataFromFolder(env, filename, false);

//Build the model
Model model = new ConceptDriftDetector(data.getAttributes());

//Learn the model
model.updateModel(data);

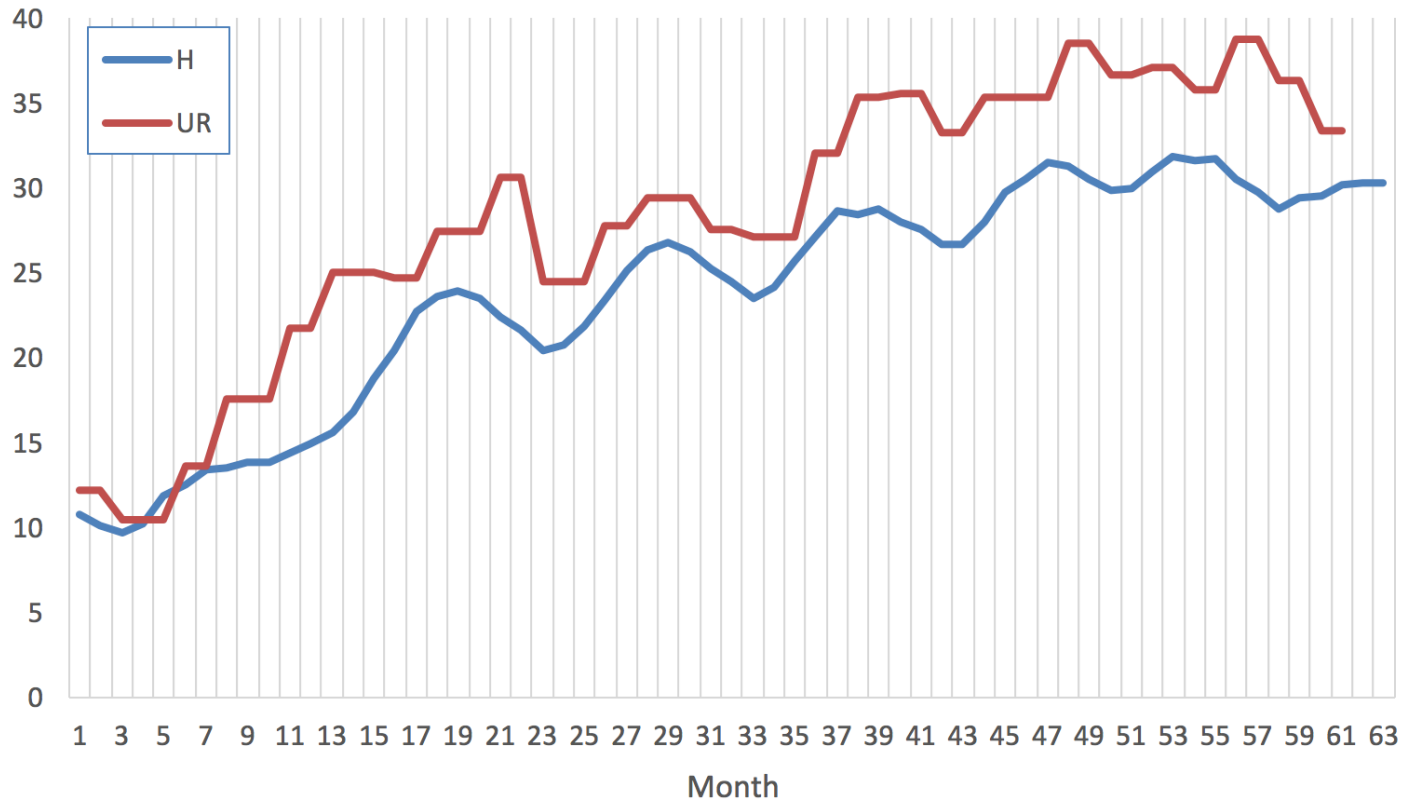
//Update your model
for(int i=1; i<12; i++) {
    filename = "dataFlink_month"+i+".arff";
    data = DataFlinkLoader.loadDataFromFolder(env, filename, false);
    model.updateModel(data);
    System.out.println(model.getPosteriorDistribution("hiddenVar").toString());
}
```





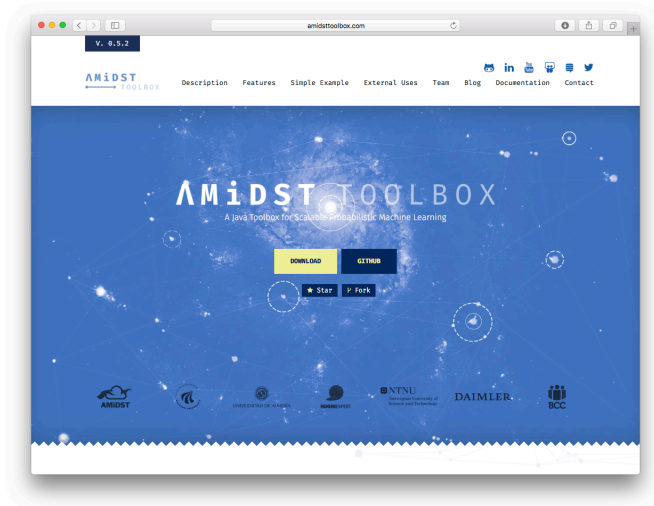
Hidden Variable Captures Concept Drift

Drift Pattern: Seasonal + Global trend

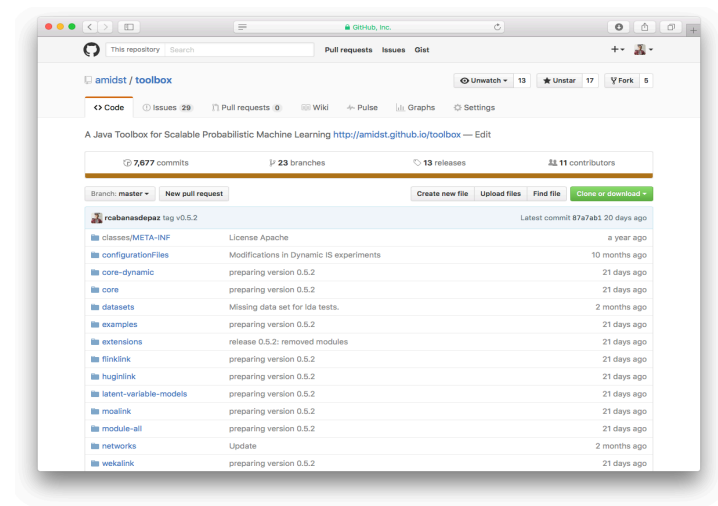


Unemployment Rate main driver of Concept Drift

Hidden Variable correlates with unemployment rate ($\rho = 0.961$)



www.amidsttoolbox.com



github.com/amidst/toolbox



Apache
License 2.0

Thanks for your attention



www

www.amidsttoolbox.com



@

contact@amidsttoolbox.com



[@AmidstToolbox](https://twitter.com/AmidstToolbox)

AMiDST
→ TOOLBOX