|  |  |  |  |
| --- | --- | --- | --- |
| **Module Title** | Big Data Analytics | **Course** | MSc Advanced Computer Science |
|  | | **Date** | 12th February 2020 |
| **Term** | Spring term 2020 | **Lab leader** | Dr. Frederic Stahl |
| **Content** | Data Stream Classification with MOA | | |

**Lab 5**

**Author: Dr Frederic Stahl**

**Task 1: Calculating the adequate window size for rule/decision tree induction**

Assume your data in the stream consists of 5 classes, you have currently buffered 200 data instances for expanding the root node, and there are 3 possible attributes to use for the expansion. Use the Hoeffding Bound to assess if the 200 data instances are enough to expand the root node. The formula for the Hoeffding Bound is given below:

Assumptions:

* The decision tree induction algorithm uses the information gain as splitting criterion => range R = log(c), where c is the number of classes, in this case 5.
* User defined classification error = .

|  |  |
| --- | --- |
| **Attribute** | **Information Gain** |
| A1 | 0.82 |
| A2 | 0.77 |
| A3 | 0.43 |

Task 1.1: Calculate the Hoeffding Bound  and . If  > , then the Hoeffding bound guarantees that within a probability of *1-δ* the true - . You can use the free space below to write down your calculations:

= 0.14030896062

Task 1.1: State and reason if there is enough data available (if *n* is big enough), to use the buffer to expand the root node.

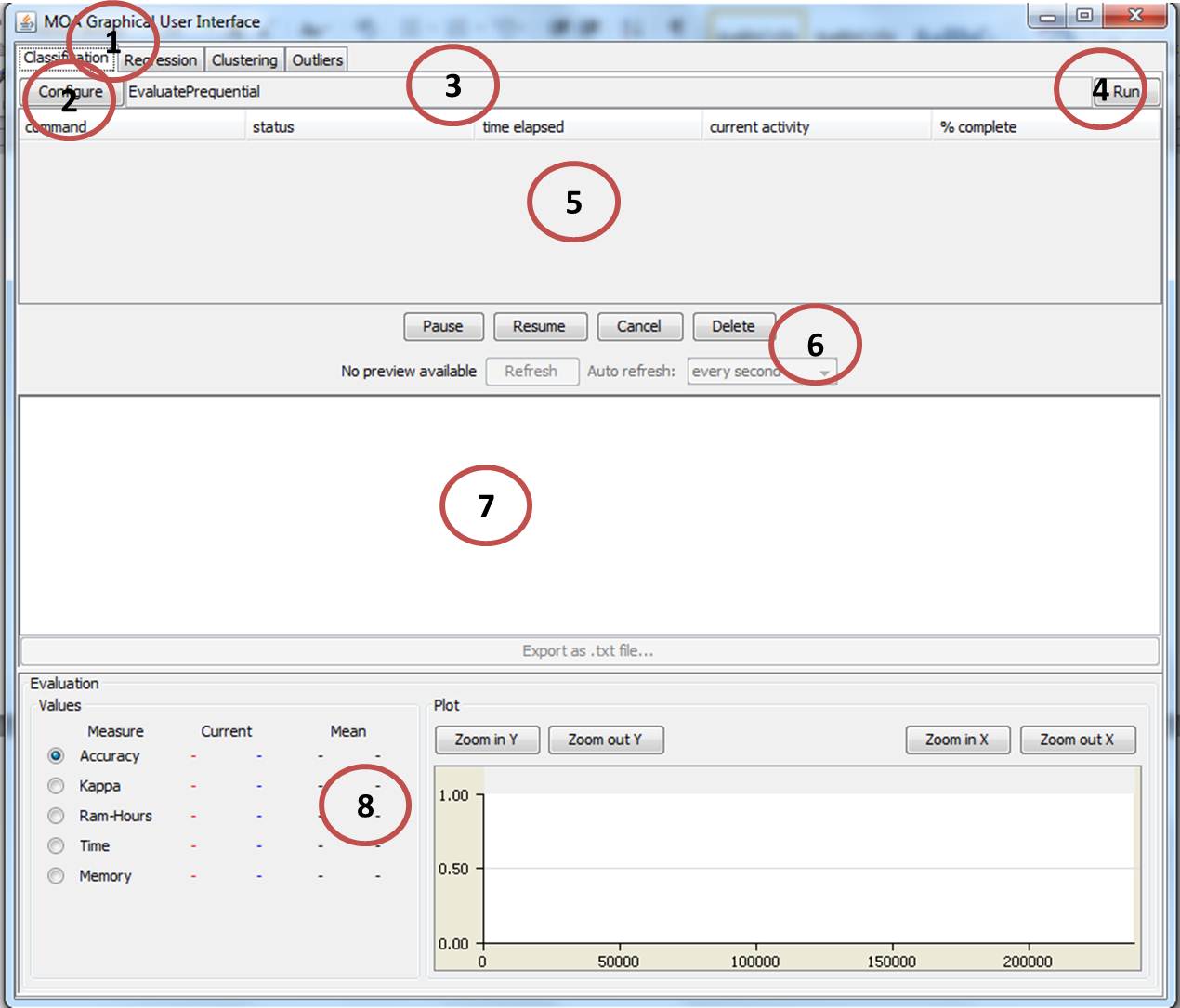
There is not enough data as

**Task 2: Finding our way with the Massive Online Analysis tool (MOA) GUI**

For this this lab we are using the MOA tool. MOA is an open source experimenter that provides implementations of various data stream mining algorithms. It also provides a simple interface to integrate your own algorithms; however, this is not subject to this lab session. For experimental purposes MOA also provides data stream generators on which you can directly apply the algorithms implemented in MOA. MOA is already installed on your computer; however, you can download your own copy from the MOA website <http://moa.cms.waikato.ac.nz/>.

Following is a quick tutorial to explain the basic MOA interface, however, you are encouraged to explore more by yourself. Here is also an introductory video presenting MOA <http://videolectures.net/wapa2010_bifet_maof/>.

Start MOA, you should see the GUI highlighted below with the Classification tab selected.



The elements numbered in the GUI are explained below:

*Element 1*: There are 4 tabs, one for classification, one for clustering, one for regression and one for detecting outliers, essentially providing 4 different views of the GUI. Here the classification tab is selected.

*Element 2*: The configuration button opens in a new window a menu for configuring a new data stream classification task.

*Element 3*: Shows the command line version of the classification task when selected using the configuration button (element 2) that can be used to run the task without the GUI (we won’t use command line in this tutorial).

*Element 4*: The run button starts the classification task configured.

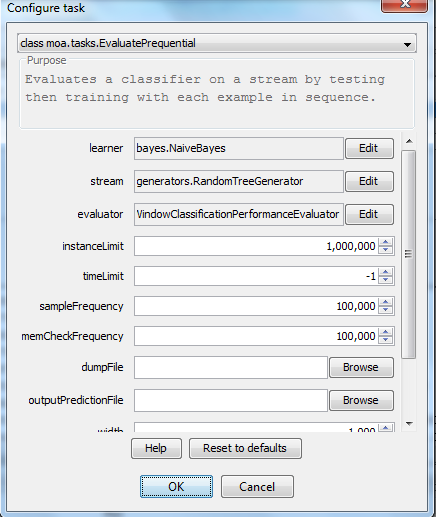
*Element 5*: Lists all the configuration tasks currently being executed or that have been executed in the past. It is possible to highlight tasks with the mouse and use the control buttons (element 6) to manipulate the tasks.

*Element 6*: Some control buttons to resume, stop, cancel or delete classification tasks that you have highlighted in element 5 above.

*Element 7*: Shows the command line output of the in element 5 selected classification task.

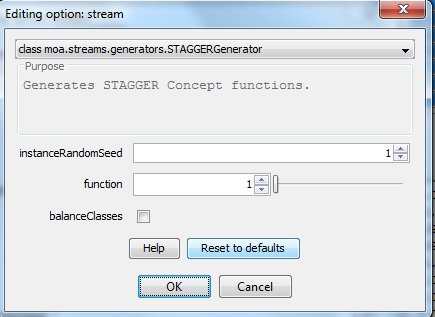
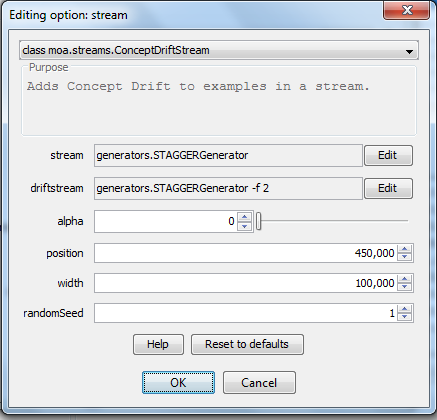
*Element 8*: Shows a selection of evaluation measures that are displayed graphically over time (or the number of data instances processed). By default the accuracy is selected. It is possible to compare the results of two different tasks: the current task is displayed in red, and the previously selected one is displayed in blue.

**Task 3: Data Stream Classification Configuration.**

Click Configure to open the configuration dialog as shown in the figure on the left. On the top you can select the evaluation method, please select “EvaluatePrequental” (this is the sliding window approach). Further select as learner “HoeffdingTrees” (this dis also known as Very Fast Decision Trees). A new window for “HoeffdingTrees” opens, leave the default settings as they are and click ok. Set the stream to “STAGGERGenerator”. A new window for STAGGER opens, leave the default settings as they are and click ok. Next set the instance limit to 1,000,000. Klick “ok” and then “run”. The task should appear in GUI element 5 and element 8 should display some performance measures in real time.

Next click configure again, use the same setting but this time choose “NaiveBayes” as learner. Yes, Naïve Bayes is a batch learning algorithms, however, it is also naturally incremental and thus applicable to data streams. Again, click ok and run the task. Now compare Naïve Bayes with Hoeffding Trees. What are your findings, which classifier performs better?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we will see how to run a classification task on a stream with concept drift. Again click on configuration. Select as learner “HoeffdingTrees”, as evaluation metric “EvaluatePrequental”, limit the maximum number of instances to 1 million, and then select as stream “ConcepDriftStream”. There are stream generators that automatically generate concept drifts, however, this stream allows to define the concept drift manually by merging two different streams. Once you selected the “ConcepDriftStream” you’ll be presented with the dialog on the right. Click on edit for the “stream” field. On the top select the “STAGGERGenerator” and the dialog displayed below should open. Select function 1, leave the rest as it is and click ok. Back in the “ConcepDriftStream” dialog click on edit for the “driftstream” field, again select the “STAGGERGenerator” this time select function 2, leave the rest as it is and click ok. Back in the “ConcepDriftStream” dialog set “position” to 450000 and width to 100000.

These setting basically run a datastream using the STAGGERGenerator with function 1 up to data instance number 450000. From then onwards data instances 450001 to 550000 contain instances generated by both, STAGGERGenerator function 1 and 2. From instances 550001 to 1000000 only instances from STAGGERGenerator using function 2 will be in the stream. Set up another stream classification task with the same data stream settings, but this time using Naïve Bayes. Regarding the evaluation metrics, which classifier performs better in your opinion and why?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Repeat the experiments above, this time use a sudden concept drift (selecting width of 1 in the “ConcepDriftStream” dialog). Which classifier performs better in your opinion and why?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Repeat the experiments above for sudden and gradual concept drift but this time use the SEAGenerator with functions 1 and 2. What are your findings for Hoeffding Tree and Naïve Bayes?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now play around, using different Stream generators, algorithms, evaluation methods and settings. Please note that not all algorithms and streams and drifts are compatible. If this is the case, then you will most likely see a Java exception in element 5. You may also want to choose a different evaluation method, i.e. you may want to try “EvaluateInterleavedTestThenTrain”, which is literally the Interleaved Test-Then-Train approach and compare with other approaches. Also you could try to combine two different data streams, not just different versions of the same stream generator.

Also you may have noticed that one of the algorithms does not cope well with the concept drift on one of the data streams. But will it recover if the concept in the datastream stays stable for a long time? Try it out!