Data Mining

Data-Stream: Stream Classification

Roc Granada Verdú, Jonas Grab, Christian Ley, Christian Stricker

For this assignment we used prequential evaluation for stream classification:

- Use same data for testing and training.
- The error of the model is computed form the sequence of examples.
- For each example in the stream, the actual model makes a prediction based only on the example attribute-values.

We add an IntelliJ project ready to run as well as the java classes separated in case runing the code is not needed. Plots for all the datasets using HoeffdingTree and NaiveBayes are also added. There are no plots for the AODE classifier since we could not make it work.

Problems with AODE

We tried everything to be able to use AODE with the MOA api but we did not manage to. The weka wrapper for MOA just works in the direction MOA → WEKA, so all the methods from weka are available in a moa classifier. However, weka does not allow Prequential Evaluation, so the wrapper we would need is WEKA → MOA which we did not find. We also tried to adapt the weka classifier with the Java Example Code of moa for prequential evaluation, but it did not work, since there are some methods needed not supported in the weka classifier or weka Instances.

Plots

We can appreciate how for smaller windows we get better results, but those results are not that much trustworthy since are very unstable. In some periodes of Instances the accuracy is very high but in others is very low. So, even if the best accuracies are worst, we should rely more on the runs with bigger windows that are more stable. Specially with the naive bayes, which gets the worst results with small windows.

Since the Electricity dataset is very small, we could not make plots. It needs just one instance batch to compute all the data.

Example runing java code with Window 1000 dataset covTypeNorm

```
Evaluate prequential using Hoeffding Tree
learning evaluation instances,evaluation time (cpu seconds),model cost (RAM-Hours),classified
instances,classifications correct (percent),Kappa Statistic (percent),Kappa Temporal Statistic
(percent),Kappa M Statistic (percent),model training instances,model serialized size (bytes),tree
size (nodes),tree size (leaves),active learning leaves,tree depth,active leaf byte size estimate,inactive
leaf byte size estimate,byte size estimate overhead
100000.0,3.525772982,0.0,100000.0,87.0,58.643507030603814,-
233.33333333333303,36.27450980392155,100000.0,0.0,41.0,21.0,21.0,9.0,0.0,0.0,1.0
200000.0,5.992280784,0.0,200000.0,94.39999999999999,88.96286397912404,-
75.0,89.94614003590662,200000.0,0.0,163.0,82.0,82.0,14.0,0.0,0.0,1.0
300000.0,8.947420218,0.0,300000.0,77.0,68.01975544846036,-
422.72727272727224,68.79240162822252,300000.0,0.0,213.0,107.0,107.0,15.0,0.0,0.0,1.0
400000.0,11.864350091,0.0,400000.0,69.1,53.713273954504196,-
586.6666666666662,36.02484472049688,400000.0,0.0,255.0,128.0,128.0,15.0,0.0,0.0,1.0
500000.0,14.757290957,0.0,500000.0,70.6,50.22946923283974,-
525.5319148936165,37.17948717948717,500000.0,0.0,297.0,149.0,149.0,18.0,0.0,0.0,1.0
```

```
581012.0,17.167508715,0.0,581012.0,95.39999999999999,81.32813768468907,-130.0,65.15151515151513,581012.0,0.0,339.0,170.0,170.0,19.0,0.0,0.0,1.0
```

Evaluate prequential using Naive Bayes

```
learning evaluation instances,evaluation time (cpu seconds),model cost (RAM-Hours),classified instances,classifications correct (percent),Kappa Statistic (percent),Kappa Temporal Statistic (percent),Kappa M Statistic (percent),model training instances,model serialized size (bytes)
100000.0,2.28639662,0.0,100000.0,81.5,50.33677038675574,-374.3589743589741,9.313725490196035,100000.0,0.0
200000.0,4.537639321,0.0,200000.0,67.30000000000001,42.622821377999806,-921.874999999999,41.29263913824058,200000.0,0.0
300000.0,6.788972156,0.0,300000.0,56.10000000000001,41.77657057369309,-897.7272727272717,40.43419267299865,300000.0,0.0
400000.0,9.042086898,0.0,400000.0,48.9,29.410432060649015,-1035.5555555555545,-5.797101449275368,400000.0,0.0
500000.0,11.303563864,0.0,500000.0,46.800000000000004,17.846338680013048,-1031.9148936170202,-13.675213675213676,500000.0,0.0
581012.0,13.160730182,0.0,581012.0,79.2,47.75339602925809,-939.999999999999,-57.575757575757535,581012.0,0.0
```