Data Stream Challenge Report

Kosar Najjari Hagh*
najjarihagh@gmail.com

ABSTRACT

In this work, we generated data sets using the SEA generator with various noise percentages; then, we classified these sets using individual and ensemble classifiers in the form of online classification. Furthermore, we compared the results, first in terms of the accuracy of different online models and second, the performance comparison of online models. At the end, we try to improve the overall performance of the online models.

KEYWORDS

data stream, neural networks, statistical learning, online classifiers.

1 INTRODUCTION

Data streams with 10,000 instances are generated by the SEA generator with no noise, 10 and 70 percentage of noise, respectively. We wrote the generated data in files, then we constructed and trained Hoeffding Tree (HT), K-Nearest Neighbor (KNN), and Multi-Layer Perceptron (MLP), composed of four layers of 200 perceptrons (50 perceptron in each layer) classifiers using these data sets. After this, we trained online ensemble classifiers such as MV and WMV that combine the above-mentioned single classifiers. We tested the performance of classifiers using the interleaved-test-then-train approach. Based on the results that we acquired, we compared the performance of classifiers in terms of their temporal accuracy. We determined that, in some cases ensemble models work better. Furthermore, we briefed the overall accuracy performance of online and batch models. Finally, we tried a way to improve the performance of some of the classifiers.
We wrote our scripts in Python (Anaconda), using the scikit-multiflow package.

2 DATA STREAM GENERATION

In this section, we generate three data sets with 10,000 samples consisting of 3 features and 2 class labels, with no noise, 10% of noise and 70% of noise, respectively, using "`skmultiflow`". We generate 3 data sets with 10,000 rows and 4 columns. We demonstrate the data set with no noise as an example in table 1 .

3 TRAINING SINGLE AND ENSEMBLE ONLINE CLASSIFIERS

In this section, we construct and train all of the single models such as HT, KNN, and MLP and also write two ensemble methods, Majority Voting (MV) and Weighted Majority Voting (WMV) from the scratch. We provide the results in figures 1 to 3.

Table 1: Data set without noise

0	1	2	3	
0	5.434049	2.783694	4.245176	1
1	0.047189	1.215691	6.707491	0
2	1.367066	5.750933	8.913220	0
3	1.853282	1.083769	2.196975	0
4	8.116831	1.719410	8.162247	1
...	...	...	...	...
9995	6.866050	3.723009	6.374975	1
9996	0.310384	4.204737	6.196646	0
9997	5.962235	9.280850	3.853130	1
9998	3.162848	8.371640	7.288275	1
9999	7.163179	6.130151	2.635034	1

4 COMPARISON OF MODELS

In this section we provide our results and compare them, in answering the questions of the assignment.

4.1 Comparison of Temporal Accuracy of Online Classifiers

Based on the plotted results in this section, we can compare the temporal accuracy of five classifiers.

As we observe in figure 1, when the data stream has no noise, the accuracy of KNN is higher and MLP's accuracy is lower than the others. Furthermore, we can point out that the convergence rate of KNN is the highest and MLP has the lowest convergence rate. This means that in no noise case, MLP needs more data samples to reach a stable predicting accuracy.

In 10% noise data stream's case we can observe in figure 2 that when learners converge, WMV has the best accuracy and MLP has the lowest accuracy among the classifiers. Besides, MV and WMV have the highest convergence rate, but MLP has poorest convergence rate. Moreover, HT classifier has relatively high over-fitting. At first, it has the highest accuracy among all of the models but eventually it converges with relatively low accuracy. Also, we can observe here that MLP has more over-fitting than what it was in no noise case.

In figure 3, for data set with noise percentage of 70, MV and HT classifiers have the highest accuracy when they converge, on the other hand, KNN is poor in terms of accuracy here. The convergence rate of all classifiers are high in this case, however, MLP classifier has relatively low convergence rate. In this case too, over-fitting is noticeably high in all of the classifiers. MLP classifier has the highest over-fitting among the others.

*Fall 2021 Semester Applicant

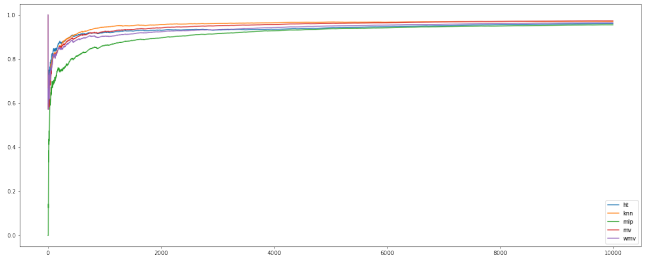

Figure 1: Accuracy results of classifiers with no noise data stream

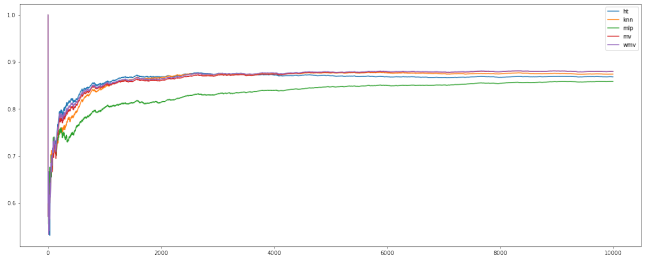

Figure 2: Accuracy results of classifiers with 10% noise data stream

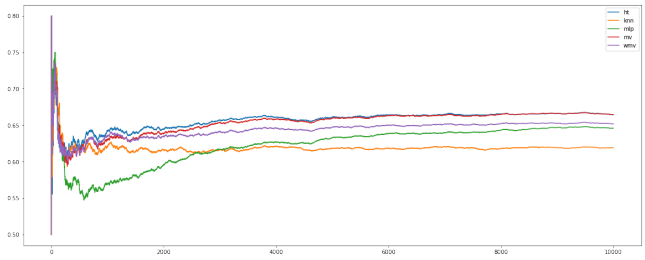

Figure 3: Accuracy results of classifiers with 70% noise data stream

4.2 Ensemble methods vs. Individual models

As figure 4 suggests, we can figure out that if single models have below 50% error rates, it would be better to apply an ensemble method because this can decrease the error rate, even more. Otherwise (with error rate greater than 50% in single classifiers), utilizing ensemble methods can be the best choice. We just need to notice that, since in using ensemble models actually multiple single classifiers are utilized instead of one classifier, it has more time complexity and needs more memory usage.

4.3 Comparison of Online and Batch Models

We can understand that in offline models since classifiers are trained with available data sets at once, they would have better accuracy in cases that convergence rate is low, in comparison to the online models which are trained with each data sample incrementally. In

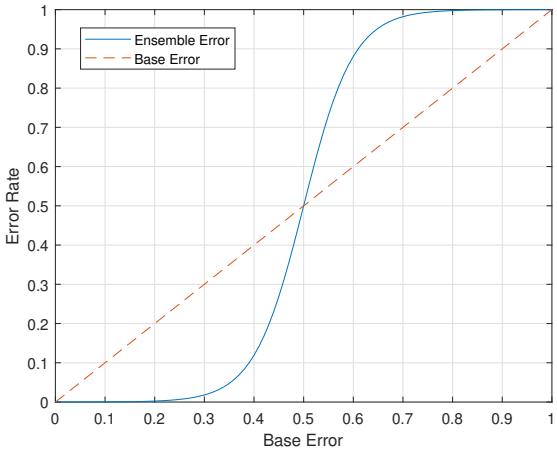

Figure 4: Comparison of Ensemble and Base errors [1]

online models, there is the possibility that **over-fitting** happens in the primary training intervals when not enough data samples are available. Furthermore, **concept drift** is a problem that occurs only in online models and degrades the accuracy of classification. So because of this, in some cases, we need to refresh the online model over time and take context into consideration, although the effect of concept drift can never be fully nullified.

5 A METHOD FOR IMPROVING PREDICTION ACCURACY

In this part, in order to improve the prediction accuracy we suggest to add Gaussian noise to the data streams. We tested this scheme on MLP classifier and as we note in figure 5, this scheme is working well in MLP. In other words, when we have fewer data samples, it reduces the over-fitting. However, this scheme is working effectively only on MLP.

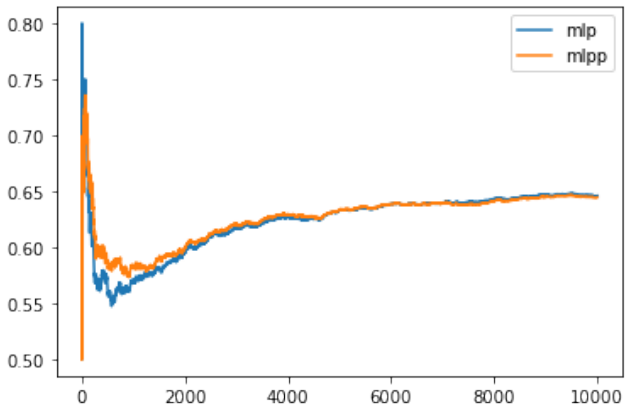

Figure 5: Result of MLP classifier for data with added Gaussian Noise

6 EFFICIENCY COMPARISON

In this section we compare the classifiers in terms of their lapsed time and peak memory usage. For each classifier, we run the time profiler seven times and average the results. The memory profiler depicts the highest level of memory usage (peak memory usage) through the execution of the corresponding classifier.

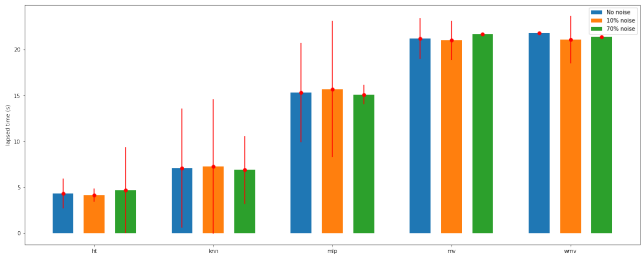

Figure 6: Time Profiler

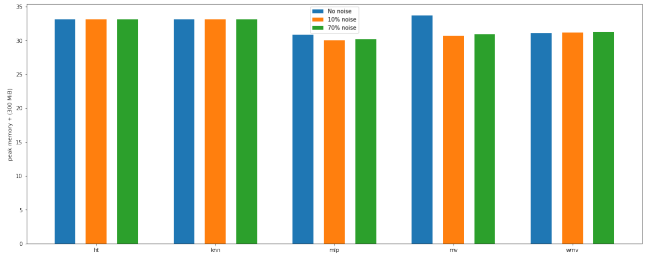

Figure 7: Memory Profiler

REFERENCES

[1] Pengyi Yang, Yee Hwa Yang, Bing B Zhou, and Albert Y Zomaya. 2010. A review of ensemble methods in bioinformatics. *Current Bioinformatics* 5, 4 (2010), 296–308.