Sakshi - 1001993702   
Varsha Nanajipuram - 1002039829   
Likitha Veluri - 1002025528   
Sandeep Penumuchu - 1002029163

Dr. Ramakrishna Koganti

CSE-5301 Data Analysis and Modelling Techniques

29/04/2022

Underwater Sonar Signals Recognition by Incremental Data Stream Mining with Conflict Analysis

Abstract

Sonar signal recognition is a dominant task in detecting the presence of some notable objects under the sea. In the world’s transportation sector, a lot of businesses depend on armed forces to extensively use sonar signals in place of visuals to navigate underwater and detect enemy submarines in proximity. But various unpredictable factors such as weather conditions or receiving an unknown object that is not yet tracked may deteriorate the impact.

Introduction

Sonar signals contain information about objects (submarines or new objects) detected underwater and sometimes sonar signals may contain noise data which contains information about ROCKS, Hard SAND, etc. By using a pairwise classifier, we can remove such noise data from sonar signals, and then using traditional machine learning classifiers such as SVM, Decision Tree, and Naïve Bayesian can be used to train with clean data. Traditional classifier models can be used to classify new signals belonging to which class such as sonar submarine or ROCK.

Problem Statement

The main problem here is sonar signals come to the classifier in the form of continuous streams and classifiers use the BATCH mode technique to retrain themselves with new data by applying to pre-process (a technique to remove noise data using a pairwise classifier that removes un-similar data and used only similar data to retrain classifiers). After applying pre-processing technique classifiers will use batches to train themselves and while training classifier will remove those models whose prediction accuracy is not better and continue this process till the classifier obtained a better accuracy model.

Objectives

The main objective of the project is to detect underwater rocks and hard sand by using sonar signal data, which may contain noise data from sonar signals. We can remove noisy data from sonar signals with the help of a pairwise classifier by using traditional machine learning classifiers such as SVM, Neural Networks, and Naïve Bayesian.

Dataset:

In this project, we have used the SONAR dataset from the UCI Machine learning website, and we are also using the same dataset. This dataset contains sonar signals related to submarines and ROCKS. Below are some examples from the dataset]

0.0200,0.0371,0.0428,0.0207,0.0954,0.0986,0.1539,0.1601,0.3109,0.2111,0.1609,0.1582,0.2238,0.0645,0.0660,0.2273,0.3100,0.2999,0.5078,0.4797,0.5783,0.5071,0.4328,0.5550,0.6711,0.6415,0.7104,0.8080,0.6791,0.3857,0.1307,0.2604,0.5121,0.7547,0.8537,0.8507,0.6692,0.6097,0.4943,0.2744,0.0510,0.2834,0.2825,0.4256,0.2641,0.1386,0.1051,0.1343,0.0383,0.0324,0.0232,0.0027,0.0065,0.0159,0.0072,0.0167,0.0180,0.0084,0.0090,0.0032,R

0.0453,0.0523,0.0843,0.0689,0.1183,0.2583,0.2156,0.3481,0.3337,0.2872,0.4918,0.6552,0.6919,0.7797,0.7464,0.9444,1.0000,0.8874,0.8024,0.7818,0.5212,0.4052,0.3957,0.3914,0.3250,0.3200,0.3271,0.2767,0.4423,0.2028,0.3788,0.2947,0.1984,0.2341,0.1306,0.4182,0.3835,0.1057,0.1840,0.1970,0.1674,0.0583,0.1401,0.1628,0.0621,0.0203,0.0530,0.0742,0.0409,0.0061,0.0125,0.0084,0.0089,0.0048,0.0094,0.0191,0.0140,0.0049,0.0052,0.0044,R

0.0047,0.0059,0.0080,0.0554,0.0883,0.1278,0.1674,0.1373,0.2922,0.3469,0.3265,0.3263,0.2301,0.1253,0.2102,0.2401,0.1928,0.1673,0.1228,0.0902,0.1557,0.3291,0.5268,0.6740,0.7906,0.8938,0.9395,0.9493,0.9040,0.9151,0.8828,0.8086,0.7180,0.6720,0.6447,0.6879,0.6241,0.4936,0.4144,0.4240,0.4546,0.4392,0.4323,0.4921,0.4710,0.3196,0.2241,0.1806,0.0990,0.0251,0.0129,0.0095,0.0126,0.0069,0.0039,0.0068,0.0060,0.0045,0.0002,0.0029,M

0.0201,0.0178,0.0274,0.0232,0.0724,0.0833,0.1232,0.1298,0.2085,0.2720,0.2188,0.3037,0.2959,0.2059,0.0906,0.1610,0.1800,0.2180,0.2026,0.1506,0.0521,0.2143,0.4333,0.5943,0.6926,0.7576,0.8787,0.9060,0.8528,0.9087,0.9657,0.9306,0.7774,0.6643,0.6604,0.6884,0.6938,0.5932,0.5774,0.6223,0.5841,0.4527,0.4911,0.5762,0.5013,0.4042,0.3123,0.2232,0.1085,0.0414,0.0253,0.0131,0.0049,0.0104,0.0102,0.0092,0.0083,0.0020,0.0048,0.0036,M

All above decimal values are the sonar signals from the UCI dataset and the last column value in the above examples are R and M. R means those signals received from ROCK and M means those signals received from submarine.

We converted SONAR signals same dataset into three parts and consider each part as one stream then we will train a classifier with the first stream and then pass the second stream to allow the classifier to retrain itself with a new stream and the same process continues for the third stream. In our application random, 20 records were used for the first stream and 40 for the next stream, and 60 for the third stream.

Model Building:

There are 3 types of Naive Bayes Classifiers –

Gaussian Naive Bayes: probabilistic machine learning algorithm( calculate the mean and standard deviation)

Bernoulli Naive Bayes: used for Boolean data they follow Bernoulli distribution. (success/failure)

Multinomial Naive Bayes used for document or text classification problems

See below Neural network example for increment learning

neuralnetwork.fit(X\_train1, y\_train1) //here neural network training with X\_train1

predictlabel = neuralnetwork.predict(X\_test1)

accuracy1 = show\_accuracy(predictlabel,y\_test1) \* 100

roc1 = roc\_auc\_score(y\_test1,predictlabel) //calculating ROC and Accuracy on stream1

nn\_acc.append(accuracy1)

nn\_roc.append((roc1\*100))

neuralnetwork.incremental\_input(X\_train1, X\_train2, y\_train2) //in this step incrementally

predictlabel = neuralnetwork.predict(X\_test2) //adding stream 2 as X\_train2

accuracy2 = show\_accuracy(predictlabel,y\_test2) \* 100

nn\_acc.append(accuracy2)

roc2 = roc\_auc\_score(y\_test2,predictlabel) //calculating ROC and Accuracy on stream2

nn\_roc.append((roc2\*100))

neuralnetwork.incremental\_input(np.row\_stack((X\_train1, X\_train2)), X\_train3, y\_train3)

predictlabel = neuralnetwork.predict(X\_test3)

accuracy3 = show\_accuracy(predictlabel, y\_test3) \* 100

nn\_acc.append(accuracy3)

roc3 = roc\_auc\_score(y\_test3, predictlabel) //same for 3rd stream also

nn\_roc.append((roc3\*100))

So, by using the above code simultaneously pre-processing and training will happen for each incoming stream. By running the application, we can see as the stream comes the accuracy of the classifier will also increase.

Project Workflow:

To run the project double, click on the ‘run.bat’ file to get the below screen

Click on the ‘Upload Sonar Dataset’ button and upload a dataset

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

In the above screen, we can see the dataset contains 207 records and each record contains sonar signal values, and each record contains 61 column values. Now click on the ‘Convert Dataset into Streams’ button to create streams from the dataset

Text

Description automatically generated with medium confidence

In the above screen, we can see the first stream taking 186 records to train the classifier and 21 records to test the classifier, and the second stream using 165 records for training and 42 records for testing and goes on. For each stream, we can see there are differences in the dataset. Now click on the ‘Run Increment SVM Algorithm’ button to train SVM incrementally

Each stream dataset is different.

Text

Description automatically generated

In the above screen we can see in the first stream SVM got 71% accuracy and in the second stream we got 81 and 3rd stream got 78% accuracy. The higher the accuracy the better the classifier model. Now click on the ‘Run Increment Naïve Bayesian’ button to run naïve Bayes with an increment

Graphical user interface, text

Description automatically generated

In the above screen, with naïve Bayes first stream got 57% accuracy and the second stream got 73% and 3rd stream got 74% accuracy. Now click on the ‘Run Neural Network Algorithm’ button to get its accuracy

Text

Description automatically generated

In the above screen, the neural network got 85% accuracy for the first stream and 92% for the second stream, and 95% for the third stream. Now click on the ‘Accuracy Graph’ button to compare all algorithm's accuracy for each stream in the graph

Chart, line chart

Description automatically generated

In the above graph, the y-axis represents accuracy for each algorithm for each stream and the blue line represents SVM and orange line represents Naïve Bayes, and the green line represents Neural networks. Here, we can conclude that the Neural network is giving the best result compared to the other two algorithms. Now click the ‘ROC Graph’ button to get the ROC graph for each stream and all algorithms

Chart, line chart

Description automatically generated

In the above graph Neural network is giving better ROC so its performance is better compared to the other two algorithms

Conclusion and Future Scope

Overall, our models are only of limited utility since none were capable of correctly predicting objects with both precision and recall greater than 50%. This seemingly low performance is likely due to the many causes of other delays being outside the scope of our data.

There are a couple of 6 machine learning algorithms such as IBK (also known as KNN or K-NEAREST NEIGHBORS), Naïve Bayes, Decision Tree, Local Weighted, Neural Network, etc. But we are implementing only 3 algorithms right now python does not support the increment process for other algorithms. We are implementing SVM, Naïve Bayesian, and Neural networks.

References

https://en.wikipedia.org/wiki/Sonar\_signal\_processing

<https://www.lowrance.com/sonar-basics/#:~:text=Sonar%20is%20a%20technique%20used%20to%20detect%20water,water%20column%20between%20the%20transducer%20and%20the%20bottom>.

https://www.lowrance.com/sonar-basics/