Reframing Bike Challenge Problem using Model Selection

Chowdhury Farhan Ahmed

ICube Laboratory, University of Strasbourg, France
cfahmed@unistra.fr

Abstract. Bike challenge is an important real-life transportation problem where we need to predict the future availability of bikes based on the currently available bikes. Here the given problem does not provide enough training data for a particular test station. Therefore, the challenge becomes to use the given models of similar stations to perform the prediction. In this paper, we propose an efficient model selection approach to solve this problem.

Keywords: Machine Learning, Reframing, Model Selection.

1 Introduction

It is natural in the real-life scenarios that we may have several data for some places and may have few data in other places. For example, there may be new bike station beside my house and an old within one kilometer. If I want to predict the available bikes of the new station for the next day, I might not have train/previous data. But, I can get an idea from the behavior of the old station nearby. The main objective of reframing is to reuse the existing models in some other places where we do not have enough data/knowledge to build a new model [1] .

Recently problem based on bike sharing has become an interesting research issue in machine learning [2]. In this discovery challenge, a problem for predicting future available bikes for some test stations has been given where a particular test station does not have enough labelled data to build a model. Existing models of several training stations have been given. For a test station, the main task to find a similar training station of that particular test station to perform prediction. Therefore, the problem is aimed at reusing of learnt knowledge which carries critical importance in the majority of knowledge-intensive application areas.

In order to solve this problem, we have selected the most similar training station of a particular test station according to performance. That means the training station whose prediction error is minimum for the given small amount of labelled data of that particular test station.

2 Our Proposed Approach

Different models of 200 training stations (numbered as 1 to 200) are given where these models have been trained with data for a long period of time (more than

two years). Unlabelled test data for 75 stations (different from the 200 training stations) are given for 3 months (Nov. 2014 to Jan 2015). These test stations are numbered as 201 to 275. However, for learning the similarities between stations, data of one month (October-2014) are given for all the 275 stations. We have tested the October-2014 data of the 75 test stations with the given models of 200 training stations in order to know which training station performs most similarly to a particular test station. We have used the short-full model of a training station and mean absolute error (MAE) as the performance metric. For example, if station 50 has the lowest MAE value for station 201, then we marked station 50 as the nearest neighbour (based on performance) of station 201. For testing the unlabelled data of a particular test station, we have used the model of its nearest neighbour train station to perform the prediction.

Therefore, the steps of our algorithm are as follows.

1. Prepare the input and output data in the proper format. Here, we have used ARFF format for using Java and Weka [3].
2. Select the best training station among the 200 models (short_full_temp) according to their MAE value for a particular test station on the given October-2014 data.
3. For the unlabelled test data (Nov. 2014 to Jan 2015), use the best training model for a particular test station to perform the prediction.

We have also tried with the short_full model in Step 2 described above. As the short_full_temp model performed better than the short_full model with the small given deployment data, we have decided to use the short_full_temp model. However, the following program files have been used.

PrepareM_CSV_ECML_Contest.java: It prepares the CSV deployment files (October-2014) of all test stations (201-275) for the conversion of ARFF format by eliminating the "NA" values by "?".

Prepare_CSV_Tst_ECML.java: It prepares the CSV test file for the conversion of ARFF format by eliminating the "NA" values by "?", adding a last column for class label named bikes and initialize it to zero.

WriterARFF_FromCSV_ECML: Converts all the preprocessed CSV deployment and test files into ARFF format.

BestTrainModel_ECML: Finds the best training model (according to MAE) for a particular test station.

Contest_ECML_Bike: Performs the predictions for the unlabelled test data by using the selected best training model for that particular test station. And write the results to the output file according to specific format.

3 Conclusions

In order to solve the bike challenge problem, we have proposed a simple but effective method. It selects the best similar model according to performance. Experimental results on the small test data show that it can achieve a good performance. Hence, we are expecting good results for the full test dataset using this method.

Acknowledgements

This work was supported by the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA).

References

1. Ahmed, C.F., Lachiche, N., Charnay, C., Braud, A.: Reframing continuous input attributes. In: Proceedings of the 26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2014), Limassol, Cyprus. pp. 31–38 (2014)
2. Fanaee-T, H., Gama, J.: Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence 2(2-3), 113–127 (2014)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)