Impact of Noise in Boosting Methods

Qi Feng, Xiaonan Zhao

***Abstract*—Organizing taxis in a metropolis has significant social and economic benefits, for instance in reducing the waiting time of passengers and improving the efficiency of taxis. Even though rearranging taxi resources with respect to their own characteristics can be illuminating, taxi arrangement is not well studied.**

In this paper, prediction problem of urban structure is defined and an item-level multi-factor prediction method is proposed. Taxi's influence on a happening of travel is revealed in a more accurate way. In this item-level approach, both the taxi characteristic and location influences are considered, compared to previous works. Taxis are clustered using 5 major attributes, while locations are divided into areas adjacent to each other. Additionally, time variation is also considered in the proposed model. The problem is then formulated as compute the probability of a specific taxi gets a travel (customer) at a specific location in certain time period of a day. The underling mathematical model is a tensor optimization model, which is later processed. The advantage of the proposed method (PHF-MF) was evaluated by computing the Root Mean Square Error (RMSE) with a comparison for two other methods, using past taxi travel records gathered in Beijing.

I. Introduction

The fundamental idea of ensemble methods is to construct a combination of weak base classifiers that are diverse and result in a high accuracy. Multiple ensemble methods, including boosting[], bagging[], and decision tree ensemble[], are being introduced in the past 20 years. Boosting algorithms took a significant place in ensemble methods. Adaboost[**?**] and the recently introduced Deepboost[**?**] are typical boosting algorithms with a good experimental result without overfitting the training set. They both have good theoretical learning bound and benefit directly from minimizing the learning bound. However, the experimental robustness of these algorithms have not been tested before.

II. Realistic Noise

The impact of noise in the performance of an algorithm is regarded as the robustness. In Diettterich's work[**?**], multiple levels of noise are added to sample dataset to test the robustness of Adaboost. However, the noise introduced was by reverting labels in training data randomly without replacement with a fraction r. This makes the noise in the training dataset unrealistic. Many other publications have used different procedure to add noise into the training data The procedure is to set each training sample's label to a random class with probability r. Both of these procedures are adding noise with a uniform distribution over the entire training data.

However, in actual datasets noise are not distributed with a uniform fashion across the entire dataset. In Xingquan et al.'s work[**?**], a general method to eliminate noise from training data is introduced. In this algorithm, noise identification is based on the majority and non-objection schemes, which is founded on the assumptions that noise is distributed according to the distribution of empirical errors. The denser the classification errors, the denser the noise in training set. More generally, realistic noise distribution should not be uniform over the entire training dataset, but with a relation to the classification error.

Following the definition of Adaboost, the distribution $D_t(i)$ updated during each iteration of Adaboost is a perfect simulation of the training error distribution after round t. Since $D_t(i)$ is updated with according to the loss function, the distribution would be updated with a higher level where empirical errors are denser. Therefore, the distribution from Adaboost after T rounds(finished) would be suitable for introducing realistic noise.

III. Methods

IV. Experiment and Results

V. Conclusion

VI. Acknowledgments

References

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.

[2] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *ICML*, vol. 3, 2003, pp. 920–927.

[3] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[4] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[5] C. Cortes, M. Mohri, and U. Syed, "Deep boosting," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1179–1187.