Ensemble Learning for Crowd Flows Prediction on Campus

Chuting Wu(✉), Tianshu Yin, Shuaijun Ge, and Ke Yu

School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications, Beijing 100876, China
{wuchuting,sjge,yuke}@bupt.edu.cn, yintianshu1994@163.com

Abstract. Campus security is an increasing-attention problem in recent years. Crowd flows prediction on campus is helpful for people monitoring and can avoid potential risks. In this paper, based on distributed visiting data collection on campus, we propose a crowd flows prediction method with ensemble learning. For feature selection, we introduce more information than people visiting data, such as vocation and weather, and evaluate the feature importance as well as their combinations. For prediction model, we use stacking method with Random Forest, Gradient Boosting Tree and XGBoost for a better performance of prediction. Experimental results show that our method obtain high accuracy for crowd flows prediction with low extra cost.

Keywords: Crowd flows prediction · Ensemble learning · Stacking

1 Introduction

Nowadays more and more universities open their doors wider in order to invite more students to receive higher education. Not only the students from other universities can attend lectures and visit labs, but also teenagers can take part in academic activities on campus. It is a good policy for the young to broaden their insights. However, the administrators of universities and governments have to face more severe security risks. They tend to implement intelligent and networked monitoring system to manage and control the people, resources and environments on campus. The called Intelligent Campus comes into being [1]. The Intelligent Campus has been implemented in many universities. The data collected from multiple sources such as faculty/student e-card and campus WI-FI reflects the people's lifestyle. Big data and machine learning techniques make it possible to analyze and predict people's behavior on campus.

Crowd flows monitoring and prediction is an important part of the intelligent campus monitoring system. The visiting data of people in different places on campus is collected by mobile phones or sensors and analyzed in real-time. By analyzing the historical visiting data collected by the monitoring system of intelligent campus, we can learn the people's daily activity pattern and understand their behavior, which is helpful to guide them to move more smoothly,

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 103–113, 2018.
https://doi.org/10.1007/978-3-319-73830-7_11

safely and comfortably from one place to another. It is also helpful to prevent stampede accident or other disasters in campus.

In this paper, we attempt a new method based on the big data collected by mobile phones to analysis the change of the number of people on campus for preventing high-risk accidents. We propose a crowd flow prediction method with ensemble learning based on the historical visiting data of people. Ensemble learning [9,10], mainly including bagging, boosting and stacking schemes, is a powerful method by using multiple learning algorithms to obtain better predictive performance than constituent learning algorithms alone. There are mainly two challenges for the prediction task. The one is how to select the features about the people activities on campus. The other is how to combine multiple learning algorithms to obtain the optimal result.

Our work makes the following contributions:

(1) We introduce the crowd flows monitoring framework and distributed visiting data collection method;
(2) Based on collected visiting data, we analyze the statistical characteristics of crowd distribution;
(3) We propose crowd flows prediction method by using ensemble learning;
(4) We verify the performance of the proposed crowd flows prediction method by experiments and result analysis.

The paper is organized as follows: Sect. 2 introduces the related work. Section 3 describes the proposed crowd flows prediction method in detail. Section 4 presents the experiments and results. Section 5 concludes the paper.

2 Related Work

Nowadays with the rapid development of Smart City, Internet of Things (IoT) and big data technologies, crowd flows analysis gains more and more attentions. There are many application examples in cities and campus. For the cities, [2] analyzes the characteristics of mixed traffic flow with non-motorized vehicles and motorized vehicles at an unsignalized intersection. [3] forecasts citywide crowd flows based on big data by decomposing flows into seasonal, trend and residual flows components, which uses different mathematical models respectively. [4] proposes deep spatial-temporal residual networks for citywide crowd flows prediction. For the campus, [5] uses the records of people's consumption and WI-FI data to mine the spatial and temporal information of human activity and mobility and predicts the distribution of people on campus. [6] explores the relationship between the achievement of a student and his study partners based on the information collected from the digital campus card. Our paper emphasizes crowd flows on campus based on the visiting data other than citywide crowd flows. Since the people distribution on campus is much more related to time and space, we design a detailed and combined feature selection method for prediction. The high-dimensional class-imbalanced data may generate the cross effect and concentrate on the majority class which lead to bad results. Focusing on

these issues, [7] raises the methods of preprocessing data and [8] comes up with Ensemble Feature Selections before machine learning.

Ensemble learning [9,10] is a hot topic in the area of machine learning. It is a way by strategically generating and combining multiple models to solve a particular learning problem. There are several types of ensemble learning. Bagging (represented by Random Forest(RF) [11]), and boosting(represented by Gradient Boosting Tree(GBT) [12] and XGBoost [13,14]), are the common ensemble methods which have improved the accuracy of prediction a lot. Stacking [15] is also a common method to assemble machine learning models to enhance the ability of prediction. There are many applications of stacking. For example, [16] uses stacking method to classify imbalanced malware without unpacking process. And [17] uses stacking method for sentiment classification and verify that stacking is consistently effective over all domains, which works better than majority voting. Our paper considers to combine the traditional prediction methods, bagging method (RF) and boosting method (GBT, XGBoost), in order to explore a better stacking method for crowd flows prediction task.

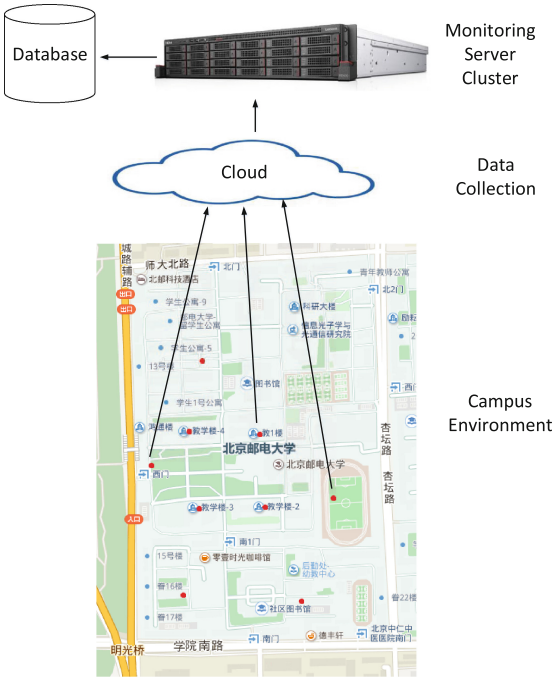

Fig. 1. Campus monitoring and prediction system

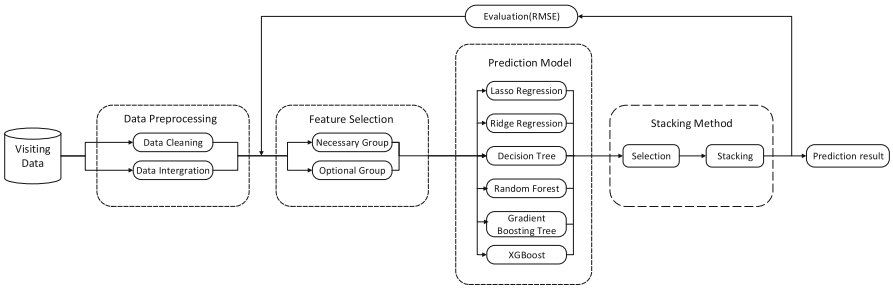

Fig. 2. Data analysis process

3 Crowd Flows Prediction Method

3.1 System Framework

Campus Monitoring and Prediction System aims at monitoring people activities and predicting the number of people at specific times and places on campus. The system framework is shown in Fig. 1. Data collection is based on distributed mobile devices and cloud. When a person arrives at one place such as classroom, library, cafeteria, and sport field, he/she will submit a request to server on Campus Network Center by mobile phone, Pad or laptop to apply Internet connection. The server's log is processed and then stored in the crowd database. Data analysis and visualization is also implemented on the server of Campus Network Center.

3.2 Algorithm Description

In this section, we present our main algorithm and Fig. 2 shows the whole process of data analysis. The description of the algorithm is given as follows:

(1) Do data cleaning and statistical analysis based on the raw visiting data in order to delete the noise data.
(2) Extract features from data and add some related features such as weather and holiday information.
(3) Divide all of the features into two groups, called necessary group and optional group respectively.
(4) Carry on crowd flows prediction by trying different features and models of machine learning.
(5) Stacking different combinations of them to obtain better results.

3.3 Data Collection and Preprocessing

3.3.1 Raw Data Collection

The raw data is collected through mobile devices of people which try to connect to the WI-FI on campus. The server of Campus Network Center receives the

request and records the detailed information about the visiting person, including time, device number, location and so on. The raw data mainly includes 3 fields as follows:

(1) person-id: each person's identifier, represented by a positive integer starting at 1.
(2) time-stamp: person's visiting time, represented by a number sequence with month, day, and hour.
(3) loc-id: id of the locations on campus, represented by a positive integer between 1 and 36.

3.3.2 Data Preprocessing

Data preprocessing includes data cleaning and data integration. The raw visiting data is transferred to structured data by deleting the incomplete and noise data. In the raw data, there are some special days in which the number of people increases or decreases sharply, for instance, the Anniversary of University and National Holiday, those points which deviate from the normal data points can't show the common rules of crowd flows and influence the accuracy of the prediction result, so we delete these noise data. It should be mentioned that although the number of people in August decreases sharply because of summer vacation, we still reserve these data points for a better result. Figure 3 shows the number of visiting people by month from July to October. It can be seen that the number of people changes with semester, vacation and special days. The visits in August are the least due to summer vacation, while the visits in October are the most due to university anniversary. These statistical analysis results should be considered in feature selection process.

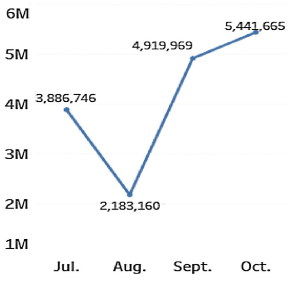

Fig. 3. Number of visiting people by month

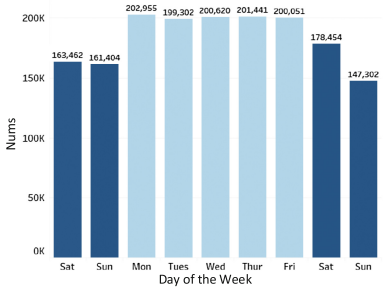

Fig. 4. Crowd distribution on weekdays and weekends

3.4 Feature Selection

3.4.1 Feature Description

The campus crowd flows prediction task is to predict the number of people in different places per hour. All of the features related to crowd should be extracted. Firstly, because we focus on the number of people on campus rather than the individual behavior, the number of people in each place per hour takes the place of person-id. Secondly, we split the time-stamp into 3 features (month, date, hour), for the reason that this way will provide more information. Thirdly, we add some new features including weather and holiday information. The crowd flows are highly related to weekdays and weekends, as shown in Fig. 4, so we add the information about Monday to Sunday. We also introduce the weather information from 2345 weather report [18] into our data, including temperature, air quality, wind level and so on. According to investigation in the campus and individual experience, we finally select 13 features and the detailed description is as Table 1.

Table 1. Feature description

Feature	Meaning	Description
month	The month of the data	Range 7 to 11
date	The date of the data	Range 1 to 31
hour	The hour of the data	Range 0 to 23
day_of_the_week	The day of the week of the date	Range 1 to 7
loc_id	The id of location	Range 1 to 36
weekend	The date is weekend or not	Weekend set 1, else 0
holiday	The date is holiday or not	Holiday set 1, else 0
air_quality	The air quality of the date	Range 1 to 6. Larger number, Worse air quality
highest_temp	The highest temperature of the date	Positive integer
lowest_temp	The lowest temperature of the date	Positive integer
wind	The wind level of the date	Range 0 to 3
PM2.5	The amount of the PM2.5 in the air of the date	Positive integer. Larger number, Higher PM2.5
rain_snow	The rain or snow level of the date	Positive integer. Larger number, Heavier rain/snow

3.4.2 Feature Ranking and Combination

Intuitively, more features can provide more information, which is helpful to increase the performance of the predict model. But some features which may interfere each other and the cross effect is likely to lead to a bad result.

And some features may not have much effect to improve the model, even damage it. For example, the date of the data is cycled and it is easy to cause misleading. So we use XGBoost to score each feature firstly, which provides the importance indication of each feature. After considering the scores, we divide all features into 2 groups, i.e. the necessary group and the optional group. Each learning iteration we select all features in necessary group and some of the features in optional group and combine them into a feature set.

3.5 Prediction Method

Model selection is another important factor to improve the performance of the prediction process. Regression analysis is a statistical process for estimating the relationships among variables. Before designing our prediction model, we test some traditional prediction models and compare their performances. The traditional prediction models include multiple linear regression, Decision Tree, and so on. Ensemble learning methods utilize multiple learning algorithms to obtain better predictive performance than any of the constituent learning algorithms alone. The ensemble learning mainly includes bagging, boosting and stacking. Random Forest is a bagging-based model by taking the majority vote in the case of single Decision Trees. XGBoost, as an upgrade version of Gradient Boosting Tree, is a very efficient scalable end-to-end tree boosting system which has been widely adopted for data analysis. Stacking is also a typical ensemble method that combines multiple machine learning models to construct a stronger one. Compared with bagging and boosting, stacking is more flexible because it combines different types of models or joints the predictive result to the features. Our proposed stacking method for crowd flows prediction is to utilize multiple predictive models' results as new features and put them into another model. Since the process of stacking needs more different kinds of models to provide diversity, we choose Lasso, Ridge, Decision Tree, Random Forest, Gradient Boosting Tree and XGBoost as optional models.

4 Experiments and Results

4.1 Experimental Dataset

The experimental dataset is provided by Campus Network Center, which contains the three fields as Sect. 3.2 from July to October, 2016. The experimental dataset has 22,600,642 items as total. Among these data, there are 16,431,540 items from July to October as train set and the others are as test set. The task of the experiment is to predict the number of people at 36 different locations per hour in November according the historical dataset. We use Root Mean Square Error(RMSE) between true and predicted values to evaluate our proposed method. The less RMSE, the better performance for the predictor.

$$X_{RMSE} = \sqrt[2]{\frac{\sum_{i=1}^{n}(x_{pred,i} - x_{true,i})^2}{n}} \tag{1}$$

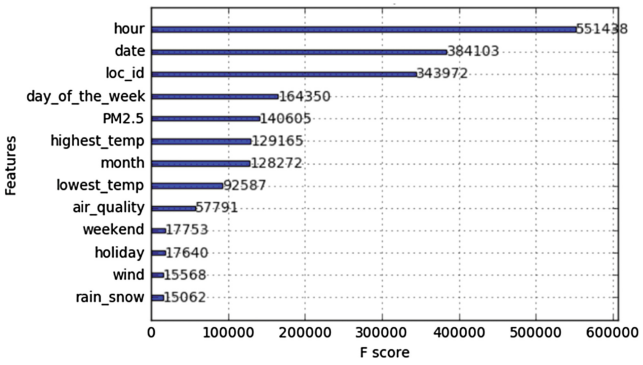

Fig. 5. Feature importance

4.2 Result Analysis

Firstly, the importance of all features calculated by XGBoost is shown in Fig. 5. We divide the features into necessary group and optional group according those computing and observation results. The necessary group including month, hour, loc_id, day_of_the_week, holiday, PM2.5, highest_temp and lowest_temp, and the optional group contains date, wind, air and weekend. It should be mentioned that from the statistic observation of raw data, we find that the crowd number change presents weekly period. However, if we reserve the two features of date and day_of_the_week, their periodic relationship leads to a negative influence to the prediction result. So we put the feature of date into optional group and calculate other features' importance again. And the feature of day_of_the_week performs well compared with the feature of weekend because the former provides more details than the latter. So different patterns of features cause different results and an appropriate pattern is important for the experiment.

Table 2 shows the RMSE of different prediction models under different feature inputs. For each model, we use 5-folds as cross-validation and Table 2 shows the best score of it. "All" means all of the necessary and optional features are used. "All-date" means all of the features are used other than date feature. From Table 2, lasso and ridge regression present worse performance and change little when using different features. Random Forest and Gradient Boosting Tree produce better RMSE. XGBoost leads to the best result and less time consuming.

According to the results, we can find that XGBoost presents it time-saving and accurate performance among the single prediction models. However, stacking is a more powerful way to take advantage of every single model and improve the prediction performance.

For stacking, we choose the best result of each prediction model to ensemble and use XGBoost as the prediction model at the second level. For each model we use 5-folds as cross-validation. The result is shown as Table 3. For each experiment, "$\surd$"means the model is used in stacking, "-" means unused. We can see from Table 3 that each stacking experiment obtains lower RMSE compared to single model in Table 2.

Table 2. Feature selection

Features	Models					
	Lasso	Ridge	DT	RF	GBT	XGBoost
All	272.66	276.80	99.88	82.36	84.00	81.14
All - date	272.66	276.65	93.94	75.61	75.42	74.31
All - date - air_quality	272.66	276.60	93.67	75.16	75.41	72.78
All - date - wind	272.66	276.43	93.94	75.15	76.52	74.24
All - date - rain_snow	272.66	276.48	91.98	75.95	76.08	75.24
All - date - wind - rain_snow	272.66	274.68	91.11	76.41	75.39	74.33

Note: Decision Tree (DT), Random Foreset (RF), Gradient Boosting Tree(GBT)

Table 3. Stacking result

Exp	Lasso	Ridge	DT	RF	GBT	XGBoost	Stacking
1	-	√	√	√	√	√	71.73
2	√	-	√	√	√	√	71.73
3	√	√	-	√	√	√	71.72
4	√	√	√	-	√	√	72.47
5	√	√	√	√	-	√	72.06
6	√	√	√	√	√	-	74.18
7	√	√	√	√	√	√	71.61

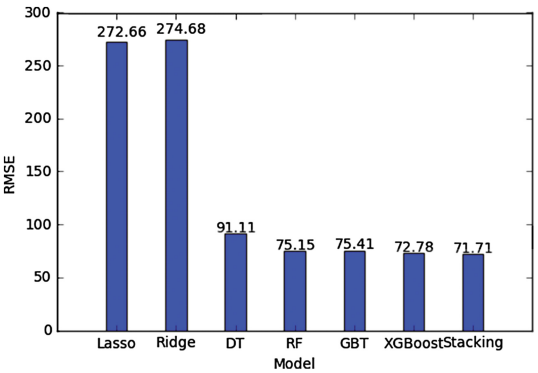

Fig. 6. Performance of different prediction models

Figure 6 shows the best results of different prediction methods, from which we can see that Stacking surpassed the others. The result verifies that the stacking method takes advantages of every prediction models and present a better performance.

5 Conclusion

In this paper, we introduce Campus Monitoring and Prediction System based on distributed people visiting data collection. We propose the feature selection and stacking-based ensemble learning for crowd flows prediction. The experimental results show that our proposed stacking method performs better for a lower RMSE. The next step is to improve the crowd flows prediction method and to implement the application system.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 61601046 and No. 61171098, and is partially supported by the 111 Project of China under Grant No. B08004, and EU FP7 IRSES Mobile Cloud Project under Grant No. 612212.

References

1. Jackson, M.: Intelligent campus. In: International Symposium on Pervasive Computing and Applications, p. 3. IEEE (2006)
2. Xie, D.F., Gao, Z.Y., Zhao, X.M., et al.: Characteristics of mixed traffic flow with non-motorized vehicles and motorized vehicles at an unsignalized intersection. Phys. A Stat. Mech. Appl. **388**(10), 2041–2050 (2009)
3. Hoang, M.X., Zheng, Y., Singh, A.K.: FCCF: forecasting citywide crowd flows based on big data. In: The ACM Sigspatial International Conference, pp. 1–10. ACM (2016)
4. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction (2016)
5. Zhu, M., Pan, C., Yang, G., et al.: Prediction of population distribution on campus based on historical location data. In: Control and Decision Conference, pp. 2849–2854. IEEE (2016)
6. Fan, S., Li, P., Liu, T., et al.: Population behavior analysis of Chinese university students via digital campus cards. In: IEEE International Conference on Data Mining Workshop, pp. 72–77. IEEE (2016)
7. Yin, H., Gai, K.: An empirical study on preprocessing high-dimensional class-imbalanced data for classification. In: IEEE International Conference on High PERFORMANCE Computing and Communications. IEEE (2015)
8. Yin, H., Gai, K., Wang, Z.: A classification algorithm based on ensemble feature selections for imbalanced-class dataset. In: IEEE International Conference on Big Data Security on Cloud, pp. 245–249. IEEE (2016)
9. Chandra, A., Yao, X.: Ensemble learning using multi-objective evolutionary algorithms. J. Math. Model. Algorithms **5**(4), 417–445 (2006)
10. Webb, G.I., Zheng, Z.: Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques. IEEE Educational Activities Department (2004)
11. Breiman, L.: Random forest. Mach. Learn. **45**, 5–32 (2001)
12. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
13. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. pp. 785–794 (2016)

14. Chen, T., He, T., Benesty, M., et al.: XGBoost: extreme gradient boosting (2017)
15. Kaggle Ensemble Guide: http://mlwave.com/kaggle-ensembling-guide
16. Zhang, Y., Huang, Q., Ma, X., et al.: Using multi-features and ensemble learning method for imbalanced malware classification. In: Trustcom/BigDataSE/ISPA, pp. 965–973. IEEE (2017)
17. Su, Y., Zhang, Y., Ji, D., Wang, Y., Wu, H.: Ensemble learning for sentiment classification. In: Ji, D., Xiao, G. (eds.) CLSW 2012. LNCS (LNAI), vol. 7717, pp. 84–93. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36337-5_10
18. 2345 Weather Report: http://tianqi.2345.com