Traffic Prediction in a Bike-Sharing System

Yexin Li [1, 2,*] , Yu Zheng[2], Huichu Zhang [2, 3], Lei Chen[1]
[1]The Hong Kong University of Science and Technology, Hong Kong, China
[2]Microsoft Research, Beijing, China
[3]Shanghai Jiao Tong University, Shanghai, China
{v-yexli, yuzheng}@microsoft.com, devillaw_zhc@163.com, leichen@cse.ust.hk

ABSTRACT
Bike-sharing systems are widely deployed in many major cities, providing a convenient transportation mode for citizens' commutes. As the rents/returns of bikes at different stations in different periods are unbalanced, the bikes in a system need to be rebalanced frequently. Real-time monitoring cannot tackle this problem well as it takes too much time to reallocate the bikes after an imbalance has occurred. In this paper, we propose a hierarchical prediction model to predict the number of bikes that will be rent from/returned to each station cluster in a future period so that reallocation can be executed in advance. We first propose a bipartite clustering algorithm to cluster bike stations into groups, formulating a two-level hierarchy of stations. The total number of bikes that will be rent in a city is predicted by a Gradient Boosting Regression Tree (GBRT). Then a multi-similarity-based inference model is proposed to predict the rent proportion across clusters and the inter-cluster transition, based on which the number of bikes rent from/ returned to each cluster can be easily inferred. We evaluate our model on two bike-sharing systems in New York City (NYC) and Washington D.C. (D.C.) respectively, confirming our model's advantage beyond baseline approaches (a 0.03 reduction rate on error), especially for anomalous periods (a 0.18/0.23 reduction rate on error).

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications - *data mining, spatial database and GIS*;

Keywords
Bike sharing systems, meteorology, traffic prediction.

1. INTRODUCTION
Bike-sharing systems are widely deployed in many major cities, like New York, Paris and Beijing, providing a convenient transportation mode for people's commutes. A user can rent (i.e. check out) a bike at a station near their origin and return (i.e. check in) it to a station close to their destination. Users are required to swipe an RFID card when checking out/in a bike. A record, consisting of the bike ID, timestamp and station ID, is generated for each card swipe.

Bike-sharing systems face challenges in bike rebalance between stations. Intrinsically, bike usage are skewed, changing over time and locations. Consequently, some stations may be jammed without enough docks for future returned bikes while some lack available bikes for interested users. Monitoring the current number of bikes at each station cannot tackle the challenge thoroughly, as it is too late to reallocate bikes after an imbalance has occurred.

SIGSPATIAL'15, November 03-06, 2015, Bellevue, WA, USA
© 2015 ACM. ISBN 978-1-4503-3967-4/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2820783.2820837

* *Yu Zheng is the correspondence author of this paper.*

To address this issue, we predict the number of bikes that will be checked out from (i.e. check-out) and checked in to (i.e. check-in) each station in a bike-sharing system during a future period, based on historical check-out/in data as well as meteorology data. The predictions can help to operate a bike-sharing system more efficiently, improving resource utilization. To achieve this goal is very challenging as *bike traffic is impacted by multiple complex factors*, such as the time of day, day of the week, meteorology, events, and the correlation between stations.

1) Meteorology: More people may check out/in a bike on a sunny day than on a rainy day. This phenomenon also exists between a cool day and a warm day; same with days of different levels of wind. However, some categories of weather, e.g. rainy, occur rarely, while other categories like sunny may be very common. In addition, some temperature & wind speed scenarios have never happened historically, but may happen in the future, e.g.(11.7 ℃, 4.6 mph) for NYC. These lead to an unbalanced distribution of data if partitioned by meteorology. Traditional machine learning models are trained to fit the majority of observations, thus would scarify the models' accuracy under minor conditions. However, being able to predict traffic under rare conditions, e.g. rainy hours, is as important as that under ordinary ones, e.g. sunny hours.

2) *The correlation between stations*: First, the bike traffic of nearby stations affect each other. For instance, when a station is full of bikes, users have to check in their bikes at nearby stations. Likewise, when a station is running out of bikes, people turn to nearby stations for checking out. Second, when the check-out in a (origin) region increases tremendously for some reason, e.g. an unusual event, check-in at other (destination) regions will be affected significantly beyond common patterns. As a result, the bike traffic at an individual station may change irregularly and the bike transition between stations may vary tremendously.

To tackle these challenges, we propose a hierarchical prediction model, which is comprised of five major components: 1) a Bipartite Station Clustering algorithm to cluster individual stations according to their geographical locations and historical transition patterns (explained in 3.1.2); 2) an Entire Traffic Prediction model to predict the total check-out of the whole city based on time and meteorology features; 3) a Check-out Proportion Prediction model to predict the check-out proportion (among total check-out) across station clusters; 4) an Inter-Cluster Transition Prediction model to predict the dynamic bike transition probability between station clusters; 5) a Check-Out/In Inference algorithm to calculate the check-out/in of each station cluster based on the outputs of the former components. Our contributions are three-fold:

- The Bipartite Station Clustering algorithm formulates a two-level hierarchy of stations where the root consists of all the stations in a city, improving the prediction accuracy for three reasons: First, the total check-out in a city (i.e. the higher level node) is much more robust, regular and easier to predict, providing a bound to the total check-out across clusters in the lower level; Second, the total traffic in a cluster is more regular and easier to predict than that at an individual station; Third, the inter-cluster transition is more concentrated, thus more robust, than that between stations. Our clustering method considers stations' geographical locations and transition patterns in an iterative approach, resulting in much better clusters than other clustering methods.

- We propose a multi-similarity-based inference model to predict the check-out proportion across clusters and the inter-cluster transition. The model integrates multiple similarities between the features of the time period to be predicted and those of historical periods. Different features have their own similarity functions, which are learned collectively from historical data and aggregated by a similarity production. The model handles the unbalanced distribution in observations.
- We evaluate our model with real data from NYC and D.C. Our model outperforms the baselines (a 0.03 reduction rate on error), especially under anomalous situations (a 0.18/0.23 reduction rate). The datasets have been released in [26].

2. Overview

This section defines the notations (in Table 1) and terminologies used in this paper. Then we provide an overview of the framework.

Table 1. Notations

Notation	Description
S_i	The i^{th} station
n	Number of stations
C_i	The i^{th} cluster
m	Number of clusters
$O_{C_i,t}/O_{S_i,t}$	Check-out of cluster C_i/ station S_i in time t
$I_{C_i,t}/I_{S_i,t}$	Check-in of cluster C_i/ station S_i in time t
E_t	Entire traffic in time t
$T_{t,m\times m}$	Inter-cluster transition matrix in time t
P_t	Check-out proportion vector in time t
f_t	Feature vector in time t
M_t	Meteorology vector in time t
$D_{m\times m}$	Trip duration matrix

2.1 Preliminary & Problem Definition

Definition 1: *Trip.* A trip $Tr = (S_o, S_d, \tau_o, \tau_d)$ is a bike usage record, where S_o denotes the origin station, consist of latitude $S_o.lat$ and longitude $S_o.lon$; S_d denotes the destination station, consist of latitude $S_d.lat$ and longitude $S_d.lon$; τ_o and τ_d are the time when the bike is checked out at S_o and checked in at S_d respectively.

Definition 2: *Meteorology.* The meteorology $M_t = (w_t, p_t, v_t)$ is a vector corresponding to period t, where w_t, p_t and v_t stand for the weather, temperature and wind speed in t respectively.

Definition 3: *Check-out/in.* Check-out $O_{C_{i,t}}$/ check-in $I_{C_{i,t}}$ is the number of bikes checked out/in in cluster C_i during time period t.

Definition 4: *Entire traffic.* Entire traffic E_t is the total number of bikes that are checked out in the whole city during time period t.

Problem Definition: *Check-out/in prediction problem.* Given a set of historical trips $T_H = \{Tr_1, Tr_2, \ldots, Tr_H\}$, we want to predict the check-out/in of each station $S_i, i = 1,2, \ldots, n$ (cluster $C_i, i = 1,2, \ldots, m$) during a future period, which is set as 1 hour in our work.

2.2 Framework

Fig.1 shows the framework of our model, which consists of two processes: offline process and online process.

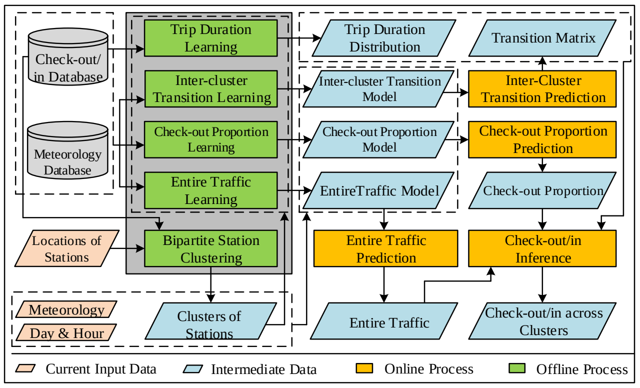


Fig. 1. Framework of our model

Offline Process: As shown with green rectangles in Fig.1, the offline process consists of the following five parts:

1*) Bipartite station clustering*: To address the irregular fluctuation issue at each individual station, we propose a bipartite clustering algorithm to cluster stations into groups. In bipartite station clustering, stations are clustered based on their geographical locations and historical transition patterns. As a result, stations in one cluster should not only be closed to each other geographically, but also have similar transition patterns to all clusters. The result of clustering is the foundation of the following steps as all models are learned based on clusters except for the entire traffic model.

2) *Entire traffic learning*: We leverage GBRT to learn an entire traffic model, which is the 'root' of our hierarchical prediction. The features considered in GBRT which affect the entire traffic significantly are extracted according to historical check-out/in data and meteorology data.

3) *Check-out proportion learning*: To allocate the entire traffic to each cluster, we need to predict the check-out proportion across clusters. Check-out proportion is predicted by a multi-similarity-based inference model. The H most recent proportions, $P_1, P_2, \ldots$, P_H, to the future period t whose proportion P_t we want to predict, are selected out. For $P_1, P_2, \ldots, P_H$ and P_t, their corresponding time and meteorology features $f_1, f_2, \ldots, f_H$ and f_t are extracted to calculate the similarities: $W(f_1, f_t), W(f_2, f_t), \ldots,$ $W(f_H, f_t)$; here, W is a similarity function. These similarities are the weights we used to calculate the weighted average of $P_1, P_2, \ldots,$ P_H, which is considered as the predicted check-out proportion in period t. The similarity function W is learnt in this step.

4) *Inter-cluster transition learning*: After a bike is checked out, predicting its transition is necessary for check-in prediction. An inter-cluster transition matrix (defined in 3.4) is used to describe where a bike will be checked in given where and when it is checked out. It is predicted using the same model with the previous step, a multi-similarity-based inference model.

5) *Trip duration learning*: The trip duration is another important element in transition. The trip duration between each pair of clusters is described by a lognormal distribution, whose parameters are calculated by maximum likelihood estimation. These two transition elements: inter-cluster transition and trip duration will be used in the check-in inference step later to help us estimate the probability that a bike will be checked in to a special cluster during a special time period, given when and where it is checked out.

Online Process: As shown with yellow rectangles in Fig. 1, the online process consists of four steps: entire traffic prediction, check-out proportion prediction, inter-cluster transition prediction and check-out/in inference. We extract the time and meteorology features of the future period t, which we want to predict, and leverage the entire traffic prediction model, check-out proportion prediction model and inter-cluster transition prediction model learned previously to predict: 1) the entire traffic; 2) the check-out proportion across clusters; 3) the inter-cluster transition matrix. Then each cluster's check-out/in can be easily inferred, incorporating the trip duration distributions learned in offline process.

3. Offline Process

3.1 Bipartite Station Clustering

3.1.1 Insights

We group individual stations into clusters according to their geographical locations and transition patterns. Fig. 2 A) presents an example of the station clustering result in NYC, where points with the same color denote stations pertaining to a same cluster. The reasons that we group stations into clusters are two-fold:

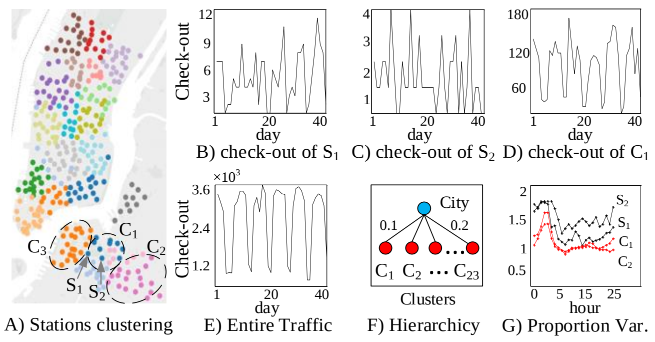


Fig. 2. Bike data from 1st Jun. to 10th Jul., 2014, in NYC

1) Affected by multiple complex factors, such as meteorology and correlation between stations, a single station's traffic seems too chaotic to predict. Fig. 2 B) and C) respectively show check-out from 9:00am to 10:00am of two nearby stations, S_1 and S_2 from a same cluster, which change from day to day very differently. In addition, it seems impossible to find any periodicity and regularity from an individual station's check-out. Particularly, at stations whose check-out is sparse, e.g. S_2, the fluctuations over its mean observation seem random. Clustering stations has three benefits:

First, after grouping some individual stations into a cluster C_1, as presented in Fig. 2 D) of its check-out during 9:00am-10:00am, the periodicity and regularity become much obvious than those of a single station, thus easier to predict. *Second*, the clusters formulate a two-level hierarchy of stations, shown in Fig. 2 F), based on which we can conduce hierarchical prediction. Fig. 2 E) shows the entire traffic from 9:00am to 10:00am. We can see that its periodicity and regularity are even more obvious and robust, which makes the entire traffic prediction easier and more accurate. Thus the hierarchical prediction can improve the accuracy as an accurate prediction in the higher level (entire traffic) can bind the prediction error in the lower level (check-out across clusters). *Third*, the check -out proportion across clusters and the inter-cluster transition are more concentrated, thus robust, than those of stations. This is confirmed by Fig. 2 G), which shows the average deviation of check-out proportion of cluster C_1, cluster C_2, station S_1 and station S_2, all in each hour of the day. The figure about transition is similar and we do not show here for space reason.

2) In reality, predicting the check-out/in of each individual station is not necessary. Understanding each cluster's check-out/in is enough for bike reallocation because users usually check out/in bikes at a random station closed to their origins/destinations. If a station is without available bikes/docks, it is convenient for a user to check out/in a bike at another station nearby. In addition, if some events happen, which may affect bike usage, they usually influence an area instead of only an individual station.

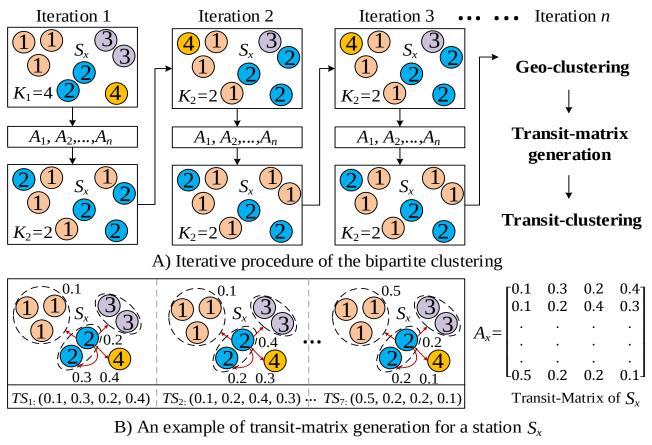
We consider the geographical location and transition pattern of a station simultaneously in an iterative approach for two reasons:

1) Considering that our aim is to offer more convenience to users, stations in one cluster should be close to each other geographically. Thus it is practical for a user whose origin/ destination is close to a station without available bikes/docks to walk/ride to another station in the same cluster to check out/in a bike.

2) In addition, as transition between clusters needs to be predicted later, we would like the inter-cluster transition to be robust in order to improve prediction accuracy. Therefore, the stations in one cluster should have similar transition patterns to all the clusters. Consequently, the transition vector of a cluster C_i, which is a multinomial distribution, would concentrate on several values instead of being more even distributed. For example, comparing two multinomial distributions, $(0.1, 0.1, \ldots, 0.1)$ and $(0.6, 0.4, 0, \ldots, 0)$, it is obvious that the first one is more even distributed but less robust than the second one.

3.1.2 Methodology

Fig. 3 A) presents the iterative procedure of the bipartite clustering algorithm, which organically combines two factors (location and transition) of a station, without needing to assign a weight for each factor, which requires for prior knowledge. Circles with the same color denote bike stations that belong to the same cluster. The algorithm repeats the following *three* steps in each iteration: geo-clustering, t-matrix generation and t-clustering.



Iteration 1
Iteration 2
Iteration 3
Iteration n
Geo-clustering
Transit-matrix generation
Transit-clustering
K1=4
K2=2
A1, A2,...,An
Sx
A) Iterative procedure of the bipartite clustering
TS1: (0.1, 0.3, 0.2, 0.4)
TS2: (0.1, 0.2, 0.4, 0.3)
TS7: (0.5, 0.2, 0.2, 0.1)
Ax=
Transit-Matrix of Sx
B) An example of transit-matrix generation for a station Sx

Fig. 3. Bipartite clustering algorithm procedure

1) *Geo-Clustering*. This step clusters stations into K_1 groups, $\{C_{1,k}\}_{k=1}^{K_1}$, according to stations' geographical locations by K-mean clustering. For the first time, Geo-clustering is conducted on all stations in a sharing system. For the following times, Geo-clustering is conducted on the stations in each cluster obtained in Step 3) proportionally. For example, if the number of stations in each cluster obtained in Step 3) is $N_1, N_2, \ldots, N_{K_2}$, then we cluster the stations in each cluster into $\left[\frac{N_1 K_1}{n}\right], \left[\frac{N_2 K_1}{n}\right], \ldots, \left[\frac{N_{K_2} K_1}{n}\right]$ groups respectively, here $[\cdot]$ is a rounding operator.

2) *T-matrix generation*. Based on clusters obtained in Step 1), we generate a t-matrix for each station. A t-matrix describes a station's transition pattern. It has seven rows, corresponding to seven time slots: 7:00am-11:00am (morning rush hours), 11:00am-4:00pm (day hours), 4:00pm-9:00pm (evening rush hours) and 9:00pm-7:00am (night hours) on weekdays; 0:00am-9:00am (night hours),

9:00am -7:00pm (trip hours) and 7:00pm-12:00pm (evening hours) on weekends/holidays. Each entry, $(A_i)_{l,j}$, is the probability that a bike will be checked in to cluster $C_{1,j}$, given that it is checked out in time slot l from station S_i. The conditional probability is estimated by maximum likelihood estimation from historical bike data. Fig. 3 B) gives a running example of t-matrix generation.

3) *T-clustering.* After obtaining a set of t-matrices, $\{A_i\}_{i=1}^{n}$, we cluster the stations into K_2 clusters, $\{C_{2,k}\}_{k=1}^{K_2}$, according to their t-matrices by K-mean clustering, here $K_2 < K_1$.

Iterate these three steps until the K_1 clusters obtained in Step 1) converge or the iteration threshold K is reached. The K_1 clusters, $\{C_{1,k}\}_{k=1}^{K_1}$, are the final clustering results. The algorithm's pseudo-code is described in Alg. 1.

```
Algorithm 1: Bipartite Clustering Algorithm
Input:  Stations $\{S_i\}_{i=1}^{n}$, historical trips $\{Tr_i\}_{i=1}^{H}$, iteration
        threshold K, parameters $K_1 > K_2$;
Output: $K_1$ clusters: $C_{1,1}, C_{1,2}, \dots, C_{1,K_1}$;
1.  Cluster $\{S_i\}_{i=1}^{n}$ into $K_1$ clusters: $C_{1,1}, C_{1,2}, \dots, C_{1,K_1}$ by K-mean
    based on locations;
2.  Initialize $k = 0$;
3.  While $k < K$ Do
4.      For $i = 1{:}n$ Do
5.          Generate t-matrix $A_i$ of station $S_i$;
6.      Cluster $\{S_i\}_{i=1}^{n}$ into $K_2$ clusters: $C_{2,1}, C_{2,2}, \dots, C_{2,K_2}$ by K-mean
        based on $\{A_i\}_{i=1}^{n}$;
7.      For $j = 1{:}K_2$ Do
8.          Cluster stations in $C_{2,j}$ into $[\frac{N_j K_1}{n}]$ clusters;
9.      Obtain $K_1$ updated clusters: $C_{1,1}, C_{1,2}, \dots, C_{1,K_1}$
10.     If $C_{1,1}, C_{1,2}, \dots, C_{1,K_1}$ do not change Then
11.         Break;
12.     K=k+1;
13. Return $K_1$ clusters: $C_{1,1}, C_{1,2}, \dots, C_{1,K_1}$.
```

Algorithm 1. Bipartite clustering algorithm

3.2 Entire Traffic Learning

In our hierarchical prediction model, the traffic in the higher level, i.e. the entire traffic, is predicted first. The entire traffic is predicted by GBRT [11], which is a non-parametric statistical learning technique for regression. GBRT is one of the most effective machine learning models for prediction. It is flexible enough to fit complex nonlinear relationships.

The general idea of GBRT is to compute a sequence of simple regression trees, $\{g_1(x), g_2(x), \dots, g_r(x)\}$, where each successive tree is built to predict the residual of the preceding trees, as shown by Eq. 1 and Eq. 2:

$$g_i = argmin_g \sum_{t=1}^{N} L(y_t - G_{i-1}(x_t), g(x_t)) \quad (1)$$

$$G_{i-1}(x) = \sum_{l=1}^{i-1} g_l(x) \quad (2)$$

Here, L is a loss function and $\{x_t, y_t\}_{t=1}^{N}$ is the training data set. Predictions are made by combining decisions of $\{g_1(x), g_2(x), \dots, g_r(x)\}$ as shown by Eq. 3:

$$G(x) = g_1(x) + g_2(x) + \dots + g_r(x) \quad (3)$$

In entire traffic prediction, the variables x_t are features corresponding to period t, which have significant influence on the entire traffic; y_t is the ground truth, i.e. the actual entire traffic in period t. Thus, we need to identify the important features first.

Time features: The entire traffic is affected by time. We identify two time features: the hour of the day and the day of the week. Fig. 4 shows the average entire traffic in Aug., 2014, during different hours and on different days. As we can see, the entire traffic on weekdays are similar, consisting of morning rush hours, day hours, evening rush hours and night hours while those on weekends/holidays are similar, consisting of night hours, trip hours and evening hours. The entire traffic on weekdays is much larger than that on weekends/holidays and that in rush hours/trip hours is much larger than those in other time slots. Therefore, the hour of the day and the day of the week are important features in entire traffic prediction.

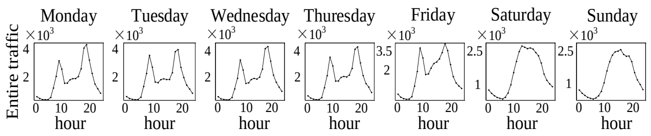


Fig. 4. Average entire traffic in NYC

Meteorology features: Bike is a kind of transportation which is affected by meteorology significantly. Three major meteorology features are identified: weather, temperature and wind speed. Fig. 5 A) shows the average entire traffic in NYC during 6:00am-7:00am, 7:00am-8:00am and 8:00am-9:00am on different days in Aug., 2014. For each special period, we can discover an obvious pattern except for the points in red circles. These exceptional points all correspond to 13th, Aug., which were rainy. A large number of this kind of exceptive examples in historical data support the conclusion that weather influences the entire traffic significantly. Based on historical data, we categorize all weather patterns into four categories: snowy, rainy, foggy and sunny. Fig. 5 B) shows the total traffic every day from Feb. to Aug., 2014, in NYC. As we can see, there is an increasing trend. This is because that the temperature keeps increasing from Feb. to Aug. Fig. 5 C) describes the average entire traffic in D.C. during 8:00am-9:00am in Sep., 2014. The anomalous point in the red circle corresponds to 25th, Sep., which was a windy day. Lots of similar anomalous points related to wind speed can be found on other days as well. In summary, three meteorology features: weather, temperature, and wind speed, affect the entire traffic significantly.

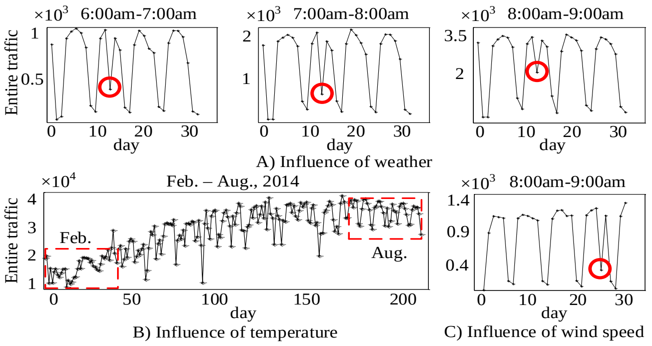


Fig. 5. Average entire traffic

After identifying the time and meteorology features, f_t, influencing the entire traffic, we can obtain a historical dataset $\{(f_t, y_t)\}_{t=1}^{T}$ to train a GBRT model for online prediction.

3.3 Cluster Check-out Proportion Learning

3.3.1 Insights

To allocate the entire traffic to each cluster, we predict each cluster's check-out proportion first. Based on the predicted entire traffic and proportion across clusters, each cluster's check-out can be easily calculated. A multi-similarity-based inference model is proposed in this section. This model has two advantages:

1) The model can handle the unbalanced meteorology distribution problem. Fig. 6 A) shows the weather distribution in NYC from 1st, Apr. to 30th, Sep. (4392 hours), most of which are sunny hours while only two hours are snowy (snowy hours cannot be seen in

Fig. 6 A) as its proportion is too small to appear). If we partition the data and learn the prediction model under each special weather category, there would be a data sparsity problem. If a regression model, e.g. linear regression, is adopted, the model is trained to fit the majority of observations, e.g. sunny hours, thereby scarifying the accuracy of the model under minor conditions, e.g. rainy hours. In addition, as shown in Fig. 6 B), the historical temperature & wind speed scenarios in NYC from 1st Apr. to 30th Sep., many scenarios did not appear historically but could possibly happen in the future. To those 'missing' meteorology scenarios, we cannot predict solely by partitioning data.

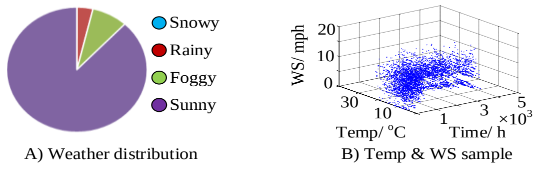


Fig. 6. Unbalanced meteorology in 2014

2) The model can guarantee that the sum of the check-out proportion across clusters is 1. Similar to the interaction between stations, clusters have influence between each other as well. This constraint incorporates the interaction between clusters into the model. If we predict the check-out of each cluster individually, the influence between clusters will be ignored.

Our multi-similarity-based inference model integrates three similarity functions between features of the period to be predicted and those of historical ones:

1) Time similarity $\lambda_1(t_1, t_2)$. Intuitively, the check-out proportion across clusters changes from time to time. *First*, the proportion changes from hour to hour in a day. For example, to the clusters which mainly consist of stations near residential areas, their check-out proportions in morning rush hours would be much larger than those in evening rush hours. *Second*, the proportion changes from day to day in a week. For example, for the clusters which mainly consist of stations close to tourist areas, their check-out proportions in trip hours on weekends/holidays would be much larger than those on weekdays.

2) *Weather similarity* $\lambda_2(w_{t_1}, w_{t_2})$. As proportion across clusters is a vector instead of a single value, we illustrate how to visualize it first. Euclidean distance is adopted to measure the difference of two proportion vectors (i.e. proportion distance). Analyzing NYC's and D.C.'s historical bike data, we extract a proportion distance pattern over one week shown in Fig. 7 C) (stations are clustered as Fig. 2 A)). Each node describes the distance between two proportion vectors in a special hour on two successive days. As we can see, the distances between vectors on two successive weekdays, e.g. Mon. and Tue., denoted by green circles are small and similar while those between vectors on a weekday and a weekend/holiday, e.g. Fri. and Sat., denoted by red circles, are much larger. Unexpectedly, the distance between vectors on Sat. and Sun., denoted by a yellow circle, is not as small as those between two weekdays. We think this is because people usually treat the first and the last day of a holiday differently. However, as it is much smaller compared with those between a weekday and a weekend/holiday, we do not differentiate them. The pattern is used to detect anomalies.

Fig. 7 A) shows the distance of two check-out proportion vectors during 7:00am-8:00am on two successive days in Aug., from the bike-sharing system in NYC, e.g., the point (5, 0.0298) means the distance between the check-out proportion vectors during 7:00am-8:00am on 5th Aug. and on 6th Aug. is 0.0298. Compared with Fig 7. C), points in solid circles are anomalous while those in the dash circles match the pattern well. The two points in the solid circle correspond to two pairs of successive days: 12th Aug. and 13th Aug., 13th Aug. and 14th Aug. Among these three time periods, 7:00am-8:00am on 13th Aug. is a rainy hour while the other two are sunny hours. Different weather patterns lead to larger difference between check-out proportion vectors. Lots of this kind of points can be found, which supports our conclusion that weather influences check-out proportion across clusters significantly.

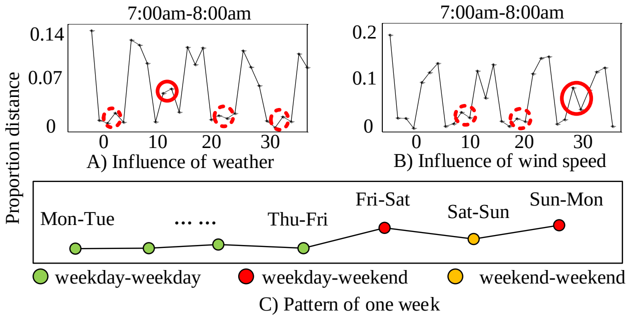


Fig. 7. Check-out proportion distance in 2014

3) *Temperature & wind speed similarity* $K\left((p_{t_1}, v_{t_1}), (p_{t_2}, v_{t_2})\right)$. Fig. 7 B) shows the distance of proportion vectors during 7:00am-8:00am on two successive days of Sep. in D.C. The anomalous points in solid circles correspond to two pairs of successive days: 24th Sep. and 25th Sep., 25th Sep. and 26th Sep., among which, 25th Sep. is a windy day while the other two days have low wind speeds. Other points of this kind can be found to confirm that wind speed influences the check-out proportion across clusters significantly. In a similar way, temperature is identified as another important feature and we do not show here for space reason.

We consider the three factors of meteorology, weather, temperature and wind speed with two different similarity functions instead of one for two reasons: *First*, the values of weather are discrete while those of temperature & wind speed are continuous, whose discretization needs prior knowledge. *Second*, to temperature & wind speed, there are 'missing' scenarios which do not exist to weather.

3.3.2 Methodology

Assume that, $1, 2, \dots, H$ are the H most recent periods to t, whose check-out proportion across clusters we want to predict. Denote their corresponding check-out proportions as $P_1, P_2, \dots, P_H$ and P_t, features as $f_1, f_2, \dots, f_H$ and f_t. Therefore, P_t can be predicted by a multi-similarity-based inference model shown in Eq. 4:

$$\widehat{P}_t = \frac{\sum_{i=1}^{H} W(f_i, f_t) \times P_i}{\sum_{i=1}^{H} W(f_i, f_t)} \qquad (4)$$

The multi-similarity function, $W(f_i, f_t)$, is obtained by Eq. 5:

$$min_W \sum_{t=H+1}^{T} L(E_t \times P_t, E_t \times \widehat{P}_t) \qquad (5)$$

Here, T is the sample size of historical data. $E_t \times P_t$ and $E_t \times \widehat{P}_t$ stand for the ground truth and prediction value of check-out across clusters, respectively; L is a loss function used to measure the prediction error. The multi-similarity function W has three components: time similarity, weather similarity, and temperature & wind speed similarity, as shown in Eq. 6:

$$W(f_i, f_t) = \lambda_1(i, t) \times \lambda_2(w_i, w_t) \times K((p_i, v_i), (p_t, v_t)) \qquad (6)$$

Time Similarity: Intuitively, check-out proportions corresponding to the same hour of a day are more similar than those corresponding to different hours, given that the other features are similar. In addition, if two proportion vectors both belong to weekdays or weekends/holidays and correspond to similar other features, the more closed the two days are, the more similar these two vectors should be. For example, the check-out proportion across clusters in 8:00am-9:00am yesterday is more similar with that in 8:00am-

9:00am today than that in the same period one month ago, given that their other features (day of the week, meteorology) are similar. Thus, we define a time similarity function as Eq. 7:

$$\lambda_1(t_1,t_2) = 1_{t_1,t_2} \times \rho_1^{\Delta h(t_1,t_2)} \times \rho_2^{\Delta d(t_1,t_2)} \tag{7}$$

$$\Delta h(t_1,t_2) = min\{r(t_1,t_2), 24 - r(t_1,t_2)\} \tag{8}$$

$$r(t_1,t_2) = mod(|t_1 - t_2|, 24) \tag{9}$$

$$\Delta d(t_1,t_2) = \left[\frac{|t_1-t_2|}{24}\right] \tag{10}$$

Here, $1_{t_1,t_2} = 1$ if t_1 and t_2 are both on weekdays or weekends /holidays, otherwise, $1_{t_1,t_2} = 0$. Δh measures the distance of two time periods from the perspective of the hour of the day. For example, assuming historical data in Jun., 2014 as the training data, there are 720 hours: $t = 1, 2, 3, \dots, 720$. Then, $\Delta h(12,34) = 2$ means that, from the perspective of time of day, the distance between $t = 12$ corresponding to 12:00am on 1st Jun. and $t = 34$ corresponding to 10:00am on 2nd Jun. is 2 hours. Similarly, $\Delta h(12,38) = 2$ means the distance between 12:00am on 1st Jun. and 2:00pm on 2nd Jun. is 2 hours as well. Δd measures the distance of two time periods from the perspective of the day of the week, e.g. $\Delta d(12,34) = \Delta d(12,38)$ =1.

Weather similarity: The weather patterns are categorized into four categories: snowy, rainy, foggy and sunny. We use a symmetric similarity matrix with *six* parameters: $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ to define the weather similarity function $\lambda_2(w_{t_1}, w_{t_2})$. The similarity matrix is shown in Fig. 8. The value of the function can be looked up from the matrix. For example, if $w_{t_1} = rainy, w_{t_2} = sunny$, then $\lambda_2(w_{t_1}, w_{t_2})$ corresponds to the entry in the second row and the forth column: α_5.

$$\begin{array}{r|cccc} & \text{snowy} & \text{rainy} & \text{foggy} & \text{sunny} \\ \hline \text{snowy} & 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ \text{rainy} & & 1 & \alpha_4 & \alpha_5 \\ \text{foggy} & & & 1 & \alpha_6 \\ \text{sunny} & & & & 1 \end{array}$$

Fig. 8. Weather similarity matrix

Based on our experience, we can add some constraints to these six parameters. As we know, the more different two weather patterns are, the smaller the similarity between them is. Thus, we add the following constraints to them:

$$\alpha_1 > \alpha_2 > \alpha_3, \alpha_4 > \alpha_5, \alpha_6 > \alpha_5 > \alpha_3, \alpha_4 > \alpha_2.$$

Temperature & wind speed similarity: Temperature/wind speed domain is continuous. Their historical data has 'missing' scenarios. Therefore, we choose a 2-D Gaussian Kernel function [16] to measure the similarity between (p_{t_1}, v_{t_1}) and (p_{t_2}, v_{t_2}) so that the similarity between 'missing' scenarios can be estimated as well. The 2-D Gaussian Kernel function is as Eq. 11:

$$K\left((p_{t_1}, v_{t_1}), (p_{t_2}, v_{t_2})\right) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\left(\frac{(p_{t_1}-p_{t_2})^2}{\sigma_1^2} + \frac{(v_{t_1}-v_{t_2})^2}{\sigma_2^2}\right)} \tag{11}$$

As the prediction errors of successive time periods are not independent, we add an error correction item to the multi-similarity-based inference model. Thus, the multi-similarity-based model we adopt practically is as Eq. 12:

$$\widehat{P_t} = \frac{\sum_{i=1}^{H} W(f_i, f_t) \times P_i}{\sum_{i=1}^{H} W(f_i, f_t)} + \sum_{j=1}^{J} \psi_j e_{t-j} \tag{12}$$

Here, the added items $e_{t-j} = P_{t-j} - \widehat{P_{t-j}}$ are the prediction errors of periods $t - j, j = 1, 2, \dots, J$; J is a threshold of time lag.

3.4 Inter-cluster Transition Learning

In order to guarantee the causality between check-out and check-in, we predict each cluster's check-in based on their check-out. Inter-cluster transition is an important factor in measuring the performances of checked out bikes.

Definition 5: *Transition Probability.* The transition probability from cluster C_i to C_j in time t is the probability that a bike will be checked in to cluster C_j given that it is checked out from C_i in time t. Note that only the check-out time is constrained to t while the check-in time can be a random possible value.

Definition 6: *Inter-cluster Transition Matrix.* An inter-cluster transition matrix in t is a matrix $T_{t,m\times m}$ corresponding to period t, each of whose entry, T_{t,C_i,C_j} is a transition probability from cluster C_i to cluster C_j in time t.

Inter-cluster transition matrix describes the transition probability between clusters. Similar to the check-out proportion across clusters, the inter-cluster transition matrix is affected by the correlations between clusters, the time, and the meteorology as well. Thus, we adopt the same model, multi-similarity-based inference model, to predict the inter-cluster transition matrix.

3.5 Trip Duration Learning

Trip duration between each pair of clusters is another important factor in measuring the behavior of a checked out bike. The trip duration distribution between a pair of clusters is mainly determined by the locations of stations in them, which are spatially instead of temporally determined. In bike traffic, jam is no longer an important factor that affects trip duration, thus under most meteorology scenarios except for the really severe ones, trip duration does not change too much. We assume that the trip duration distribution between each pair of clusters is constant. The severe meteorology scenarios are ignored because we have discovered that the severe ones mainly affect the bike traffic demand instead of the trip duration according to historical bike data. We think this is because if a customer chooses to ride a bike on a snowy day, for example, this means that the snow may not be a big obstacle on their ways to the destination, otherwise, they may prefer another kind of transportation, which results in a decrease in bike traffic demand.

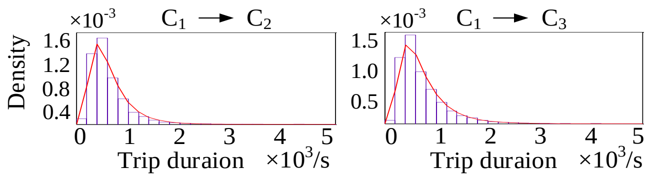


Fig. 9. Trip duration distribution

According to NYC's bike data in 2014, we fit the trip duration between each pair of clusters by a lognormal distribution. As shown in Fig. 9, lognormal distribution fits the real data very well. By maximum likelihood estimation, we obtain a symmetric matrix $D_{m\times m}$ (i.e. a trip duration matrix). Each entry D_{ij} is a vector (μ_{ij}, σ_{ij}) corresponding to the two parameters of a lognormal distribution, describing the trip duration between cluster C_i and C_j.

4. Online Process

Check-out inference: In online prediction process, for a future period t, we first extract its feature f_t and then leverage the three models trained above to obtain the entire traffic prediction E_t, check-out proportion prediction P_t and inter-cluster transition matrix prediction $T_{t,m\times m}$, respectively. Then the check-out of each cluster C_i can be easily calculated by Eq. 13:

$$O_{C_i,t} = E_t \times P_{t,i} \quad (13)$$

Check-in inference: To predict each cluster's check-in, we can adopt the same model with that used for check-out prediction: predict the total check-in of a city; predict the check-in proportion across clusters; and calculate each cluster's check-in. In this section, we introduce another algorithm to infer the check-in of each cluster based on the check-out, inter-cluster transition matrix and trip duration. The insight is that this can guarantee the causality between check-out and check-in. Thus, if an anomaly happens, its influence to check-in can be predicted at the same time when it influences the check-out instead of after a time delay, as check-in is always after check-out. We will show the effectiveness of this inference algorithm under anomalous scenarios with experiments later. In application, these two methods for check-in prediction can be used complementarily: the first one for common scenarios and the second one for anomalous scenarios. The details of the second inference algorithm are illustrated as follow.

We divide the check-in into two parts. Assume that the current time period is t (Denote its corresponding time interval as $(\tau-\delta,\tau]$.), then the check-in to cluster C_i in $t+\delta$ are the bikes which have been checked out before $t+\delta$ and will be checked in to C_i in $t+\delta$, and the bikes that will be checked out in $t+\delta$ and be checked in to C_i in $t+\delta$.

Denoting the bikes that have been checked out before $t+\delta$ but have not been returned as $\{B_1, B_2, \dots, B_{n_1}\}$, to each of them, B_j, we know its original cluster C_{B_j} and its check-out time τ_{B_j}. Based on the inter-cluster transition matrix and trip duration, we can infer its probability to be checked in to cluster C_i in $t+\delta$ as Eq. 14.

$$P_{B_j,i} = T_{t,C_{B_j},C_i} \times \int_{\tau-\tau_{B_j}}^{\tau+\delta-\tau_{B_j}} D_{B_j,i} \quad (14)$$

Therefore, the expectation of the number of bikes in $\{B_1, B_2, \dots, B_{n_1}\}$ that will be checked in to cluster C_i in $t+\delta$ can be predicted as Eq. 15.

$$E_{1,i} = \sum_{B_j} P_{B_j,i} \quad (15)$$

For bikes that will be checked out in $t+\delta$, we can predict their check-in to cluster C_i in $t+\delta$ in a similar way. Assuming that each cluster's check-out in $t+\delta$ is $\{O_{C_1,t+\delta}, O_{C_2,t+\delta}, \dots, O_{C_m,t+\delta}\}$, we divide δ, which is set as 1 hour in our work, into 60 time slots $\{ \left(\tau, \tau+\frac{\delta}{60}\right], \left(\tau+\frac{\delta}{60}, \tau+\frac{\delta}{30}\right], \dots, (\tau+\frac{59\times\delta}{60}, \tau+\delta] \}$. In other words, we set every minute as a time slot. Then the average number of bikes checked out in each time slot Δt from cluster C_j is $\frac{O_{C_j,t+\delta}}{60}$. These bikes have $60-\Delta t$ time left to be returned in $t+\delta$. Therefore, we can obtain the expectation of the number of bikes that checked out and checked in to C_i during $t+\delta$ by Eq. 16.

$$E_{2,i} = \sum_{\Delta t=1}^{60} \sum_{j=1}^{m} \frac{O_{C_j,t+\delta}}{60} \times T_{t+\delta,C_j,C_i} \times \int_0^{60-\Delta t} D_{ji} \quad (16)$$

Consequently, the check-in at cluster C_i in $t+\delta$ is the sum of these two expectations shown in Eq. 17.

$$I_{C_i,t+\delta} = E_{1,i} + E_{2,i} \quad (17)$$

5. Experiments

5.1 Settings

5.1.1 Datasets

We conduct experiments on four datasets (bike data and meteorology data) from NYC and D.C., as presented in Table 2. For bike data, the records with a trip duration less than 1 minute are considered as noise data and not taken into account. For meteorology data, some hours are without a record. We complete the missing meteorology data according to the records in their previous and next hours; i.e. for missing temperature/wind speed, the average value of those in its previous and next periods is used; for missing weather, the value in the previous period is used.

NYC Data: We use the data of Citi Bike system, which is in NYC, from 1st Apr. to 30th Sep. in 2014 as the bike data. There are 5,359,995 records. The data format is: (trip duration, start station ID, end station ID, start time, end time). We transfer them into a trip set $\{(S_{i,o}, S_{i,d}, t_{i,o}, t_{i,d})\}_{i=1}^{5,359,995}$ .We use the meteorology data of NYC, from 1st, Apr. to 30th, Sep., 2014. The data format is: (weather, temperature, wind speed). We extract a meteorology record for each hour and transfer them into a meteorology set $\{w_i, p_i, v_i\}_{i=1}^{4,392}$. We set the data from 1st Apr. to 10th Sep. as training data and those from 11th to 30th Sep. as testing data.

D.C. Data: We use the data of Capital Bikeshare system, which is mainly in D.C., from 1st Apr. to 30th Sep. in 2014 as the bike data. There are 1,886,144 records: $\{(S_{i,o}, S_{i,d}, t_{i,o}, t_{i,d})\}_{i=1}^{1,886,144}$. The meteorology data in D.C., from 1st, Apr. to 30th, Sep., 2014 are transferred into a meteorology set $\{w_i, p_i, v_i\}_{i=1}^{4,392}$. We set the data from 1st Apr. to 10th Sep. as training data and those from 11th to 30th Sep. as testing data.

Table 2. Details of the datasets in 2014

Data Sources			NYC	D.C.
Time Span			1st, Apr-30th, Sep	1st, Apr-30th, Sep
Bike Data	# Stations		344	351
	# Bikes		6,800+	3000+
	# Records		5,359,995	1,886,144
Meteo-rology Data	Weather (# hours)	Snowy	2	0
		Rainy	231	149
		Foggy	303	150
		Sunny	3856	4093
	Temperature / °C		[0, 33]	[−2,36]
	Wind speed / mph		[0,18]	[0,29]

5.1.2 Baselines & Metric

The methods proposed in our work to predict the check-out and check-in are respectively denoted as: *Hierarchical prediction (HP) based on bipartite clustering (BC) and multi-similarity-based inference (MSI)*; *check-in prediction based on bipartite clustering, inter-cluster transition and trip duration (P-TD).* In order to confirm the effectiveness of our models, we conduct experiments to compare our methods with ***nine*** baselines.

HA: We predict the check-out/in by the average value of historical check-out/in in the corresponding periods. E.g., for 1:00pm-2:00 pm on Friday, its corresponding periods are all the historical time intervals from 1:00pm to 2:00pm on weekdays.

ARMA: The check-out/in is a time series. Thus it can be predicted by ARMA, which is a common tool for understanding and predicting future values in a time series. Similar to experiments with HA, in experiments with ARMA, we differentiate the hour of the day and the day of the week as well.

GBRT: Similar to entire traffic prediction, each cluster's check-out/in can be predicted by GBRT directly and individually.

HP-KNN: Hierarchical prediction method based on KNN is used. We predict the entire traffic of the city first and then allocate the entire traffic to each cluster based on the proportion across clusters, which is predicted by KNN prediction [3].

GC: Uniform geographical grid clustering. This means we divide the city into uniform grids. The stations which fall into the same grid form a cluster. GC is commonly used in many works. In our work, we use GC instead of K-mean clustering because the stations in a city (especially those in NYC) are distributed almost evenly instead of into communities. However, K-mean can still be used.

The ***nine*** baselines compared with *HP-BC-MSI* and *P-TD* are: *1)* HA based on GC; *2)* HA based on BC; *3)* ARMA based on GC; *4)* ARMA based on BC; *5)* GBRT based on GC; *6)* GBRT based on BC; *7)* HP based on GC and KNN; *8)* HP based on BC and KNN; *9)* HP based on GC and MSI/ P-TD.

Metric: The metrics we adopt to measure the results are *Root Mean Squared Logarithmic Error (RMLSE)* and *Error Rate (ER)*.

$$RMLSE = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\log(\widehat{X_{C_i,t}}+1)-\log(X_{C_i,t}+1))^2}$$

$$ER = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{i=1}^{m}|\widehat{X_{C_i,t}}-X_{C_i,t}|}{\sum_{i=1}^{m}X_{C_i,t}}$$

Here, $X_{C_i,t}$ is the ground truth of the check-out/in of cluster C_i during t while $\widehat{X_{C_i,t}}$ is the corresponding prediction value.

5.1.3 Anomalous periods

An anomaly is a period which satisfies one of the following two conditions: 1) The entire traffic in this period is much different from the pattern; 2) The check-out/in across clusters in this period is much different from the pattern. For example, Fig. 10 shows the entire traffic in every hour from 11th, Sep. to 17th, Sep. in NYC. Points in red rectangles are anomalous periods whose entire traffics deviate from the pattern significantly. The ones satisfying the second condition can be detected in a similar way.

As the number of anomalous periods in our testing data are small and their deviations are obvious, we can detect them manually. However, for massive detection, a formal definition is required. We can define 'much different' as the data deviate with c×σ from the mean, here c is a parameter, σ is the standard deviation of historical data. In our experiments, we detect 18 and 12 anomalous periods in NYC's and D.C.'s testing data, respectively.

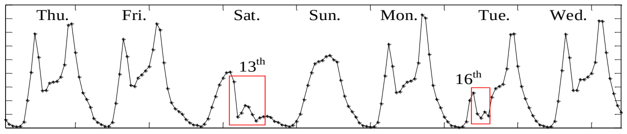


Fig. 10. Anomalous periods

5.2 Results

Results of clustering: It is intuitive that the larger the number of clusters is, the lower the prediction accuracy will be. When there is only 1 cluster, its check-out is the entire traffic which can be predicted with a high accuracy; when there are n clusters, which means that each station forms a cluster, the check-out/in of a cluster fluctuates tremendously and an accurate prediction is difficult if not impossible. However, on the other side, the number of clusters cannot be too small because if the clusters are too large, e.g. 1 cluster containing all the stations, reallocating bikes to clusters cannot offer convenience to users. Therefore, the number of clusters should be chosen by knowledge and experiences.

In our experiments, we cluster all the stations in the two bike-sharing systems in NYC and D.C. into 23 and 27 clusters respectively, by two clustering methods, GC and BC. As shown in Fig. 11, the stations in one cluster are close to each other. However, the results of BC are much different from those of GC because of the additional transition pattern constraint.

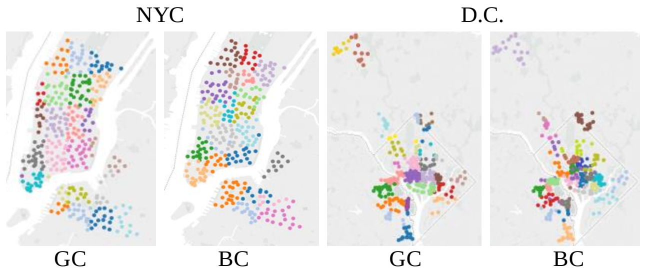


Fig. 11. Clustering Results

Results of Check-out: Based on each clustering result, we compare the check-out prediction results of different methods: HA, ARMA, GBRT, HP-KNN and HP-MSI (We concentrate on the results of NYC data as those of D.C. data are similar). Table 3 shows the average RMLSE and ER of all the predicted hours (common and anomalous hours) and the anomalous hours, respectively.

In Table 3, as we can see, when adopting HP-KNN and HP-MSI, the results obtained under BC are better than those under GC because both of them are hierarchical prediction methods and BC can improve the accuracy of proportion prediction. We propose BC to guarantee the inter-cluster transition matrix to be more robust. Although to ensure more robust check-out proportion is not considered explicitly in the procedure of BC, it is a byproduct. This is because, by BC, the stations clustered into one group perform similarly in transition. Similar transition patterns of stations usually mean similar points of interest (POIs) around them. Stations having similar POIs are more likely to have similar check-out/in performances. Thus, in a special time period of a day, the check-out proportion vector would concentrate on several values instead of being more evenly distributed.

When adopting HA, ARMA and GBRT, BC is not always better than GC. This is because each cluster's check-out is predicted individually. Proportion across clusters, which can be predicted more accurately by BC, is not required.

For anomalous hours, the RMLSE and ER of HA and ARMA are very large because neither of these two models have taken any external features into account. But on the contrary, the performances of GBRT, HP-KNN and HP-MSI are much better as they all consider temporal features when predicting. Note that although HP-KNN performs better than GBRT when comparing their results for all the predicted hours, its accuracy for anomalous hours is lower than that of GBRT. This is because HP-KNN only takes temporal features into consideration for higher level prediction but not for lower level prediction. HP-MSI, which considers the features for both higher and lower level predictions, performs the best no matter for common or anomalous conditions.

In summary, compared to GBRT (common used in practical problems), HP-MSI based on BC can reduce ER by 0.03 for all the predicted hours and 0.18 for anomalous hours.

Results of Check-in: Table 4 shows the average RMLSE and ER of check-in prediction for all the predicted hours and anomalous hours. The analysis to HA, ARMA, GBRT and HP-KNN on data from NYC is similar with that of check-out prediction. The only difference is that we add another prediction method, P-TD, to predict the check-in of each cluster. As we can see, the RMLSE and ER of P-TD for all the predicted hours are larger than those of HP-MSI. However, for anomalous hours, they are much smaller. We explain the observations with two reasons: (1) P-TD is based on the check-out of each cluster, which is obtained by HP-MSI. Thus the additional predictions of inter-cluster transition matrix and trip duration add error. (2) In anomalous periods, as the check-out reduces significantly, the majority of the checked in bikes come from the ones that were checked out before. As illustrated in Fig. 12, the prediction error to the bikes checked out before is smaller than that of bikes checked out in the predicted period. Thus it is obvious that the less bikes in the pink rectangle (bikes checked out in the predicted period), the smaller the error.

In summary, compared to HP-MSI based on BC, P-TD based on BC can further reduce ER by 0.05 for anomalous hours. Because of their different performances under different scenarios, in implementation, we adopt HP-MSI and P-TD for check-in prediction complementarily: HP-MSI for common hours and P-TD for anomalous hours. The results of experiments on data from D.C., shown in Table 3 and Table 4, are similar with those on NYC data, confirming that our model is applicable to different systems.

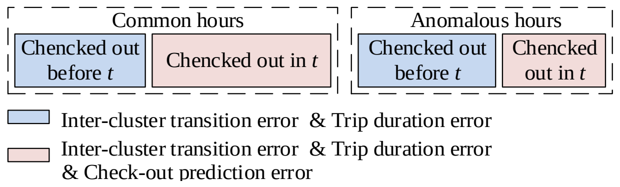

Common hours
Chencked out before t
Chencked out in t
Anomalous hours
Chencked out before t
Chencked out in t
Inter-cluster transition error & Trip duration error
Inter-cluster transition error & Trip duration error & Check-out prediction error

Fig. 12. Error analysis

6. Related Work

6.1 Prediction in Bike-sharing System

Bike-sharing system has received increasing attention since its beginning in Europe. Demaio [9] and Shaheen et al. [18] overviewed its history, analyzing the benefits and detriments in the past and present, and gave a look into its future. Studies on bike-sharing system are summarized into four categories: system design, system pattern analysis, system prediction and system operation.

System design: To set up a bike-sharing system, we need to make sure there is demand from citizens, after which reasonable design is required. Methodologies in [10] estimated the potential demand for bike and the willingness of users to pay in a city; designed the locations for stations and the price policies of a sharing system. Lin et al. [14] proposed a mathematical model for system planers to determine the number and locations of the stations, the network structure of bike paths connected stations, as well as the travel paths for users between each pair of origin and destination. This work addressed the system design problem in an integrated view: setup cost, reallocation cost and travel cost.

System pattern: Understanding the behavior pattern of a bike-sharing system helps to know the mobility of a city. Jon Froehlich et al. [12] and Kaltenbrunner et al. [13] provided a spatiotemporal analysis of Barcelona's bike station usage pattern by clustering techniques. Borgnat et al. [4][5] detected the mobility pattern of Vélo'v by interpreting the system as a dynamical network and analyzing how the flows are distributed spatially along the network. Vogel et al. [19] adopted clustering and clustering validation to analyze the bike usage pattern in Vienna. Bargar et al. [1] compared usage patterns between different sharing systems and revealed the difference in ridership between them. The authors also performed community detection to detect areas with high connectedness. Works in [8] proposed a model to analyze the patterns in different areas of a city by different functions, considering the latent factors of each station.

System prediction: Prediction is another important topic. Jon Froehlich et al. [12] compared *four* simple predictive models to predict the availability of bikes at each station: last value, historical mean, historical trend and Bayesian network. Kaltenbrunner et al. [13] adopted a statistical model to predict the number of available bikes and docks for each bike station. Borgnat et al. [4] used the bike data of Lyon's bike-sharing system, Vélo'v, to predict the entire traffic in each hour of the day by a combination model: the non-stationary amplitude for a given day added with the fluctuation at a specific hour. Vogel et al. [19][20] adopted time series analysis to forecast the bike demand in Vienna. Based on bike data from Dublin, Yoon et al. [22] proposed a modified ARIMA model, considering spatial interaction and temporal factors, to predict the available bikes/docks at each station.

System operation: In system operation, reallocation of bikes is necessary in order to compensate for the unbalanced bike usage. Workers usually reallocate the bikes by vehicle. Contardo et al. [7], Benchimol et al. [2], Chemla et al. [6] presented mathematical formulations to route vehicles to transit the bikes, considering external features, such as the capacity of a vehicle, how unbalanced

Table 3. Prediction error of check-out across clusters

Method	All Hours								Anomalous Hours							
	RMLSE				ER				RMLSE				ER			
	NY		WA		NY		WA		NY		WA		NY		WA	
	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC
HA	0.387	0.372	0.439	0.451	0.353	0.355	0.453	0.489	1.038	1.027	**0.653**	0.715	1.964	1.968	2.111	2.136
ARMA	0.371	0.354	0.413	0.421	0.346	0.346	0.416	0.445	1.114	1.105	0.680	0.722	2.276	2.273	2.245	2.109
GBRT	0.386	0.369	0.423	0.425	0.311	0.314	0.371	0.375	0.647	0.621	0.686	0.670	0.696	0.683	0.830	0.847
HP-KNN	0.377	0.358	0.424	0.410	0.298	0.299	0.364	0.359	0.664	0.642	0.685	0.694	0.692	0.685	0.836	0.838
HP-MSI	0.371	**0.349**	0.421	**0.407**	0.288	**0.282**	0.351	**0.347**	0.646	**0.597**	0.679	**0.664**	0.637	**0.503**	0.794	**0.783**

Table 4. Prediction error of check-in across clusters

Method	All Hours								Anomalous Hours							
	RMLSE				ER				RMLSE				ER			
	NY		WA		NY		WA		NY		WA		NY		WA	
	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC	GC	BC
HA	0.377	0.365	0.435	0.448	0.347	0.352	0.448	0.485	0.954	0.982	0.617	0.672	1.837	1.835	2.201	2.217
ARMA	0.363	0.352	0.409	0.418	0.340	0.344	0.405	0.445	1.025	1.046	0.631	0.700	2.152	2.143	2.123	2.288
GBRT	0.382	0.365	0.420	0.422	0.309	0.309	0.370	0.375	0.624	0.653	0.689	0.701	0.681	0.671	0.834	0.835
HP-KNN	0.375	0.360	0.415	0.411	0.302	0.295	0.367	0.361	0.659	0.647	0.703	0.686	0.694	0.684	0.830	0.830
HP-MSI	0.365	**0.350**	0.408	**0.402**	0.297	**0.290**	0.353	**0.340**	0.646	0.608	0.675	0.660	0.642	0.506	0.810	0.802
P-TD	0.384	0.373	0.425	0.419	0.335	0.302	0.365	0.359	0.626	**0.598**	0.564	**0.558**	0.498	**0.445**	0.802	**0.789**

the distribution of bikes is, the routes between each pair of stations, etc. Because of the complexity of computation, lower bound methods are proposed to extend the models to large databases. Currently, reallocation is executed after monitoring, which is not efficient. Prediction works can help to detect the potential unbalances in advance.

However, previous prediction methods cannot be adopted to our work directly, as our problem is to predict the check-out/in of each station cluster instead of the entire traffic. In addition, data formats are different, e.g. our dataset does not contain information about the number of available bikes/docks at each station while those in [12][13][22] do. As bike traffic is impacted by multiple complex factors, a more flexible model is required to guarantee a higher accuracy.

6.2 Traffic Prediction in Urban Computing

A series of traffic prediction research using data-driven methods has been done in urban computing [25], which aims to use big data to tackle urban challenges. For instance, Shang et al. [17] inferred the travel speed and traffic volume on each road segment, based on GPS trajectories of a sample of vehicles as well as map data and weather conditions. Wang et al. [21] proposed a model to predict time-dependent travel time of a path based on taxi trajectories, POIs, and road networks. Yuan et al. [23][24] presented a Cloud-based system that computes the quickest driving routes based on traffic conditions and driver behaviors. They even predicted the travel time between two landmark locations. Pan et al. [15] proposed to detect traffic anomalies based on GPS trajectories and social media. Our research is also a step towards traffic prediction in urban computing, but different from existing projects in terms of predictive goals.

7. Conclusion

We propose a hierarchical model, which contains a bipartite clustering algorithm, a multi-similarity-based inference model, and a check-in inference algorithm, to predict the check-out/in of each station cluster in a bike-sharing system, based on historical bike data and meteorology data. We evaluate our model on four datasets from NYC and D.C., obtaining performances which are significantly beyond those of baseline methods, especially under anomalous conditions (For NYC data, ER is reduced by 0.03 for all the hours and 0.18/0.23 for anomalous hours. Similar results on D.C. data are obtained.), confirming that our model is better and applicable to different bike-sharing systems. In the future, we would like to consider more factors, such as events, when predicting the traffic under unusual situations.

8. Acknowledgement

This work is supported by Hong Kong RGC Project N, HKUST637/13, National Grand Fundamental Research 973 Program of China under grant 2014CB340303, NSFC Guang Dong Grant NO. U1301253, NSFC Grant No. 61572488, Microsoft Research Asia Gift Grant.

9. Reference

[1] Bargar A., Gupta A., Gupta S., Ma D. 2014. Interactive visual analytics for multi-city bikeshare data analysis. In *Proc. of* the 3rd Urbcomp.

[2] Benchimol M., Benchimol P., Chappert B., Taille A. D. L., Laroche F., Meunier F., and Robinet L. 2011. Balancing the stations of a self-service “bike hire” system. RAIRO-Operations Research, vol. 45, no. 1, pp. 37-61.

[3] Bhatia, N. and Vandana. 2010. Survey of nearest neighbor techniques. International Journal of Computer Science and Information Security, vol. 8, no. 2, pp. 302-305.

[4] Borgnat P., Abry P., Flandrin P., Robardet C., Rouquier J., and Fleury E. 2011. Shared Bicycles in a City: a Signal Processing and Data Analysis Perspective. Advances in Complex Systems, vol. 14, no. 3, pp. 415-438.

[5] Borgnat P., Robardet C., Abry P., Flandrin P., Rouquier J., and Tremblay N. 2013. A dynamical network view of Lyon’s V*é*lo’v shared bicycle system. Dynamics On and Of Complex Networks, vol. 2, pp. 267-284.

[6] Chemla D., Meunier F., and Wolfler-Calvo R. 2011. Balancing a bike-sharing system with multiple vehicles. In *Proc. of* Congress annual de la société Française de recherche opérationelle et d’aidea la décision.

[7] Contardo C., Morency C., and Rousseau L. 2012. Balancing a dynamic public bike-sharing system. CIRRELT, vol. 4.

[8] Côme E., Oukhellou L. 2014. Model-based count series clustering for bike sharing sbystem usage mining, a case study with the V*é*lib’s system of Paris. ACM Transactions on Intelligent Systems and Technology, vol. 5, no. 3, pp. 39:1- 39:2.

[9] DeMaio P. 2009. Bike-sharing: History, impacts, models of provision, and future. Journal of Public Transportation, vol. 12, no. 4, pp. 41-56.

[10] Dell’Olio L., Ibeas A., and Moura J. L. 2011. Implementing bike-sharing systems. Proceedings of the ICE-Municipal Engineer, vol. 164, no. 2, pp. 89-101.

[11] Friedman J. H. 2001. Greedy function approximation: a gradient boosting machine. The Annals of Statistics, vol. 29, no. 5, pp. 1189-1232.

[12] Froehlich J., Neumann J., and Oliver N. 2009. Sensing and Predicting the Pulse of the City through Shared Bicycling. In *Proc. of the 21st IJCAI*.

[13] Kaltenbrunner A., Meza R., Grivolla J., Codina J., and Banches R. 2010. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. Pervasive and Mobile Computing, vol. 6, no. 4, pp. 455-466.

[14] Lin J. and Yang T. 2011. Strategic design of public bicycle sharing systems with service level constraints. Transportation research part E: logistics and transportation review, vol. 47, no. 2, pp. 284-294.

[15] Pan B., Zheng Y., Wilkie D., Shahabi C. 2013. Crowd Sensing of Traffic Anomalies based on Human Mobility and Social Media. In *Proc. of the 23rd ACM SIGSPATIAL GIS*.

[16] Seeger M. 2004. Gaussian Processes for Machine Learning. International Journal of Neural Systems, vol. 14, no. 2, pp. 69-106.

[17] Shang J., Zheng Y., Tong W., Chang E., and Yu Y. 2014. Inferring Gas Consumption and Pollution Emission of Vehicles throughout a City. In *Proc. of the 20th KDD*.

[18] Shaheen S., Guzman S., and Zhang H. 2010. Bikesharing in Europe, the Americas, and Asia: past, present, and future. Transportation Research Record: Journal of the Transportation Research Board, no. 2143, pp. 159-167.

[19] Vogel P., Greiser T., and Mattefeld D. C. 2011. Understanding bike-sharing systems using data mining: Exploring activity patterns. Procedia-Social and Behavioral Sciences, vol. 20, pp. 514-523.

[20] Vogel P. and Mattfeld D. C. 2011. Strategic and operational planning of bike-sharing systems by data mining- a case study. Computational Logistics, pp. 127-141.

[21] Wang Y., Zheng Y., and Xue Y. 2014. Travel Time Estimation of a Path using Sparse Trajectories. In *Proc. of the 20th KDD*.

[22] Yoon J. W., Pinelli F., and Calabrese F. 2012. Cityride: a predictive bike sharing journey advisor. In *Proc. of the 13th IEEE ICMDM*.

[23] Yuan J., Zheng Y., Xie X., and Sun G. 2011. Driving with Knowledge from the Physical World. In *Proc. of the 17th KDD*.

[24] Yuan J., Zheng Y., Zhang C., Xie W., Xie X., Sun G., and Huang Y. 2010. T-Drive: Driving Directions Based on Taxi Trajectories. In *Proc. of the 18th ACM SIGSPATIAL GIS*.

[25] Zheng Y., Capra L., Wolfson O., and Yang H. 2014. Urban Computing: concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology, vol. 5, no. 3, pp. 38:1-38:55.

[26] Data: http://research.microsoft.com/apps/pubs/?id=255961