Dynamic Cluster-Based Over-Demand Prediction in Bike Sharing Systems

Longbiao Chen[1,2,3], Daqing Zhang[2,5], Leye Wang[2], Dingqi Yang[4], Xiaojuan Ma[6], Shijian Li[1], Zhaohui Wu[1], Gang Pan[1], Thi-Mai-Trang Nguyen[3], Jérémie Jakubowicz[2]

[1]Zhejiang University, China; [2]Institut Mines-Télécom, Télécom SudParis, CNRS SAMOVAR, France
[3]Sorbonne Universites, UPMC Univ Paris 06, UMR 7606, LIP6, 4 Place Jussieu, 75005 Paris, France
[4]University of Fribourg, Switzerland; [5]Peking University, China
[6]Hong Kong University of Science and Technology, Hong Kong
{longbiaochen,gpan}@zju.edu.cn, {daqing.zhang, jeremie.jakubowicz}@telecom-sudparis.eu

ABSTRACT

Bike sharing is booming globally as a green transportation mode, but the occurrence of over-demand stations that have no bikes or docks available greatly affects user experiences. Directly predicting individual over-demand stations to carry out preventive measures is difficult, since the bike usage pattern of a station is highly dynamic and context dependent. In addition, the fact that bike usage pattern is affected not only by common contextual factors (e.g., time and weather) but also by opportunistic contextual factors (e.g., social and traffic events) poses a great challenge. To address these issues, we propose a dynamic cluster-based framework for over-demand prediction. Depending on the context, we construct a weighted correlation network to model the relationship among bike stations, and dynamically group neighboring stations with similar bike usage patterns into clusters. We then adopt Monte Carlo simulation to predict the over-demand probability of each cluster. Evaluation results using real-world data from New York City and Washington, D.C. show that our framework accurately predicts over-demand clusters and outperforms the baseline methods significantly.

ACM Classification Keywords

H.2.8 Database applications: Data mining.

Author Keywords

Bike sharing system; over-demand prediction; urban data

INTRODUCTION

In response to the growing concerns over urban sustainability, practices of green transportation such as bike sharing [1] have emerged. Today, more than 700 cities worldwide have launched bike sharing systems [2]. These systems allow people to pick up and drop off public bikes at self-service stations scattered around a city to make short trips. Given the large investment in infrastructure necessary to support a bike sharing system, such as setting up bike stations and renovating bike lanes, it is important for city authorities to ensure that the system is fully functional [3]. One of the key requirements is to prevent stations from *over-demand*, i.e., being completely empty or full over an extended period of time [4, 2]. Users' experiences may be greatly impaired if they run into an over-demand station, as they need to find another available station to rent or return the bike, which may ultimately hinder user participation in the bike sharing system [2, 5]. Therefore, city authorities often urge bike sharing system operators to resolve and prevent the over-demand problem, for example, by issuing fines when it occurs [6].

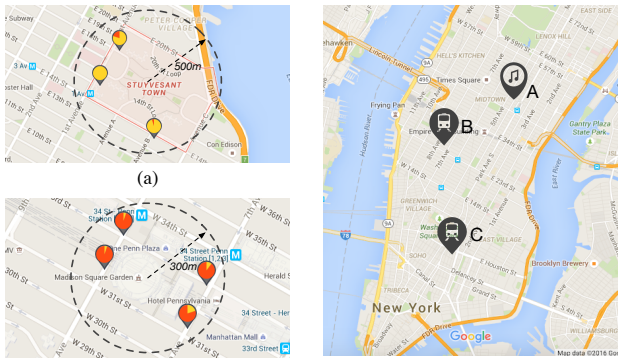


Figure 1. Examples of bike usage patterns in different contexts. (a) Almost empty stations near a residential area in morning rush hours (7:00–8:00, 06/17/2015). (b) Almost full stations near a stadium before a concert (19:00–20:00, 05/13/2014). (c) An opportunistic context with a concert and two subway delays (12:00–13:00, 11/17/2015).

Operators have implemented different strategies to address the over-demand issue [7, 6], such as sending trucks to redistribute bikes before rush hours [8], or setting up temporary bike corrals for large social events to provide extra docks [7]. The ability to accurately foresee over-demand stations in the system is critical to the success of these strategies. However, predicting over-demand of individual stations is difficult as users usually choose a station near their origins or destinations on an ad hoc basis [2]. As a result, existing station-level bike demand prediction methods [9, 10] usually have relatively low accuracy.

UbiComp '16, September 12-16, 2016, Heidelberg, Germany

 ISBN 978-1-4503-4461-6/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2971648.2971652

Based on our observation, while the bike usage of a single station might exhibit high variability, the bike usage of the stations in a certain area over a certain time window (e.g., one hour) can have similar trends. For example, stations near a residential area in morning rush hours usually have more bikes rented than returned (Figure 1(a)), and stations near a stadium usually have a surge in dock demand before concerts (Figure 1(b)). Such bike usage patterns are highly *context dependent* [11, 12]: time of the day, day of the week, weather condition, social events, and traffic conditions can all lead to different bike usage patterns [4, 13, 14, 15]. Hence, we propose to *cluster* neighboring stations with similar bike usage patterns according to context, and predict over-demand at the cluster level. We define an over-demand cluster as a cluster containing at least one over-demand station in a given time window. Although some existing work on bike demand prediction [16, 5] also considers station clustering to boost performance, they usually group stations into static clusters regardless of the context, which do not obtain consistent prediction accuracy when the context varies.

However, clustering stations and consequently predicting over-demand occurrence according to the varied and highly dynamic context is not trivial. In fact, bike usage patterns are mainly impacted by two types of contextual factors: (1) the *common contextual factors* that occur frequently and affect all the stations, such as time and weather, and (2) the *opportunistic contextual factors* that happen irregularly and only affect a subset of stations, such as social and traffic events. An intuitive method to cluster stations according to context is to build a statistical clustering template using historical records (e.g., a cluster template for sunny weekday rush hours). Then, given a specific context in a future time window, we can simply apply its corresponding template to cluster the stations and make cluster-level over-demand prediction. Although this template-based method can cope with the common contextual factors, it does not work well when incorporating the opportunistic contextual factors (events) that have rather few instances in history. In other words, these opportunistic events are sparse in time, making it difficult to find enough historical records containing the same events to generate a template. For example, Figure 1(c) shows a sunny weekday afternoon (12:00–13:00, 11/17/2015) with a concert in a stadium (Event A) and two subway delay events (Event B and C); no historical records having the same context can be found during the period from 01/01/2014 to 12/31/2015. Therefore, we need to design an effective method to model the impact of both common and opportunistic contextual factors simultaneously, which allows us to cluster station and predict over-demand accordingly.

In this paper, we propose a *dynamic cluster-based* framework to predict over-demand occurrence in bike sharing systems according to context. First, we extract the common and opportunistic contextual factors from various urban data [17, 18, 19]. Then, depending on the current context, we construct a weighted correlation network [20] to model the relationship among bike stations. Specifically, we take each station as a node and connect neighboring stations with links. We use the link weight of two stations to model the relationship between them with consideration of both common and opportunistic contextual factors. The link weight of two stations associated with the common contextual factors is calculated based on the correlation between their historical bike usage patterns, such that two stations with similar bike usage patterns have high link weight. The link weight of two stations with respect to the opportunistic contextual factors is calculated based on the number and types of events taking place near the stations, such that two stations impacted by the same array of events have high link weight. We then build the complete network by merging the two sets of link weights, and group highly connected stations into clusters, so that each cluster consists of neighboring stations with similar bike usage patterns. Finally, we estimate the number of bikes rented and returned in each cluster, and predict the cluster over-demand probability accordingly. The contributions of this paper include:

1. To the best of our knowledge, this is the first work on dynamic cluster-based over-demand prediction according to context. Such a dynamic clustering approach leads to high and consistent over-demand prediction accuracy in bike sharing systems.
2. We propose a two-phase framework to predict over-demand clusters by considering both common and opportunistic contextual factors. In the *dynamic station clustering* phase, depending on the context, we build a weighted correlation network to model the relationship among bike stations, and propose a geographically-constrained clustering method to dynamically cluster stations over the network. In the *over-demand cluster prediction* phase, we first estimate the number of bikes rented and returned in each cluster, and then adopt Monte Carlo simulation to predict the cluster over-demand probability.
3. We evaluate the performance of our framework using two years of real-world bike sharing data and urban data in New York City and Washington, D.C. Results show that our framework accurately predicts over-demand clusters across different contexts in both cities (e.g. with 0.882 precision and 0.938 recall in NYC), and outperforms the start-of-the-art methods.

PRELIMINARY AND FRAMEWORK

We define the terms used in this paper as follows.

Definition 1. Station Status: the status of station i at time t is defined as a tuple $\langle B_i(t), D_i(t)\rangle$, where $B_i(t)$ and $D_i(t)$ are the number of available bikes and docks in station i at time t, respectively.

Definition 2. Bike Usage: the bike usage of station i in a given time window $[t, t+\Delta t]$ is defined as a tuple $\langle U_i^-(t), U_i^+(t)\rangle$, where $U_i^-(t)$ and $U_i^+(t)$ are the number of bikes rented from and returned to station i during $[t, t+\Delta t]$, respectively. We further define $U_i^-(t)$ and $U_i^+(t)$ as the *bike rental number* and *bike return number*, respectively, and the sum of absolute values of the bike rental and return number as the *bike usage number*.

Definition 3. Context: we denote the context of a bike sharing system in a time window $[t, t+\Delta t]$ as $\Psi(t) = \langle \Psi_c(t), \Psi_o(t)\rangle$, where $\Psi_c(t)$ denotes the common contextual factors includ-

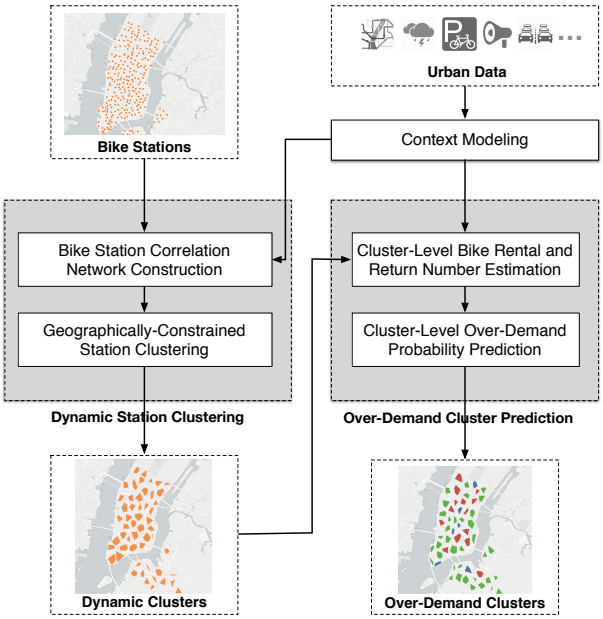


Figure 2. Overview of the framework.

ing time and weather, and $\Psi_o(t)$ denotes the opportunistic contextual factors including social and traffic events.

Definition 4. Over-Demand Station: we define a station i as an over-demand station if the station is full or empty for a period of time longer than a threshold. In this paper, we empirically set the threshold as 10 minutes.

Definition 5. Cluster: we define a set of neighboring stations with similar bike usage patterns in a given time window as a cluster C. We define the bike usage number of a cluster as the sum of the bike usage number of its member stations.

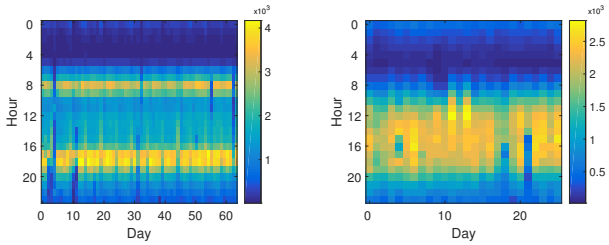
Definition 6. Over-Demand Cluster: we define an over-demand cluster as a cluster containing at least one over-demand station in a given time window[1].

We propose a two-phase dynamic cluster-based framework to predict over-demand occurrence in a bike sharing system according to context. As shown in Figure 2, we extract discriminative features from urban data to model the contextual factors relevant to bike usage, such as weather condition and social events. In the dynamic station clustering phase, we first construct a weighted correlation network to model the relationship among bike stations according to the current context, and then propose a geographically-constrained clustering method to cluster stations over the network. In the over-demand cluster prediction phase, we first estimate the bike rental and return number in each cluster, and then predict the cluster over-demand probability.

CONTEXT MODELING LEVERAGING URBAN DATA

The bike usage pattern of a bike sharing system may be affected by various contextual factors, such as weather condition and social events [13, 14]. Traditionally, collecting city-wide

[1]Our solution in this paper can directly adapt to the definition of 'at least K over-demand stations' if necessary. For clarity, we focus on the definition of $K = 1$ now and discuss it later.



(a) Weekdays (b) Weekends and holidays

Figure 3. The bike usage number of all stations in two months (06/01/2014–07/31/2014).

Table 1. Groups for modeling temporal context

Day type	Group name	Time span
Weekdays	morning rush hours	07:00–11:00
	day hours	11:00–16:00
	evening rush hours	16:00–20:00
	night hours	20:00–24:00
Weekends/Holidays	day hours	09:00–19:00
	night hours	19:00–01:00

context information usually requires substantial time and labor [18]. With the ubiquity of urban sensing infrastructures and paradigms [18], these contextual factors can now be captured at low cost via assorted urban data [17]. However, given the considerable volume and variety of urban data, we need to identify factors relevant to bike usage patterns for modeling contexts. To this end, we conduct a series of empirical studies to analyze the relationship between bike usage number and various contextual factors as follows.

Common Contextual Factors

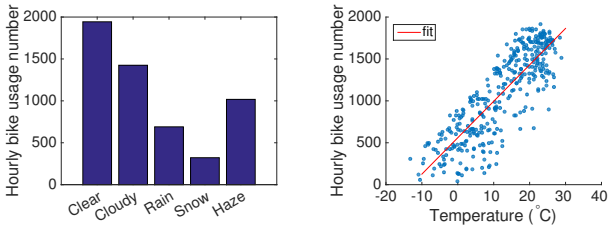
Based on previous studies and surveys [6, 7, 16], the common contextual factors relevant to bike usage patterns usually include *date and time*, *weather condition*, and *air temperature*. By exploiting the bike sharing data from the NYC Citi Bike system [21] and the meteorological data from the Weather Underground API [22], we study the impact of the common contextual factors as follows.

Date and Time

Intuitively, the bike usage pattern of a station might be different according to time of the day, and day of the week. However, there may be correlations and similarities among different temporal groups. Figure 3 shows a sample of the bike usage number of all Citi Bike stations in two months from 06/01/2014 to 07/31/2014. We observe different bike usage patterns between weekdays and weekends/holidays, as well as between different hours of a day. Based on such observations, we derive six different *temporal groups*, as shown in Table 1. Note that we only consider the *active hours* with intensive bike usage, and discard temporal groups of 0:00–7:00 in weekdays and 1:00–9:00 in weekends/holidays.

Weather Condition

As presented in previous studies [23, 7], bike usage patterns may vary significantly under different weather condition, such



(a) Weather condition (b) Air Temperature

Figure 4. The hourly bike usage number of all stations across different meteorological contexts.

as rain or snow. We quantitatively study the relationship between the bike usage number and weather condition leveraging the hourly weather forecast data during the year of 2014. Specifically, we define the following five weather condition categories: *clear, cloudy, rain, snow, and haze*. Figure 4(a) shows the average hourly bike usage number of all stations under different weather condition. We observe that in rainy and snowy days, the bike usage number drops significantly, suggesting that weather condition should be considered as an important contextual factor impacting the bike usage patterns.

Air Temperature

Similarly, air temperature is also considered as an important factor impacting the bike usage patterns [23, 13]. By exploiting the same weather forecast data, we study the relationship between the hourly bike usage number and the air temperature over the year of 2014. As shown in Figure 4(b), we observe strong correlation between the two variables. We empirically split the air temperature range into four groups according to the seasonal temperature variations, i.e. *below zero* ($< 0°C$), *cold* ($[0°C, 10°C)$), *comfortable* ($[10°C, 22°C)$), and *warm* ($\geq 22°C$).

Opportunistic Contextual Factors

The opportunistic contextual factors, including *social events* and *traffic events*, may cause unusual bike usage in a subset of stations near the event locations [14, 23, 24]. For social events, the impact on bike usage may be observed before, during and after the events. As the information about the event time and location is usually posted by organizers in advance, we can model the impact of these social events in the corresponding time windows. For traffic events (e.g., subway delays), the impact on bike usage is usually observed after the occurrence of the events with a delay. As such traffic events are published by urban authorities in real time, we can model the after-event impact for these traffic events.

Social Event

Riding public bikes to attend social events is a convenient transportation mode, especially when there are vehicle restrictions or traffic congestion in the event locations. In order to quantitatively study the impact of social events on bike usage, we collect the event bulletin data from the Eventful API [25]. Figure 5 shows an example event bulletin for a concert with detailed event name, type, time, and location. For each event, we select the stations located within a walking distance τ of the event location (we empirically set $\tau = 620m$ based on experiment results as discussed later), and then compare the bike

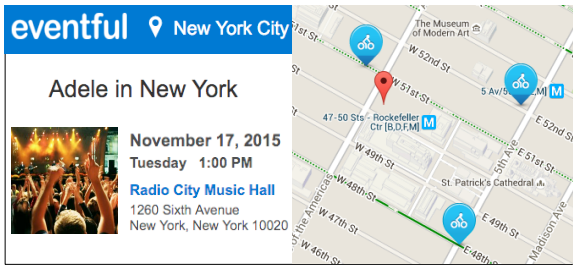


Figure 5. An example event bulletin containing event name, type, time, and location. Bike icon denotes the nearby bike sharing stations.

Table 2. Top 5 most impactive social and traffic event types

Social event	IF	Traffic event	IF
City festival	4.73	Subway delay	1.47
Sports game	3.30	Traffic accident	1.23
Concert	3.24	Road restriction	1.20
Street fair	2.67	Traffic congestion	1.14
Parade/Marathon	2.33	Transit incident	1.11

usage number of these stations from one hour before the event start time to one hour after the event end time with the value in the same time window without event. We define the *impacting factor (IF)* of each event as the ratio of the event-time bike usage number to the normal value, and derive the *IF* of each event type. Table 2 shows the top 5 most impactive social event types on bike usage with regard to the *IF*.

Traffic Event

Previous surveys [7, 1] have shown that people might resort to public bikes as an alternative means to avoid transportation problems, such as subway delays and traffic accidents. We quantitatively study the impact of these traffic events by exploiting the NYC 511 traffic data feed [26] and the subway delay alerts from the NYCT Subway Twitter account [27]. We employ a similar method as mentioned in the social event analysis to calculate the impacting factor for each type of traffic event on its nearby stations in the next hour after the traffic event occurs. The top 5 most impactive traffic event types are also presented in Table 2.

DYNAMIC STATION CLUSTERING

In this phase, our objective is to dynamically group neighboring stations into clusters according to context, so that the stations in the same cluster have similar bike usage patterns. To this end, we first model the relationship among bike stations using a *weighted correlation network* [20], which has been widely used in bioinformatics applications such as gene co-expression network analysis [28, 29]. Specifically, we regard bike stations as nodes, and connect two stations with a link if they are geographically close to each other. We calculate the weight of each link according to the associated common and opportunistic contextual factors, and merge them together to construct the network.

We then group neighboring stations with similar bike usage patterns into clusters. These clusters can be considered as communities that are densely connected internally and loosely connected between each other [30]. In the literature, various algorithms have been proposed to find community structures in a network, such as the Label Propagation algorithm [31]

and the Girvan-Newman algorithm [32]. However, directly applying these algorithms to detect communities may not be adequate in our scenario, since we also need to constrain the geographic span of the formed clusters within a reasonable bound for practical purposes. For example, a single cluster spanning several kilometers is not useful for operators to schedule bike redistribution routes or set up temporary bike corrals. Therefore, we proposed a *Geographically-Constrained Label Propagation (GCLP)* method to solve this problem.

Station Correlation Network Construction

We model the relationship among bike stations as an undirected, weighted network $G = (V, E)$, where $V = \{s_1, \ldots, s_N\}$ denotes the set of N stations, and E denotes the set of links between two stations. We then define the adjacency matrix A of network G, which is an $N \times N$ symmetric matrix with entries $a_{i,j} = 1$ when there is a link between station s_i and station s_j, and $a_{i,j} = 0$ otherwise ($i, j = 1, \ldots, N$). We further determine the weight of each link $w(s_i, s_j)$ based on the common and opportunistic contextual factors.

Adjacency Matrix

By definition, only neighboring stations could be grouped into the same cluster. Therefore, we use the geographic distance of two stations to determine whether they are adjacent or not. More specifically, for station s_i and station s_j, we define:

$$a_{i,j} = \begin{cases} 1, & \text{if } dist(s_i, s_j) \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $dist(s_i, s_j)$ is the geographic distance between the two stations[2], and τ is a *neighborhood threshold* controlling the geographic distance of neighboring stations.

Link Weight

We determine the link weight by considering both common and opportunistic contextual factors as follows:

$$w(s_i, s_j) = a_{i,j} \times (\mu\, w_c(s_i, s_j) + (1 - \mu)\, w_o(s_i, s_j)) \tag{2}$$

where $w_c(s_i, s_j)$ and $w_o(s_i, s_j)$ correspond to the link weight associated with the common and opportunistic contextual factors, respectively, as detailed later. $\mu \in (0, 1)$ controls the influence degree of each type of contextual factor. We consider the case of normalized symmetric positive weights ($w(s_i, s_j) \in [0, 1]$) with no loops ($w(s_i, s_i) = 0$). We note that $w(s_i, s_j) = 0$ when there is no link between s_i and station s_j ($a_{i,j} = 0$).

In order to calculate the link weight associated with the common contextual factors $w_c(s_i, s_j)$, we characterize the two stations by the historical bike usage records having the same common contexts. More specifically, for the two stations s_i and s_j composing the link, we construct a corresponding feature vector $\mathbf{f}_c(s_i) = [U_i^+(t_1), U_i^-(t_1), \ldots, U_i^+(t_K), \ldots, U_i^-(t_K)]$ and $\mathbf{f}_c(s_j) = [U_j^+(t_1), U_j^-(t_1), \ldots, U_j^+(t_K), \ldots, U_j^-(t_K)]$, respectively, using the bike rental and return number of historical records having the same common contexts Ψ_c. We remove records with over-demand stations, since in these situations the observed bike rental or return number may be relatively small and not rewarding the potential demand on the station, as users are not able to rent or return bikes in the station. We then calculate the Pearson correlation coefficient [33] of $\mathbf{f}_c(s_i)$ and $\mathbf{f}_c(s_j)$, denoted as $corr_c(s_i, s_j)$, and normalize it to $[0, 1]$ to obtain the link weight associated with the common contextual factors, i.e.,

$$w_c(s_i, s_j) = \frac{1 + corr_c(s_i, s_j)}{2} \tag{3}$$

In order to calculate the link weight associated with the opportunistic contextual factors $w_o(s_i, s_j)$, we characterize the two stations by the number and type of events taking place near the stations. More specifically, for the two stations s_i and s_j composing the link, we search for the events taking place within the neighborhood threshold τ of each station, and count the number of events by type as defined in Table 2. We construct a feature vector $\mathbf{f}_o(s_i) = [V_i(1), \ldots, V_i(10)]$ and $\mathbf{f}_o(s_j) = [V_j(1), \ldots, V_j(10)]$, where each $V_i(m)$ and $V_j(m)$ ($1 \leq m \leq 10$ since we consider 5 social event types and 5 traffic event types) corresponds to the number of events of type m taking place near station s_i and s_j, respectively. Similarly, we then calculate the Pearson correlation coefficient of $\mathbf{f}_o(s_i)$ and $\mathbf{f}_o(s_j)$, denoted as $corr_o(s_i, s_j)$, and normalize it to $[0, 1]$ to obtain the link weight associated with the opportunistic contextual factors, i.e.,

$$w_o(s_i, s_j) = \frac{1 + corr_o(s_i, s_j)}{2} \tag{4}$$

Geographically-Constrained Station Clustering

Problem Formulation

In this step, we need to group stations into clusters, so that each cluster consists of neighboring stations with similar bike usage patterns. In the constructed station correlation network, as the link weight encodes the similarity between the two nodes, we need to cluster nodes with high link weights together, which can be identified as a community detection problem [32]. Specifically, given the weighted correlation network $G = (V, E)$, we first define a set of clusters $\mathbb{P} = \{C_1, \ldots, C_K\}$, where

$$\cup_{\forall C_k \in \mathbb{P}} = V \quad \text{and} \quad \cap_{\forall C_k \in \mathbb{P}} = \emptyset \tag{5}$$

Then, given a node v, we define the *connectivity* of v to a cluster C as the sum of link weights between v and the nodes in the cluster C:

$$con(v, C) = \textstyle\sum_{v' \in C} w_{v,v'} \tag{6}$$

Finally, we define the *adjacent clusters* $\mathbb{C}(v)$ of node v as

$$\mathbb{C}(v) = \{C | con(v, C) > 0, C \in P\} \tag{7}$$

With the above definition, our objective is to find an optimal set of clusters $\mathbb{P}$, such that the internal connectivity within a cluster is higher than the inter-cluster connectivity, i.e.,

$$\forall v \in C_k,\ con(v, C_k) \geq max\{con(v, C_l), C_l \in \mathbb{P}\} \tag{8}$$

We also need to bound the geographic span of a cluster within the neighborhood threshold, i.e.,

$$\forall v, v' \in C_k,\ dist(v, v') \leq \tau \tag{9}$$

[2]Here we use the city-block distance to approximate the real-world walking or riding distance between stations.

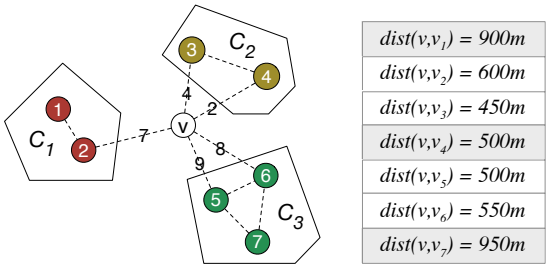


Figure 6. An illustrative example of the node assignment process of the GCLP algorithm. The number on each link denotes the weight, and *dist* is the geographic distance between two nodes.

Clustering Method

To obtain clusters with high internal connectivity (Equation 8) while meeting the geographic constraint (Equation 9) at the same time, we propose the Geographically-Constrained Label Propagation (GCLP) algorithm, which is built on the popular community detection algorithm Label Propagation [31]. The basic idea of GCLP is iteratively assigning nodes to the adjacent clusters, where the gain of assigning node v to cluster C is evaluated by a *value function*. Based on previous discussion, the value function should be designed to reward the connectivity $con(v,C)$ and penalize the geographic span $dist(v,v'), \forall v' \in C$. Therefore, we define the value function as

$$value(v,C) = con(v,C) \times log\Big(\frac{\tau}{max\{dist(v,v')\}}\Big) \qquad (10)$$

The GCLP method greedily assigns the node to the adjacent cluster with highest value[3] until none of the nodes are moved among clusters [31]. As the convergence of such a greedy approach is hard to prove [34], we set a maximum iteration number *max_iter* to ensure that the algorithm will stop.

Example We use an example to illustrate the node assignment process. As shown in Figure 6, node v has three adjacent clusters C_1, C_2, C_3, and the connectivity between v and each adjacent cluster is $7, 4+2, 9+8$, respectively. The maximum distance between v and each cluster is $dist(v,v_1) = 900m, dist(v,v_4) = 500m, dist(v,v_7) = 950m$, respectively. Suppose the neighboring threshold $\tau = 620m$, then the value function of each cluster will yield $-1.13, 0.65, -1.30$, respectively. Hence, we assign node v to cluster C_2 with the highest value.

Algorithm The GCLP algorithm is initialized by assigning each node in the network to a unique cluster label. In each iteration, we randomly populate a list of nodes L, and traverse the list to update the cluster label of each node. The label update process is as follows. First, we remove the node from its current cluster, and find the set of adjacent clusters to the current node. Then, we compute the value function for all the adjacent clusters, and assign the node to the cluster with the highest value. We mark the the node as *moved* among clusters if its new cluster label is different from the old one. After we finish iterating over the node list, we decide whether to perform another iteration or finish the algorithm based on the following stop criteria: (1) the user specified maximum iteration number *max_iter* is reached, or (2) none of the nodes are moved among clusters.

[3]If two clusters yield the same value, we randomly choose one.

Time Complexity For each iteration of the GCLP algorithm, it first takes $O(|V|)$ steps for node permutation, and then processes all the links when computing the value function for each node, taking $O(|V| * |E|)$ steps in the worst case. Since we limit the number of iterations by *max_iter*, the final time complexity of the algorithm is $O(|V| * |E|)$.

OVER-DEMAND CLUSTER PREDICTION

After grouping stations into clusters, our objective in this phase is to predict the occurrence of over-demand clusters. An intuitive method is to directly model the cluster over-demand probability with regard to the contextual factors. However, since the opportunistic contextual factors are sparse in time, it is difficult to find enough samples for a specific context to train the model. Moreover, the ad hoc bike usage behaviors within a cluster also introduce uncertainty in over-demand prediction. To address these issues, we first estimate the bike rental and return number of each cluster, and then adopt Monte Carlo simulation to predict the cluster over-demand probability.

We separately exploit the common and opportunistic contextual factors to estimate the bike rental and return number of a cluster. Specifically, we first estimate the *base* bike rental and return number of the cluster leveraging historical records having the same common contextual factors. We then infer an *inflation rate* [35] to quantitatively measure the impact of the nearby social and traffic events on the cluster. Finally, we multiply the base bike rental and return number by the inflation rate to obtain the final estimation value the cluster.

With the estimated bike rental and return number and the current station status of a cluster, we adopt Monte Carlo simulation [36] to predict the over-demand probability for each cluster. Specifically, we first model the bike rental and return events in the prediction time window as a Poisson process [37] parameterized by the predicted bike rental and return number. We then generate two stochastic sequences [38] of bike rental and return events based on the corresponding distributions. We simulate the bike rental and return process in the cluster by randomly dispatching the events to available stations in the cluster in chronological order, until a station over-demand occurs (i.e., the station stays full or empty for more than 10 minutes) or both sequences are traversed. We repeat the simulation for Γ times (e.g., 10,000 times), and use a discrimination threshold to classify over-demand clusters.

Bike Rental and Return Number Estimation

First, we estimate the base bike rental and return number of a cluster using the cluster's average value in historical records having the same common contextual factors. Note that we deliberately remove records with social or traffic events in the cluster, since in these records, the bike rental and return number caused by opportunistic events are mixed with the ones related to the common contextual factors.

Then, we model the inflation rate at the event type level. We assume that under the same common context, the same type of events have similar inflation rates on the nearby clusters. Here we define an event as being near a cluster if the geographic distance of the event and the cluster center is within the neighborhood threshold τ. Specifically, under a common

context $\Psi_c(t)$, we denote the inflation rate of event type i as θ_i ($i = 1, \ldots, 10$ corresponding to the types in Table 2). For cluster C, the overall inflation rate is then $\sum_{i=1}^{I} n_i\, \theta_i$, where n_i is the number of events of type i observed near the cluster. In order to infer each θ_i, we select historical records of cluster C with events under the same common contexts $\Psi_c(t)$, and calculate the overall inflation rate in each record by dividing the bike rental and return number by the base number of the cluster (which is calculated in the previous step). We collect the corresponding event number and the overall inflation rate from all clusters, and train a linear regression [39] model to infer each θ_i. With the learned θ_i, we calculate the overall inflation rate for cluster C.

Finally, we multiply the base bike rental (return) number by the overall inflation rate to obtain the final prediction of the bike rental (return) number for each cluster.

Over-Demand Probability Prediction

Given the predicted bike rental and return number in a cluster, we adopt a Monte Carlo method to simulate the bike rental and return process in the cluster. According to [40], the number of bikes rented or returned in the predicted time window follows a Poisson distribution. We take the bike return number as an example to elaborate on the details. Given a cluster C with the predicted bike return number N_C in the time window $[t, t+\Delta_t]$ (e.g., one hour), we divide Δ_t into T small consecutive time intervals $\delta_t = \Delta_t / T$ (e.g., one minute). The number of bikes returned to this cluster k in δ_t follows a Poisson distribution with mean parameter $\lambda = N_c/T$:

$$p(k|\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}, \quad k = 0, 1, 2, \ldots \tag{11}$$

We then generate a stochastic sequence $Q_i^+ = [k_1, \ldots, k_T]$ from the distribution to simulate the bike return events in the cluster. Similarly, we generate a stochastic sequence Q_i^- based on the bike rental distribution for the bike rental events.

Afterward, we randomly dispatch the bike return and rental events from both sequences to any available stations in chronological order[4]. If a station is observed to be full or empty for more than 10 minutes, we mark the cluster as an over-demand cluster and stop the process. Otherwise we traverse the sequences and output the cluster as a normal cluster in the given time window. We note that if we define the over-demand cluster as 'containing at least K over-demand member stations', our method can directly adapt to the new definition by observing K over-demand stations in the cluster before marking the cluster as being an over-demand cluster.

We repeat the simulation for Γ times to count the over-demand occurrences γ, and estimate the over-demand probability of the cluster as the rate $p = \gamma/\Gamma$. We use a *discrimination threshold* ϵ to classify a cluster as an over-demand cluster if $p \geq \epsilon$.

[4]In reality, users might have preferences on specific stations, while such preferences are not always significant and consistent within a small cluster based on our observations on the dataset. We plan to model user preferences in our future work.

Table 3. Summary of Datasets

Data type	Item	New York City	Washington, D.C
Bike sharing	# Stations	327	203
	# Bike trips	18,019,196	6,138,428
	# Station status	hourly	hourly
	# Over-demand	626,856	318,576
Contextual factors	# Weather forecast	hourly	hourly
	# Social events	435	329
	# Traffic events	958	745
Data collection period		01/01/2014–12/31/2015	

EVALUATION

Experiment Settings

Datasets

We evaluate our framework in New York City and Washington, D.C., respectively. We collect bike sharing data and context data for two years (01/01/2014–12/31/2015), as presented in Table 3. The data processing details are as follows.

- *Bike sharing data*: we collect two years' bike trip historical records from the data portals of NYC Citi Bike [21] and DC Capital Bikeshare [41], respectively. The data format of each trip record is: *(rental station, rental time, return station, return time)*. Based on the records, we count the bike rental number and bike return number in each hour for each station, respectively. We also collect the hourly *station status* data from the Citi Bike station feed [21] and the Capital Bikeshare station feed [41], respectively, to obtain the number of available bikes and docks in each station at the beginning of each hour.
- *Meteorological data*: we retrieve the hourly weather forecast data for both cities from the Weather Underground API [22], and parse the weather condition and air temperature value for each hour based on the data.
- *Social event data*: we compile a list of social events from the Eventful API [25] in the two years for both cities. We select events based on the types defined in Table 2. Each social event record contains the following fields: *(name, type, time, location)*.
- *Traffic event data*: we retrieve the traffic events of NYC from the NYC 511 traffic feed and the NYCT Subway Twitter account, and the traffic events of DC from the DC Police Traffic Twitter account [42]. We process these data records and filter relevant traffic events based on Table 2.

  We collect the ground truth of over-demand clusters as follows: at the beginning of the hour, we obtain the current numbers of available bikes and docks in each station of a cluster from the station feeds, and then update the status of each station based on the bike rental and return data during the hour. As soon as we observe a station staying full or empty for more than 10 minutes, we mark the enclosing cluster as an over-demand cluster. Otherwise, we mark the cluster as normal in the hour. In this way, we obtain 626,856 and 318,576 over-demand events in NYC and DC during the two years, respectively. These over-demand events usually occur in stations near transportation hubs during rush hours, and stations near parks during weekend daytime.

Table 4. The contingency table with an example

Prediction \ Truth	Over-demand clusters	Normal clusters
Over-demand clusters	True Positive (TP) 11	False Positive (FP) 2
Normal clusters	False Negative (FN) 1	True Negative (TN) 54

Evaluation Plan

We use the data of 2014 as the training set to learn the relationship between bike usage patterns and contextual factors, and use the data of 2015 for evaluation. We perform a prediction every hour during the active hours of a day. For each prediction, we first obtain the context of the corresponding time window, including the temporal group, the weather and temperature forecast, the social events starting/happening/ending in the next hour, and the traffic events occurred in the previous hour. We then dynamically cluster stations according to the context, and predict the over-demand clusters for the corresponding time window.

Evaluation Metrics

We compare the over-demand prediction of each cluster to the ground truth, and organize the results according to Table 4. For example, Table 4 shows a clustering scheme with 68 clusters, among which 12 clusters are over-demand, and the proposed method successfully predicts 11 of them. We define the following metrics to evaluate the prediction accuracy [43]:

$$precision = \frac{|TP|}{|TP| + |FP|}, \quad recall = \frac{|TP|}{|TP| + |FN|} \tag{12}$$

$$F1\text{-}Score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{13}$$

To further evaluate the prediction performance, we draw the ROC Curve [44] by plotting the true positive rate ($\frac{|TP|}{|TP|+|FN|}$) against the false positive rate ($\frac{|FP|}{|FP|+|TN|}$) under various discrimination threshold settings. We compute the AUC (Area Under ROC Curve) [44] values as another metric to evaluate the prediction performance.

Baseline Methods

We name our method WCN-MC (Weighted Correlation Network and Monte Carlo simulation), and compare our method with two sets of baselines, i.e., the *station-level* and the *cluster-cluster* prediction methods. In particular, we design three station-level baselines:

- **ARIMA**: this baseline method models the number of available bikes (docks) in a station as a time series, and uses an auto-regressive integrated moving average (ARIMA) model [9] to predict the station status in the future. It then detects the occurrence of over-demand stations based on the predicted station status.
- **B-MC**: this baseline method uses a Bayesian network to model and predict the bike rental and return number of each station leveraging station status and the context features [5]. It then directly applies the Monte Carlo simulation method on each single station for over-demand prediction.
- **ANN-S**: this baseline method directly models the over-demand probability with regard to the current station status and the context features by leveraging an Artificial Neural Network (ANN) model.

To make a fair comparison with our method, for each of these station-level baselines, we further infer its cluster-level prediction by clustering the stations in the same way as our method. We also design three cluster-level baselines as follows:

- **SC-MC**: the Static Clustering (SC) baseline method uses the clustering approach proposed by [16] to group stations into static clusters based on the geographic distance and the bike usage patterns of stations in all contexts . It then uses the same Monte Carlo method as in WCN-MC to predict over-demand clusters.
- **CCF-MC**: the Common Contextual Factor-based Clustering (CCF) method does not consider the opportunistic contextual factors and use a template-based method in station clustering. It then applies the same Monte Carlo method as in WCN-MC to predict over-demand clusters.
- **ANN-C**: this baseline method uses the same clustering results from our method, and then directly predicts cluster over-demand probability based on the status of stations in the cluster and the context features using an ANN model. We design this method to verify the effectiveness of our Monte Carlo-based method.

Evaluation Results

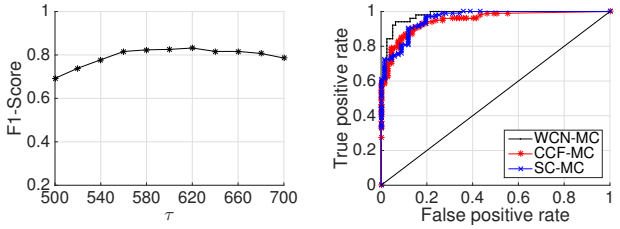
We first present the overall prediction results in both cities, and then study the impact of two parameters (neighborhood threshold τ and discrimination threshold ϵ) on the NYC results, while the results of DC are similar.

Overall Prediction Results

We compare the over-demand prediction results of different methods in Table 5. Our WCN-MC method achieves 0.882 precision and 0.938 recall in NYC, and 0.857 precision and 0.923 recall in DC, outperforming all the baseline methods. In general, the cluster-level methods achieve higher accuracy than the station-level methods. In particular, among the station-level methods, the context-aware method B-MC achieves significantly better results than the time series-based method ARIMA, which justifies the necessity of incorporating context information in over-demand prediction. Among the cluster-level methods, CCF-MC outperforms SC-MC by involving the common contextual factors in the clustering phase. Our WCN-MC method further improves the performance upon CCF-MC by considering not only the common contextual factors but also the opportunistic contextual factors. We also note that the ANN-S and ANN-C methods do not achieve best results in the corresponding station-level and cluster-level baseline groups, indicating that directly exploiting context features to model the over-demand probability does not achieve consistent improvement in prediction accuracy. In contract, our method separately models the impact of the common and opportunistic contextual factors and consistently achieves high over-demand prediction accuracy.

Table 5. Over-demand prediction results of different methods

Methods	Precision	Recall	F1	Precision	Recall	F1
	NYC			DC		
ARIMA	0.548	0.506	0.526	0.520	0.541	0.530
B-MC	0.753	0.656	0.692	0.636	0.539	0.583
ANN-S	0.776	0.571	0.658	0.667	0.428	0.521
SC-MC	0.790	0.647	0.711	0.793	0.821	0.807
CCF-MC	0.833	0.832	0.828	0.815	0.880	0.846
ANN-C	0.673	0.852	0.752	0.857	0.600	0.706
WCN-MC	**0.882**	**0.938**	**0.909**	**0.857**	**0.923**	**0.889**



(a) Neighborhood threshold (b) ROC curves

Figure 7. Parameter impact analysis.

Parameter Impact Study

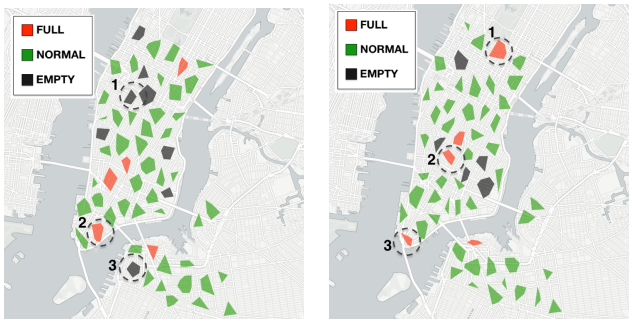
We examine the impact of the neighborhood threshold τ on the prediction performance. Based on bike sharing system operation reports [3, 7], we vary the threshold τ from 500m to 700m, corresponding to the common walking distance range of users. Figure 7(a) shows the F1-Score under different τ values. We can see that setting a small neighborhood threshold leads to relatively lower accuracy, probably because the resulting clusters might be too small to exhibit consistent bike usage pattern. On the other hand, a large cluster might not be practically useful for operators. Therefore, we set $\tau = 620m$ in our experiments, and obtain an average of 67.08 clusters out of 327 stations. Each cluster contains an average of 4.74 stations with an average geographic span of 613.40m. Based on this setting, we then determine the optimal influence degree $\mu = 0.53$ which maximizes the F1-Score.

We also study the prediction performance under different discrimination thresholds by varying the values of ϵ from 0 to 1. Figure 7(b) shows the ROC curve of our WCN-MC method as well as the two cluster-level baselines CCF-MC and SC-MC. Our method achieves an AUC of 0.97, which is higher than the two baselines (0.93 for CCF-MC and 0.89 for SC-MC, respectively). Based on the ROC plot, we select $\epsilon = 0.71$ as the optimal discrimination threshold in our experiments.

Case Studies

Weekday Rush Hours

Figure 8(a) shows the dynamic clustering and over-demand prediction results during the morning rush hours of a typical weekday (8:00–9:00, 06/07/2015), where the red/green/black colors encode full/normal/empty cluster status, respectively. We observe several clusters near major transportation hubs and business/residential districts, such as the Penn Station area (Circle 1), the Wall Street area (Circle 2), and the Brooklyn Heights area (Circle 3). During rush hours, these clusters



(a) Weekday morning rush hours (b) Sunny spring weekend

Figure 8. Dynamic clustering and prediction results in different temporal and weather contexts.

are usually full or empty, revealing the underlying dynamics and directions of the commuting flow. With such knowledge, bike sharing system operators could take preventive actions to ensure the station availability, such as sending trucks to redistribute bikes among these areas before rush hours.

Weather Condition and Air Temperature

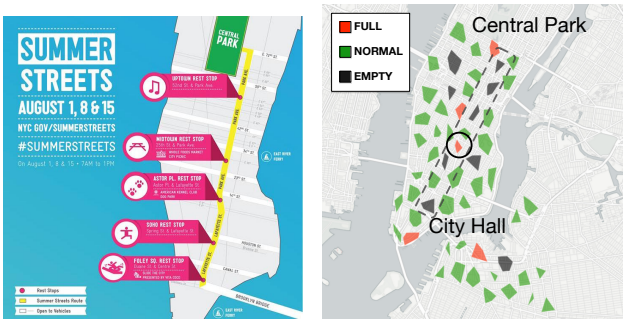
We present the result of a sunny spring weekend afternoon (14:00–15:00, 05/24/2015) in Figure 8(b). We observe several full clusters near the major parks of NYC, such as Central Park (Circle 1), Union Square Park (Circle 2), and Battery Park (Circle 3). A possible explanation is that people like to ride bikes to parks to enjoy outdoor activities in the springtime [45]. With such knowledge, bike sharing system operators can provide more pleasant weekend riding experience by, for example, setting up temporary bike corrals around these parks to ensure that there are sufficient docks.

Social Events

We study the case of the city festival Summer Streets [46] in 2015. Summer Streets is a celebration of NYC's streets on three Saturdays in August (we present the results of 12:00–13:00, 08/08/2015), featuring bike tours, block parties, and street arts along Park Avenue from Central Park to New York City Hall (Figure 9(a)). Taking the event information into account, our dynamic clustering and prediction method successfully identifies several empty clusters along Park Avenue near Central Park and City Hall, as highlighted in Figure 9(b). Interestingly, we notice a full cluster near Union Square (the circle in Figure 9(b)). We examine the events and find the *Union Square Greenmarket* [47] is being held in the park. The greenmarket features foods and cooking demonstrations, which might attract large crowds of riders to stop for a rest. With the prediction, operators can adjust bike redistributing plans in Park Avenue before the festival and set up temporary bike corrals near Union Square.

Running Time Analysis

We evaluate the runtime efficiency of our approach on a 64-bit server with an quad-core 3.20GHz CPU and 32GB RAM. We find that the prediction accuracy regarding F1-Score does not increase significantly when the Monte Carlo simulation times Γ exceeds 8,000. Therefore, we set $\Gamma = 8,000$ in each prediction cycle, and present the detailed processing time in Table 6. The average time for running a prediction is about



(a) Summer Streets poster (b) Over-demand clusters

Figure 9. Clustering and prediction results of NYC Summer Streets.

Table 6. Running time analysis

Procedures	Time (ms)	
	NYC	DC
GCLP clustering	863	532
Bike usage number prediction	701	428
Monte Carlo simulation	8,523	5,349
Total	10,087	6,309

10 seconds for NYC Citi Bike system and about 6 seconds for DC Capital Bikeshare system, respectively.

RELATED WORK

Recently, bike sharing systems have been intensively studied from different perspectives, including bike sharing history [48], infrastructure [49], worldwide deployment [1, 50], and bike usage patterns [51, 4, 40, 9]. The research interests mainly focus on the following problems: (1) *system planning*, such as determining the number, capacity and locations of stations [52, 53, 54]. (2) *system balancing*, such as strategies to transport bikes among stations [8], and mechanisms to encourage users to rent bikes from (or return bikes to) specific stations through incentives [2, 55]. (3) *system prediction*, such as predicting station status and bike usage number using different models. Since our work is related to bike sharing system prediction, we review the existing work from two aspects, i.e., station-level and cluster-level prediction.

The earlier work mainly focuses on predicting the number of available bikes and docks (i.e. station status) in the station level. For example, Froehlich et al. [5] adopted a Bayesian network to predict station status based on the current time and current bike/dock number. Kaltenbrunner et al. [9] proposed to model and predict the station status as a time series using an ARIMA model. However, due to the impact between neighboring stations [10] and the complicated contextual factors impacting bike usage (e.g., weather, temperature, social events) [13, 7, 52], these station-level prediction methods do not consistently achieve accurate results [16].

To address this issue, researchers have proposed to cluster similar stations into clusters, and then predict bike usage on a cluster-level. For example, Li et al. [16] first proposed a method to cluster stations based on their geographical locations and transition patterns, then predicted the bike usage of the whole system, and finally allocated the overall bike rental and return number to each cluster based on a proportion learned from historical data. However, the cluster scheme is static across different contexts. Etienne et al. [40] introduced a model-based method to group stations with similar bike usage patterns, such as stations near restaurants and train stations, and predicted their bike usage pattern in different temporal settings. These cluster-level prediction methods could improve the prediction accuracy, however the clusters used in these methods are static regardless of context at the time. Since the bike usage patterns of stations might be affected by various contextual factors such as weather condition and social events [13, 7, 14], the prediction results of static clusters may not yield consistent accuracy across different contexts.

In this work, we use a weighted correlated network [29, 20] to model the relationship among bike stations in dynamic contexts. Weighted correlated networks have been used to model social networks [56], biological networks [57, 58], transportation networks [59], etc. The clusters can then be regarded as small communities in the network, which can be found using various algorithms such as Label Propagation [31], Hierarchical Clustering [60] and the Girvan-Newman algorithm [32]. In this paper, we use the greedy algorithm Label Propagation as it can identify communities in nearly linear time by iteratively assigning nodes to highly connected clusters[31]. However, the original algorithm does not constrain the size of clusters and might result in very large communities which are not practically useful in our scenario. Ciglan et al. [34] proposed a size-constrained Label Propagation algorithm *SizConCD* to constrain the number of nodes in a cluster. However, SizConCD still cannot be directly used in our work as we need to constrain the geographic span of a cluster instead of the number of nodes in the cluster. Therefore, we proposed the Geographic-Constrained Label Propagation algorithm to solve our clustering problem.

CONCLUSION

In this paper, we propose a dynamic cluster-based framework to predict over-demand occurrence in bike sharing systems according to the varied and highly dynamic contexts. To effectively model the relationship among bike stations, we consider two sets of contextual factors, i.e., the common contextual factors including time, weather, and temperature, and the opportunistic contextual factors including social and traffic events. We model the relationship using a weighted correlation network, and propose a geographically-constrained clustering method to group stations into clusters. Evaluation results on NYC and DC show that our framework consistently achieves high over-demand prediction accuracy in both cities across different contexts, and outperforms the start-of-the-art methods.

In the future, we intend to improve this work from the following aspects. First, we plan to better characterize the contexts with richer urban data, such as incorporating the social network check-ins. Second, we plan to explore the impacts of newly established stations and cluster size on the prediction accuracy. Third, we plan to evaluate our method on bike sharing systems in other cities with different cultural settings.

ACKNOWLEDGMENT

We would like to thank the reviewers for their constructive suggestions. Paul Gibson contributes useful comments and inputs to this paper. This research was supported by Natural Science Foundation of China (61572048), National Key Research and Development Plan (2016YFB1001200), Zhejiang Provincial Natural Science Foundation of China (LR15F020001), Program for New Century Excellent Talents in University (NCET-13-0521), and the Swiss National Science Foundation (PP00P2_153023). The corresponding author is Gang Pan. This work was done when Longbiao Chen was working in Institut Mines-Télécom; CNRS, France.

REFERENCES

1. S. Shaheen, S. Guzman, and H. Zhang, "Bikesharing in Europe, the Americas, and Asia," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2143, no. 1, pp. 159–167, 2010.
2. A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing Users for Balancing Bike Sharing Systems," in *Proc AAAI*.
3. A. M. Burden and R. Barth, *Bike-Share Opportunities in New York City*. New York: Department of City Planning, 2009.
4. X. Zhou, "Understanding Spatiotemporal Patterns of Biking Behavior by Analyzing Massive Bike Sharing Data in Chicago," *PLoS ONE*, vol. 10, no. 10, p. e0137922, 2015.
5. J. Froehlich, J. Neumann, and N. Oliver, "Sensing and Predicting the Pulse of the City through Shared Bicycling." in *Proc. IJCAI*, vol. 9, pp. 1420–1426.
6. S. M. Kaufman, L. Gordon-Koven, N. Levenson, and M. L. Moss, "Citi Bike: The First Two Years," 2015.
7. LDA Consulting, *2013 Capital Bikeshare Member Survey Report*, Washington, D.C., 2013.
8. C. Contardo, C. Morency, and L.-M. Rousseau, *Balancing a dynamic public bike-sharing system*. CIRRELT, 2012, vol. 4.
9. A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, "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, 2010.
10. J. W. Yoon, F. Pinelli, and F. Calabrese, "Cityride: A Predictive Bike Sharing Journey Advisor," in *Proc MDM*, pp. 306–311.
11. A. K. Dey, "Understanding and Using Context," *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001.
12. G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*. Springer Berlin Heidelberg, Sep. 1999, no. 1707, pp. 304–307.
13. K. Gebhart and R. B. Noland, "The impact of weather conditions on bikeshare trips in Washington, DC," *Transportation*, vol. 41, no. 6, pp. 1205–1225, 2014.
14. L. Chen, D. Yang, J. Jakubowicz, G. Pan, D. Zhang, and S. Li, "Sensing the Pulse of Urban Activity Centers Leveraging Bike Sharing Open Data," in *Proceedings of the 12th IEEE International Conference on Ubiquitous Intelligence and Computing*.
15. P. Midgley, "The role of smart bike-sharing systems in urban mobility," *Journeys*, vol. 2, pp. 23–31, 2009.
16. Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic Prediction in a Bike-Sharing System," in *Proc. SIGSPATIAL'15*.
17. Y. Zheng, "Methodologies for Cross-Domain Data Fusion: An Overview," *IEEE Transactions on Big Data*, vol. 1, no. 1, pp. 16–34, 2015.
18. D. Zhang, B. Guo, and Z. Yu, "The emergence of social and community intelligence," *Computer*, vol. 44, no. 7, pp. 21–28, 2011.
19. D. Zhang, B. Guo, B. Li, and Z. Yu, "Extracting Social and Community Intelligence from Digital Footprints: An Emerging Research Area," in *Proc. UIC'10*, 2010, pp. 4–18.
20. P. Langfelder and S. Horvath, "WGCNA: an R package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, p. 559, 2008.
21. Citi Bike, "Citi Bike System Data," 2016. http://www.citibikenyc.com/system-data
22. Weather Underground Inc, "Weather API," 2016. https://www.wunderground.com/weather/api/
23. J. Dill and K. Voros, "Factors Affecting Bicycling Demand: Initial Survey Findings from the Portland, Oregon, Region," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2031, pp. 9–17, 2007.
24. F. Zhang, D. Wilkie, Y. Zheng, and X. Xie, "Sensing the Pulse of Urban Refueling Behavior," in *Proceedings of the 15th ACM International Conference on Ubiquitous Computing*, pp. 13–22.
25. Eventful, Inc, "Events Feed," 2016. https://api.eventful.com/
26. New York State, "511ny Data Feed," 2016. https://511ny.org/developers/resources
27. Twitter, "New York City Subway," 2016. https://twitter.com/NYCTSubway
28. J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules," *Science*, vol. 302, no. 5643, pp. 249–255, 2003.
29. B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical applications in genetics and molecular biology*, vol. 4, no. 1, 2005.

30. M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, pp. 1082–1097, 2009.

31. U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

32. M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, 2004.

33. J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge, 2013.

34. M. Ciglan and K. Nørvåg, "Fast Detection of Size-Constrained Communities in Large Networks," in *Web Information Systems Engineering – WISE 2010*. Springer Berlin Heidelberg, Dec. 2010, no. 6488, pp. 91–104.

35. Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting Fine-Grained Air Quality Based on Big Data," in *Proceedings of the 21th SIGKDD conference on Knowledge Discovery and Data Mining*.

36. C. Z. Mooney, *Monte Carlo Simulation*. SAGE Publications, Apr. 1997.

37. A. E. Raftery and V. E. Akman, "Bayesian Analysis of a Poisson Process with a Change-Point," *Biometrika*, vol. 73, no. 1, pp. 85–89, 1986.

38. B. D. Ripley, *Stochastic Simulation*. John Wiley & Sons, 2009.

39. H. F. Senter, "Applied linear statistical models," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 880–880, 2008.

40. C. Etienne and O. Latifa, "Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the VÉLib' System of Paris," *Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, pp. 39:1–39:21, 2014.

41. Capital Bikeshare, "Capital Bikeshare System Data," 2016. http://www.capitalbikeshare.com/system-data

42. Twitter, "DC Police Traffic," 2016. https://twitter.com/DCPoliceTraffic

43. D. M. Powers, "Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation," 2011.

44. J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

45. L. Gordon-Koven and N. Levenson, "Citi Bike Takes New York," *Rudin Center for Transportation Management and Policy*, 2014.

46. The City of New York, "Summer Streets," 2016. http://nyc.gov/summerstreets

47. Weather Underground Inc, "Union Square Saturday Greenmarket," 2016. http://www.grownyc.org/greenmarket/manhattan-union-square-sa

48. P. DeMaio, "Bike-sharing: History, impacts, models of provision, and future," *Journal of Public Transportation*, vol. 12, no. 4, pp. 41–56, 2009.

49. J. Pucher, J. Dill, and S. Handy, "Infrastructure, programs, and policies to increase bicycling: An international review," *Preventive Medicine*, vol. 50, pp. 106–125, 2010.

50. S. Shaheen, H. Zhang, E. Martin, and S. Guzman, "China's Hangzhou public bicycle," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2247, no. 5, pp. 33–41, 2011.

51. P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns," *Procedia - Social and Behavioral Sciences*, vol. 20, pp. 514–523, 2011.

52. J. C. Garcia-Palomares, J. Gutierrez, and M. Latorre, "Optimizing the location of stations in bike-sharing programs: A GIS approach," *Applied Geography*, vol. 35, no. 1–2, pp. 235–246, 2012.

53. J.-R. Lin and T.-H. Yang, "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, 2011.

54. L. Chen, D. Zhang, G. Pan, X. Ma, D. Yang, K. Kushlev, W. Zhang, and S. Li, "Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data," in *Proc. UbiComp*, pp. 571–575.

55. P. Aeschbach, X. Zhang, A. Georghiou, and J. Lygeros, "Balancing Bike Sharing Systems through Customer Cooperation–A Case Study on London's Barclays Cycle Hire," *In Proc. IEEE CDC*, 2015.

56. Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory." *IEEE Data Engineering Bulletin*, vol. 33, no. 2, pp. 32–39, 2010.

57. A.-L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

58. Z. Wang, D. Zhang, X. Zhou, D. Yang, Z. Yu, and Z. Yu, "Discovering and Profiling Overlapping Communities in Location-Based Social Networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 4, pp. 499–509, Apr. 2014.

59. P. Kaluza, A. Kölzsch, M. T. Gastner, and B. Blasius, "The complex network of global cargo ship movements," *J. R. Soc. Interface*, vol. 7, no. 48, pp. 1093–1103, 2010.

60. G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007.