Inferring bike trip patterns from bike sharing system open data

Conference Paper · October 2015

DOI: 10.1109/BigData.2015.7364115

2 authors:

Longbiao Chen
Xiamen University
24 PUBLICATIONS **157** CITATIONS
SEE PROFILE

Jeremie Jakubowicz
Institut Mines-Télécom
57 PUBLICATIONS **1,327** CITATIONS
SEE PROFILE

Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Understanding Bike Trip Patterns Leveraging Bike Sharing System Open Data

Longbiao CHEN[1,2,3], Xiaojuan MA[4], Thi-Mai-Trang NGUYEN[2], Gang PAN[3], Jérémie JAKUBOWICZ(✉)[1]

1 Institut Mines-Télécom; Télécom SudParis; UMR CNRS SAMOVAR, Evry 91000, France
2 LIP6, University of Paris 6, Paris 75005, France
3 College of Computer Science, Zhejiang University, Hangzhou, 310027, China
4 Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong

Abstract Bike sharing systems are booming globally as a green and flexible transportation mode, but the flexibility also brings difficulties in keeping the bike stations balanced with enough bikes and docks. Understanding the spatio-temporal bike trip patterns in a bike sharing system, such as the popular trip origins and destinations during rush hours, is important for researchers to design models for bike scheduling and station management. However, due to privacy and operational concerns, bike trip data are usually not publicly available in many cities. Instead, the station feeds about real-time bike and dock number in stations are usually public, which we refer to as bike sharing system open data. In this paper, we propose an approach to infer the spatio-temporal bike trip patterns from the public station feeds. Since the number of possible trips (i.e., origin-destination station pairs) is much larger than the number of stations, we define the trip inference as an ill-posed inverse problem. To solve this problem, we identify the sparsity and locality properties of bike trip patterns, and propose a sparse and weighted regularization model to impose both properties in the solution. We evaluate our method using real-world data from Washington, D.C. and New York City. Results show that our method can effectively infer the spatio-temporal bike trip patterns and outperforms the baselines in both cities.

Keywords bike sharing system, open data, ill-posed inverse problems, urban computing

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *jeremie.jakubowicz@telecom-sudparis.eu*

1 Introduction

Many cities have deployed bike sharing systems to promote a greener transportation mode and a healthier life style. Riding public bikes for commuting, traveling, and exercising has become increasingly popular among citizens [1,2]. For example, in 2014, riders in New York City and Washington, D.C. went on 8,081,188 and 1,869,980 bike trips, respectively. However, since riders can flexibly pick up and drop off a bike at different bike stations, these stations can become unbalanced and even unavailable (e.g., being full or empty) in popular trip origins and destinations. The users' experience may be greatly impaired if they run into an unavailable station, which may ultimately hinder the user participation of bike sharing systems. Understanding the spatio-temporal patterns [3] of public bike trips, such as the popular origins, destinations, and trajectories during rush hours or social events, can help researchers and urban planners develop better bike re-balancing strategies [4], optimize bike station placement [5], and help urban authorities manage human flows during events [6–8].

However, Due to privacy and operational concerns, the bike trip data about trips from one station to another are usually not publicly available. Without such information, researchers have to resort to bike user surveys or probabilistic simulations for modeling bike trip patterns. For example, Garcia-Palomares et al. [9] estimated the number of bike trips in each transport zone based on annual people mobility

surveys, while Contardo et al. [10] simulated bike trips using probability distributions to evaluate their bike scheduling model. Due to the fact that riders are free to pick up or drop off bikes at any stations during the operation hours without a reservation, real-world bike trip patterns can be quite asymmetric in space and fluctuating throughout the day [11]. Survey or simulation-based methods often fail to capture these details, inevitably introducing risks to decision making based on their bike trip estimation results.

Fortunately, most bike sharing systems publish real-time *station feeds*, i.e., the number of bikes and docks available in each bike station, together with its location and maximum capacity of the station, to help riders find nearby available stations. Although these data do not carry information about individual bike trip, we can potentially infer the spatio-temporal bike trip patterns from the variations of bikes and docks in stations. More specifically, in a time window (e.g., one hour), we first derive the *bike traffic* of each station, i.e., the number of bikes departing from or arriving at the station, by continuously monitoring the up-to-date station feeds in the hour. We then model the traffic of a station as the sum of several *bike flows* between the station and other stations. Finally, we design an algorithm to infer the bike flows from the station traffic, which gives us a panoramic view of the spatio-temporal bike trip patterns in the system.

However, since the number of origin-destination station pairs are much larger than the number of stations, inferring bike flows from station traffic becomes an *ill-posed inverse problem* [12]. As illustrated in Figure 1, for a bike sharing system with n stations, we can derive n incoming traffic and n outgoing traffic in each time window. However, we need to infer n^2 station-to-station bike flows. Since $2n \ll n^2$ for most bike sharing systems, such a problem is *ill-posed*, yielding many possible solutions [12]. For example, Figure 1 shows two different bike flow settings that can be inferred from the same station traffic data.

Ill-posed inverse problems have been investigated for decades [12, 13], and it has been proved that the main challenge can be properly tackled if we have enough *a priori* information to constrain the solution in a lower dimensional space than its original formulation [12]. This *a priori* information is usually injected through *regularization* [13]. In order to obtain the *a priori* information in the bike flow inference problem, we conducted an empirical study on a sample bike trip history dataset. We identify two important bike flow properties, namely: (1) *sparsity of strong flows*, meaning that the strong bike flows (e.g., more than δ trips per hour) only occur in 0.62% of all the station pairs citywide,

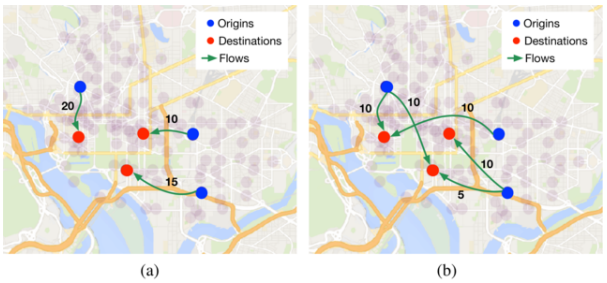


Fig. 1 An illustrative example of two different bike flow patterns inferred from the same station traffic data. Blue dots and red dots represent origin and destination stations, respectively, and the number over each green arrow indicates the bike flow intensity.

and (2) *locality of strong flows*, meaning that 90% of the strong bike flows are observed in station pairs within a distance of 2 km. We impose the *sparsity* property into the solution by leveraging the ℓ_1 regularization technique [14] to constrain the number of nonzero flows, and incorporate the *locality* property by adding a *weighted* ℓ_2 regularization term to penalize station pairs with geographically-distant components. Finally, we identify our problem as convex and practically solvable using convex optimization methods [15]. The contributions of this work include:

1. To the best of our knowledge, this is the first work on spatio-temporal bike trip patterns inference from bike sharing system station feeds.
2. We formulate the station traffic to bike flow inference as an ill-posed inverse problem, and propose a *Sparse and Weighted Regularization (SWR)* method to constrain the solution space in a solvable dimension by applying the sparsity and locality properties of bike flows identified from a sample bike trip dataset.
3. We evaluate our method using real-world bike sharing system open data from Washington, D.C. and New York City. Results show that our method effectively infers the bike trip patterns in both cities, and outperforms two baselines significantly.

The remainder of this paper is organized as follows. We first review the related work in Section 2, and then present our empirical study on public bike trip properties in Section 3. We formulate the bike flow inference problem in Section 4, and elaborate on the details of the proposed SWR method in Section 5. We report the evaluation results in Section 6. Finally we conclude the paper in Section 7.

2 Related Work

In this section, we first survey existing work related to understanding bike trip patterns from bike sharing system open data, and then present existing methods to solve ill-posed inverse problems.

2.1 Understanding Bike Trip Pattern

Bike sharing systems have been deployed in many cities to improve urban sustainability and promote a healthier lifestyle. Meanwhile, a large volume of bike sharing system data are generated, providing invaluable resource for researchers to design new applications that can further improve the bike system management and planning [16]. For example, Froehlich et al. [17] collected bike sharing station data from Barcelona, Spain to study the station-level bike usage patterns (e.g., number of bikes in a station during different period of time), and predict the station availability in the future. Chen [5] proposed a framework to determine optimal bike station placement using bike sharing data and other urban open data. Zhao et al. [18] proposed a bike usage prediction system to help riders find potentially available bikes in rush hours. However, prior research did not capture the spatio-temporal bike trip patterns due to the lack of bike trip information. In fact, the bike trip data are usually not publicly available due to privacy concerns and extra operational costs in many cities. Although some cities such as Washington, D.C. provide bike trip data to the public, the data is published with a delayed of months, hindering the real-world application of research work relying on such data.

To address this issue, some work [9] leveraged human mobility survey results to model bike trip demands, while other attempted [10] used probabilistic simulations to design strategies for bike balancing operations (i.e., transporting bikes from full stations to empty stations). However, conducting surveys is usually costly in time, money, and human resources, and the simulation data might not correspond to the highly dynamic real-world bike usage patterns. Randriamanamihaga et al. [19] conducted a clustering analysis of the bike flows in the Paris Vélib system, however the data is from a private company and not made public. Our work aims to infer the spatio-temporal patterns of public bike trips directly from the publicly available station feeds, which has not yet been explored and will benefit the above-mentioned applications, including but not limited to bike traffic prediction and station balancing.

2.2 Solving Ill-Posed Inverse Problems

An inverse problem refers to the process of extracting the causal factors from a set of observations entailed [20]. Such problems are often *ill-posed*, i.e., when the number of factors are larger than the number of observations [12], there might be potentially infinite number of solutions. However, by incorporating *a priori* information into the problem formulation using *regularization*, we can constrain the solution in a lower dimensional space and obtain a desirable solution with a high probability [13].

Compressive sensing [21] techniques have been widely exploited to solve inverse problems [20, 22]. The rationale behind compressive sensing is that the solutions to the inverse problems are usually sparse in nature. For example, Wang et al. [23] proposed a compressive crowdsensing scheme to adaptively recover the city-wide air quality map from a limited number of sensing data, and Chawla et al. [24] defined the causality inference of traffic anomalies as an inverse problem and solved it by applying a compressive sensing style technique. In this paper, we formulate the flow inference from traffic as an inverse problem. The traffic in each station can be regarded as the compressive sensing values of certain flows. We propose a sparse-and-weighted regularization method to exploit the sparsity and locality properties of bike flows to solve the problem.

3 Data Analysis

In this section, we elaborate how we collect and process the real-time bike sharing systems open data to derive station traffic. We also present an empirical study on a sample bike trip history dataset from Washington, D.C. to understand the common bike trip properties.

3.1 Data Collection and Processing

Most bike sharing systems provide real-time station feeds, i.e., the number of bikes and docks available in all station at query time. Users can easily search for a nearby bike station and check the availability of bikes and docks. Meanwhile, real-time data streaming APIs are also available for developers to embed the data in their applications. For example, the Capital Bikeshare system of Washington, D.C. provide a live station maps[1)] and a data streaming API[2)] for riders. In this paper, we automatically query the APIs of bike

[1)] https://secure.capitalbikeshare.com/map/
[2)] https://www.capitalbikeshare.com/data/stations/bikeStations.xml

sharing systems every one minute to obtain the station feed data. We note that some bike rental and return events might be overlooked if they occur in the same minute window. However, the statistics over a long period of time could still be quite accurate as such events are rare. For example, only 0.05% of the bike rental or return events occur in the same minute window in our sample dataset (as detailed below) during 2014 in Washington, D.C.

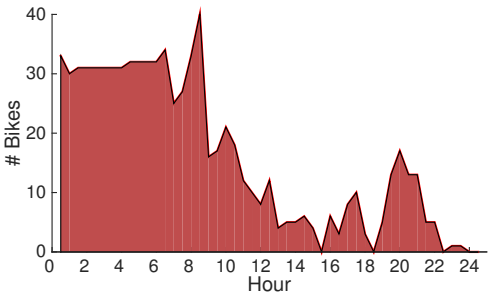
Based on the collected station feed data, we can derive the *station traffic*, i.e., the number of incoming and outgoing bikes at each station during a given period of time (e.g., one hour). More specifically, for a station s and a time interval of Δ_t minutes, we calculate the number of incoming traffic $N^+(s, \Delta_t)$ as

$$N^+(s, \Delta_t) = \sum_{t=2}^{\Delta_t} \delta_t^+ \tag{1}$$

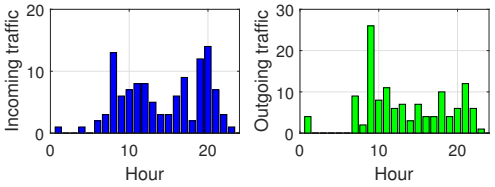
where

$$\delta_t^+ = \begin{cases} B(s,t) - B(s,t-1), & \text{if } B(s,t) > B(s,t-1) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

in which $B(s,t)$ denotes the number of available bikes in station s at time t (in minute). We calculate the outgoing traffic of station s in the same period $N^-(s, \Delta_t)$ in a similar way. For example, Figure 2 shows the number of available bikes in a station (No. 31217) during a weekday (July 9, 2014), and the derived incoming and outgoing station traffic in every hour. This bike station is located near the Dupont Circle, a hybrid area of residential neighborhoods, transit hubs (subway and bus), and business centers in Washington, D.C. We can see that the station is heavily used during the morning and evening rush hours. However, one can not directly infer any bike trip patterns from this figure, such as the common trip destinations, or the average distance traveled from this station.



(a) Station feed (number of bikes available).



(b) Station traffic (number of incoming and outgoing bikes).

Fig. 2 An illustrative example of a station feed (Station 31217) and the derived station traffic in a day (July 9, 2014).

3.2 Empirical Study on A Sample Bike Trip History Dataset

In order to understand the bike trip patterns, we conduct an empirical study on a sample bike trip history dataset from Washington, D.C., which is available on the Capital Bikeshare website[3)]. We note that even though such trip data are publicly available in some bike sharing systems, they are usually published with a delay of several months, hindering the up-to-date modeling of bike trip patterns in the system.

This sample bike trip history dataset contains 1,869,980 bike trips between 201 stations in Washington, D.C. in 2014. Each trip record contains the bike departure time and departure station, as well as the arrival time and arrival station. We aggregate these bike trips according to their origins and destinations. For each station pair, we calculate the number of trips observed over a period of time, and denote it as the *bike flow* of the station pair during that period. In summary, we obtain 40,401 station pairs. We take the flows of a typical weekday morning rush hours (7:00–9:00, average over the year 2014) as an example (Figure 3). We observe a clear flow structure between a limited number of station pairs (yellow lines in Figure 3). Since our objective in this work is to understand the spatio-temporal patterns of bike flows, we focus on recovering these flow structures. To this end, we use a threshold δ to filter *strong flows*, i.e., flows flows with more than δ bikes per hour. Based on the dataset, we observe the following two important properties of the strong flows.

3.2.1 Sparsity of Strong Flows

Based on our observation, these strong flows only exist between a limited number of station pairs. For example, in the sample dataset, only 0.62% of the 40,401 station pairs show strong flows (i.e., more than ten bike trips per hour) during weekday morning rush hours, indicating the *sparse* nature of these strong flows. We also notice that stations associated with these strong bike flows are usually located near metro stations, tourist attractions, residential

[3)] https://www.capitalbikeshare.com/trip-history-data

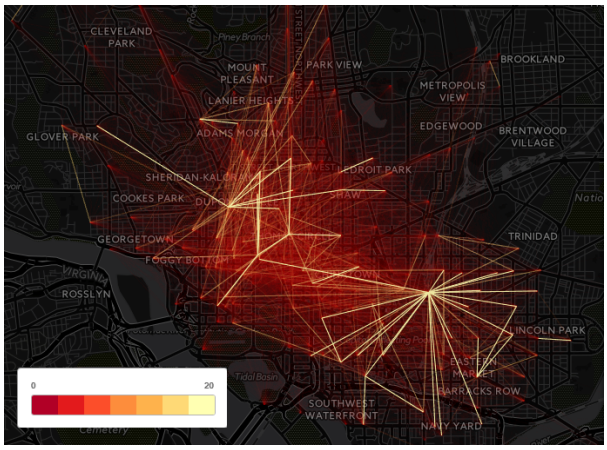


Fig. 3 An illustrative example of the bike flow patterns in weekday morning rush hours. Each link denotes a station-to-station flow, while the link color corresponds to the average flow intensity.

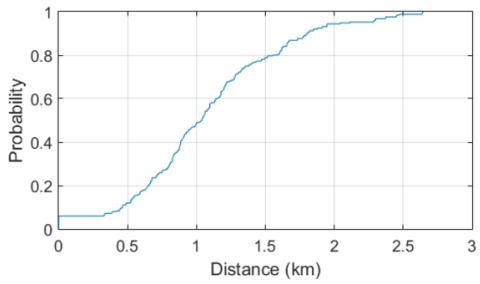


Fig. 4 CDF of the distance between station pairs with strong flows.

areas, and downtown centers. These places tend to attract larger scales of human flow, and the corresponding bike stations are designed to have a larger capacity, which might explain the causality of these strong flows. In summary, the sparse structure of these strong flows inspires us to seek a sparse solution to the flow inference problem.

3.2.2 Locality of Strong Flows

We further investigate the geographic characteristics of the station pairs with these flows. We compute the distance of these station pairs based on their coordinates, and plot the cumulative distribution function (CDF) in Figure 4. It is shown that more than 90% of these station pairs are located within a distance of 2 km. In other words, the probability of people riding public bikes between geographically-distant stations is relatively low. We note that such observation agrees well with the design purposes of bike sharing systems, such as providing bikes for short-distance trips in an urban area, and connecting users to public transit stations to solve the *last mile* problem [2, 25].

In summary, the strong flows between station pairs in a biking network are very sparse, and most bike rides travel between geographically-close stations. Based on our observations on the sample dataset, such sparsity and locality properties persist in other time windows (e.g., weekend evenings). We also assume that these properties exist in other cities, although the flow structure and distance parameter might differ. In the following sections, we explain how we incorporate such *a priori* information into the bike trip pattern inference problem.

4 Problem Formulation

4.1 Notations

We denote the set of stations by $\mathcal{V}$ and the set of directed links between any station pair $(u, v) \in \mathcal{V}^2$ by $\mathcal{L} = \mathcal{V}^2$. We then denote the cardinality of set $\mathcal{V}$ (i.e., the number of stations), and the geographic-distance between two stations by $d(u, v)$, assumed symmetrical, i.e., $d(u, v) = d(v, u)$, for our later use.

We define the *flow* f as a function from $\mathcal{L}$ to $\mathbb{R}_+$ taking nonnegative values on each link, i.e., $f : \mathcal{L} \to \mathbb{R}_+$. We think of $f(u, v)$ as the number of bikes going from station u to station v in a given time window. We then define the *incoming traffic* $g_{in} : \mathcal{V} \to \mathbb{R}_+$ and the *outgoing traffic* $g_{out} : \mathcal{V} \to \mathbb{R}_+$ as the number of bikes arriving at and departing from a station during a given time window, respectively. The couple $g = (g_{in}, g_{out})$ forms what we call a *traffic*. Based on the definitions of flow and traffic, we have

$$g_{in}(v) = \sum_{u \in \mathcal{V}} f(u, v) \tag{2}$$

$$g_{out}(v) = \sum_{u \in \mathcal{V}} f(v, u) \tag{3}$$

4.2 Problem

Given the above-mentioned definitions, we formulate our *flow inference* problem as follows.

Problem. *For a directed network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$, given the traffic g, infer the flow f.*

We note that such a problem is an *ill-posed inverse* problem, since there are only $2n$ station traffic variables, but we need to infer n^2 unknown flows between stations. Moreover, since our objective is to reveal the structure of the bike trips, we need to focus on recovering the strong flows from the traffic. In order to solve this problem, we propose

a sparse-and-weighted regularization method to exploit the sparsity and locality properties of the strong bike flows, as analyzed before.

5 Methodology

In this section, we first elaborate on the modeling of the relationship between traffic and flow in the bike trip network, and then present our flow inference method. In particular, we adopt the concepts from the *network tomography* [22] and the *compressive sensing* [21] communities to explain the rationale behind our method. More specifically, we regard the flow of the network as a high-dimensional vector, and the traffic of each node as a linear *measurement* of the flow vector that computes the sum of a specific subset of the flow entries. For example, the outgoing traffic of a node corresponds to a measurement of the flows from that node, and the incoming traffic to a measurement of the flows into the node.

Now our objective is to recover the N-dimensional flow vector from the M-measurements (traffic), where $N = n^2$ and $M = 2n$. Since $M \ll N$, this problem appears *ill-posed*. If, however, the flow vector itself is sparse, i.e., having only K non-zero entries, then the problem can be solved using a sparse regularization method provided that $M > K$. Moreover, if we impose more *a priori* information about the vector, e.g., locality, the recovered flow vector can better approximate the real-world bike trip patterns.

5.1 Modeling the Relationship between Traffic and Flow

We define the traffic and vectors as $\boldsymbol{y} = g(v), v = 1, 2, \ldots, n$ and $\boldsymbol{x} = f(u, v), u, v = 1, 2, \ldots, n$, respectively. As the traffic vector $\boldsymbol{y}$ is a measurement of the flow vector $\boldsymbol{x}$, we define an *incidence matrix* $\boldsymbol{A}$ to represent their relationship. More specifically, $\boldsymbol{A}$ is a binary matrix, having one row for each element of $\boldsymbol{y}$ and one column for each element of $\boldsymbol{x}$. The entries of the matrix are given by

$$\boldsymbol{A}_{j,i} = \begin{cases} 1, & \text{if traffic } y_j \text{ measures flow } x_i \\ 0, & \text{otherwise} \end{cases}$$

In this way, the incidence matrix $\boldsymbol{A}$ transforms the station-to-station flows into the corresponding incoming and outgoing traffic of stations. We give an example to illustrate the use of the incidence matrix in Figure 5. This bike network contains 3 nodes ($n = 3$), and thus having $n^2 = 9$ node-to-node flow entries and $2n = 6$ traffic entries, and yields a 6×9 incidence matrix. For example, the first traffic element is the inner product of the first row of the incidence matrix and the flow vector, corresponding to a measurement on the 1st, 4th, and 7th flow entries.

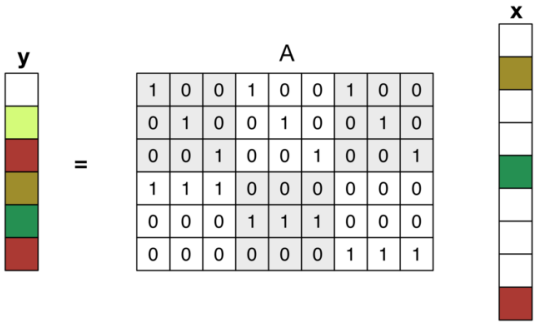


Fig. 5 An illustrative example of the use of the incidence matrix. Colored blocks in $\boldsymbol{x}$ denote non-zero flow entries, indicating that $\boldsymbol{x}$ is a sparse signal.

Consequently, the relationship between flow and traffic can be simply denoted as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{4}$$

5.2 Inferring Flow from Traffic

Since $|\boldsymbol{x}| = n^2$ and $|\boldsymbol{y}| = 2n$, A is a $2n \times n^2$ matrix. Hence, when $n > 2$, A necessarily admits a *non-degenerate kernel* [26]. In other words, the system of equation (4) is *under-constrained*, resulting in infinitely number of possible solutions of $\boldsymbol{x}$ given $\boldsymbol{y}$. This issue can be addressed by specifying the type of the solution that is required by the application. For example, we can require the solution to have only a limited number of non-zero entries (i.e., sparse). Based on the *a priori* information about the bike trip patterns, we impose the sparsity and locality of the strong flows on the solution using *regularization* technique.

5.2.1 Imposing Sparsity of Strong Flows

A natural way to enforce sparsity on the strong flow vector $\boldsymbol{x}$ is to use the ℓ_0 norm, which counts the number of non-zero entries in $\boldsymbol{x}$ [21]. Formally, the ℓ_0 norm of $\boldsymbol{x}$ is defined as

$$\|\boldsymbol{x}\|_0 = |\{x^{(i)} | x^{(i)} \neq 0\}| \tag{6}$$

Then, we can find the sparse solution by solving the optimization problem

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}(\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_0) \tag{7}$$

where λ is the regularization parameter to control the trade-off between sparsity and reconstruction fidelity. Unfortunately, solving (7) is both numerically unstable and NP-

complete, requiring an exhaustive enumeration of all the non-zero entries in $\boldsymbol{x}$ [21]. However, the solution can still remain sparse when relaxing the ℓ_0 norm with the convex ℓ_1 norm, i.e., $\|\boldsymbol{x}\|_1 = \sum_i |x^{(i)}|$ can still be sparse [21]. Formally, the relaxed optimization problem becomes

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}(\|\boldsymbol{A}\boldsymbol{x} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1) \tag{8}$$

which is a convex optimization problem that is conveniently reduced to a linear program known as basis pursuit [27].

5.2.2 Imposing Locality of Strong Flows

Sparsity is only one aspect of the *a priori* information we know about the bike trip patterns. The other important property is the locality, meaning that most of the strong flows exist between geographically-close nodes. Without imposing such constraints, the solution may contain many non-zero flow values between very distant stations, which might be unlikely to happen in the real-world situation.

Therefore, we propose a *weighted regularization* method to enforce locality of strong flows. The basic idea is to give larger regularization weights to geographically-distant node pairs, and smaller weights to close pairs. To this end, we add a ℓ_2 regularization term to constrain the intensities of $x^{(i)}$ between geographically-distant node-pair. In particular, we construct a weight vector $\mathbf{w}$, where the value of each w_i is calculated based on the geographic-distance of the node-pair corresponding to x_i, i.e., $w_i = h(d(u, v))$, if and only if $x_i = f(u, v)$. In this paper, we simply choose h to be a linear function that normalizes the values of $d(u, v)$ to $[0, 1]$. Finally, the objective function becomes

$$\underset{\boldsymbol{x},\boldsymbol{x}}{\operatorname{argmin}}(\|\boldsymbol{A}\boldsymbol{x} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + (1-\lambda)\|\boldsymbol{w} \circ \boldsymbol{x}\|_2) \tag{9}$$

where $\circ$ denotes the element-wise product of two vectors, and $(1 - \lambda)$ is the regularization parameter to control the trade-off between locality and reconstruction fidelity.

5.2.3 Formulating the Ill-Posed Problem

Finally, we determine the bounds of the solution $\boldsymbol{x}$. The upper bound for each strong flow entry is set to be the minimum of the traffic of its corresponding stations. We use an intensity threshold δ as the lower bound of each strong flow, i.e., $x^{(i)} \geqslant T$. In summary, we formate the optimization problem as

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{x}}{\operatorname{argmin}} \quad & \|\boldsymbol{A}\boldsymbol{x} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + (1-\lambda)\|\boldsymbol{w} \circ \boldsymbol{x}\|_2 \\ \text{subject to} \quad & \delta \leqslant x^{(i)} \leqslant \min_{\forall A_{j,i}=1} y_j \\ & i = 1, 2, \ldots, n^2,\ j = 1, 2, \ldots, n \end{aligned} \tag{10}$$

As studied in [24], (10) is still a convex optimization problem. We use the Matlab convex solver **cvx** [28] to efficiently search for the solution.

6 Evaluation

In this section, we evaluate the performance of our method by assessing its ability to infer the actual bike trip patterns in Washington, D.C. and New York City. We first describe the experiment setup, and then present the parameter setting process. Finally we show the evaluation results comparing our method to two baselines in both cities.

6.1 Experiment Setup

6.1.1 Dataset Summary

We retrieve station feed data from the Capital Bikeshare System in Washington, D.C.[4)] and the Citi Bike System in New York City[5)], respectively. We automatically query both APIs at a frequency of one query per minute. After a data preprocessing step to remove invalid and abnormal values (e.g. negative bike number), we compile the *hourly station traffic data* over one year. In order to verify the flow inference results, we also collect the bike trip history datasets from both cities. Note that these datasets are usually released with a delay of three months or more. Table 1 shows a summary of the collected data from both cities.

Table 1 Summary of the collected datasets

Dataset	Item	DC	NYC
Station Feed	Duration	2014–2015	2014–2015
	Stations	201	328
	Bikes	3,296	4,077
Trip History	Duration	2014–2015	2014–2015
	Records	1,869,980	8,081,188

[4)] http://www.capitalbikeshare.com/data/stations/bikeStations.xml
[5)] http://www.citibikenyc.com/stations/json

6.1.2 Evaluation Metrics

We perform flow inference on a two-hour window basis. We separate the dataset into two half-year sets, each containing data in every other day. We learn the parameter δ using one half year's data, and test the performance using the other half year's data. We first evaluate the inference accuracy of the strong flows, which reflects the structure of city-wide bike trips. To this end, we compare the inferred results with the ground truth to compute the *precision* and *recall* of strong flow inference. More specifically, if an inferred strong flow between a directed station pair actually exists in the ground truth, we call it a *hit*. Based upon this, we define the precision of inference as:

$$precision = \frac{|\{\text{real-world strong flow}\} \cap \{\text{inferred strong flow}\}|}{|\{\text{inferred strong flow}\}|} \tag{11}$$

and the recall of inference as:

$$recall = \frac{|\{\text{real-world strong flow}\} \cap \{\text{inferred strong flow}\}|}{|\{\text{real-world strong flow}\}|} \tag{12}$$

We also compute the *F1-Score* [29] as:

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

to assess the overall performance of our model and assist in the model parameter selection.

In order to evaluate the performance of the overall flow inference, we compute the *Mean Absolute Percentage Error (MAPE)* as

$$MAPE = \frac{1}{n^2} \| \frac{\boldsymbol{x} - \hat{\boldsymbol{x}}}{\boldsymbol{x}} \|_2 \tag{14}$$

6.1.3 Baseline Methods

We compare our *Sparse-and-Weighted Regularization (SWR)* method with the following two baselines that use different regularization techniques.

- *Energy-based Regularization (ER)*: This is the classical approach to inverse problems, which uses ℓ_2 regularization to find a solution with the smallest energy (i.e., sum of squares) [21]. The objective function of this approach is written as

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}(\|\boldsymbol{A}\boldsymbol{x} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2) \tag{15}$$

  We note that this problem has the convenient closed-form solution $\boldsymbol{x} = (\boldsymbol{A}\boldsymbol{A}^T + \lambda I)^{-1}\boldsymbol{A}^T\mathbf{y}$. Unfortunately, this simple ℓ_2 regularization method fails to impose the sparsity and locality properties of the strong flows, and thus might not be able to recover the real-world flow patterns we need.
- *Sparse Regularization (SR)*: This method only imposes the sparsity property via ℓ_1 regularization, as formulated in Equation (8). However, as we have mentioned, this method does not address the locality of the bike trip patterns, and might obtain results with long-distance trips that are rarely observed in the real world.

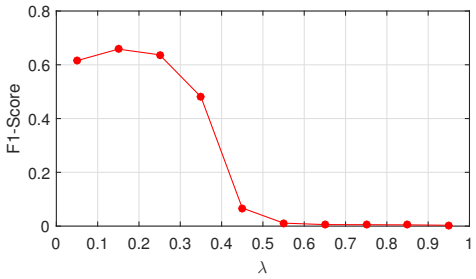


Fig. 6 Variation of *F1-Score* w.r.t. λ

6.2 Parameter Settings

Based on the first half year's training data, we set $\delta = 5$ for DC and $\delta = 12$ for NYC, respectively. The other important parameter is the regularization parameter λ, as presented in Equation (10). Applying a large λ might overstress the sparsity of strong flows, leading to unrealistic huge flows between certain station pairs. A small λ, on the other hand, might lead to small and dense flows, and thus affecting the accuracy of inference. In order to study the relationship between λ and the flow inference accuracy, we repeat the experiment by increasing λ from a small value to a relatively large one, and compute the corresponding *F1-Score* of inference. We use the first half year's data for λ selection. Based on the results (Figure 6), we select $\lambda = 0.15$ as the optimal value in our following experiments.

6.3 Flow Inference Results

We first show an overview of the flow inference results in both cities, and then discuss the performance of the flow inference methods.

6.3.1 Overview of the Inferred Flows

Figure 7 shows the flow inference results during the morning rush hours (7:00–9:00) of a typical weekday in DC and NYC. We can see that our method successfully recovers the strong flows in both cities, giving a overview of the flow patterns

during the morning rush hours in both cities.[6)]

From Figure 7(a) and 7(b), we observe two important morning rush hour flow patterns in DC. The first one is the massive incoming trips to the *Dupont Circle*, the center of a huge residential neighborhood with many foreign embassies, which indicates that people are probably riding bikes to work from home. The second pattern is the considerable outgoing trips from the *Union Station*, which is the transportation hub of subways and trains, indicating that people might ride bikes to complete their *last mile* trip to work. Similarly, from Figure 7(c) and 7(d), we observe that most of the strong flows concentrate on the *Penn Station* area, which is a huge transit hub of subway, train and bus. Most of the flows are outgoing trips to the Midtown and Downtown areas, indicating that the riders are connecting from commuter rail or bus. These observations bring to light one of the bike sharing system's greatest challenges: bike balancing, i.e., real-time relocating bike supplies to meet the demand at these popular locations.

6.3.2 Accuracy of Flow Inference

We conduct the flow inference experiments on a two-hour time window for both cities respectively, and separately compute the precision, recall, and *F1-Score* of the strong flow inference, as well as the *MAPE* of all the flows. The results of our method and the baselines are shown in Table 2.

Based on the results, we conclude that our method outperforms the two baselines in both cities. More specifically, the *ER* method minimizes the overall *MAPE* by recovering many nonzero flows, but it fails to identify the strong flows patterns, resulting in low *F1-Score* of flow inference. On the other hand, the *SR* method can infer several strong flows, however its overall error (*MAPE*) is relatively high. The most probable reason is that the *SR* method do not consider the geographic constraints of the bike flows, resulting in flows between geographically-distant stations, which is not likely to appear in reality. The proposed *SWR* method infers strong under the sparsity and locality constraints, resulting in relatively accurate strong flow inference while ensuring the overall performance.

Table 2 Flow inference results

	Precision		**Recall**		**F1-Score**		**MAPE**	
	DC	NYC	DC	NYC	DC	NYC	DC	NYC
ER	0.473	0.512	0.431	0.483	0.451	0.497	0.261	0.192
SR	0.712	0.837	0.672	0.712	0.691	0.769	0.648	0.554
SWR	0.755	0.835	0.781	0.803	0.768	0.819	0.373	0.288

7 Conclusion

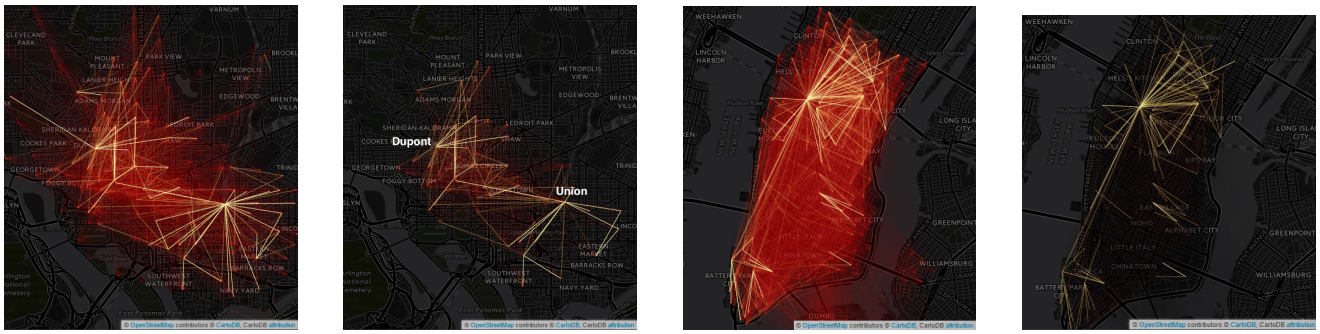
The emerging bike sharing systems have generated a large volume of bike usage data, providing an invaluable resource for researchers to understand human mobility patterns in urban environments. In this paper, we propose a sparse-and-weighted regularization method to infer bike trip patterns directly from the public station feeds, enabling applications such as bike balancing and station management in a timely manner. Through analysis of an empirical bike trip history dataset, we identify the sparsity and locality properties of public bike trip patterns. We then formulate bike trip pattern inference as an ill-posed inverse problem, and propose a sparse and weighted regularization method to incorporate the sparsity and locality in our solution. Finally, we evaluate our method using real-world bike sharing system data from two cities. The results show that our method outperforms two baseline methods in both cities by effectively recovering the strong bike flows.

In the future, we plan to investigate more fine-grained properties of public bike flow patterns, that are related to contextual factors such as weather conditions or altitude of stations. We also plan to apply the flow inference technique to other urban transportation systems, such as bus and metro, to study a wider variety of human mobility patterns in the cities.

[6)] We will discuss the flow directions in the following as we are not able to visualize them on the map.

References

1. Shaheen S, Guzman S, Zhang H. Bikesharing in Europe, the Americas, and Asia. Transportation Research Record: Journal of the Transportation Research Board, 2010, 2143(1): 159–167
2. LDA Consulting . 2013 Capital Bikeshare Member Survey Report. Washington, D.C.: Washington, D.C., 2013
3. Wang J, Gao F, Cui P, Li C, Xiong Z. Discovering Urban Spatio-temporal Structure from Time-Evolving Traffic Networks. In: Proceedings of Web Technologies and Applications. 2014, 93–104
4. Chemla D, Meunier F, Wolfler Calvo R. Bike sharing systems: Solving the static rebalancing problem. Discrete Optimization, 2013, 10(2): 120–146
5. Chen L, Zhang D, Pan G, Ma X, Yang D, Kushlev K, Zhang W, Li S. Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data. In: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, 571–575
6. Ji R, Gao Y, Liu W, Xie X, Tian Q, Li X. When Location Meets Social Multimedia: A Survey on Vision-Based Recognition and Mining for



(a) Real-world flow, DC (b) Inferred strong flow, DC (c) Real-world flow, NYC (d) Inferred strong flow, NYC

Fig. 7 The real-world and inferred strong flows during the rush hours (7:00–9:00) of a typical weekday morning in NYC. Color brightness corresponds to flow intensity.

Geo-Social Multimedia Analytics. ACM Transactions on Intelligent Systems and Technology, 2015, 6(1): 1–18

7. Yu Z, Xu H, Yang Z, Guo B. Personalized Travel Package With Multi-Point-of-Interest Recommendation Based on Crowdsourced User Footprints. IEEE Transactions on Human-Machine Systems, 2016, 46(1): 151–158
8. Chen L, Yang D, Jakubowicz J, Pan G, Zhang D, Li S. Sensing the Pulse of Urban Activity Centers Leveraging Bike Sharing Open Data. In: Proceedings of the IEEE International Conference on Ubiquitous Intelligence and Computing. 2015, 1–8
9. Garcia-Palomares J C, Gutierrez J, Latorre M. Optimizing the location of stations in bike-sharing programs: A GIS approach. Applied Geography, 2012, 35(1–2): 235–246
10. Contardo C, Morency C, Rousseau L M. Balancing a Dynamic Public Bike-Sharing System. CIRRELT, 2012
11. Singla A, Santoni M, Bartók G, Mukerji P, Meenen M, Krause A. Incentivizing Users for Balancing Bike Sharing Systems. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI'15. 2015, 723–729
12. Tikhonov N, Vasilii A. Solutions of Ill-Posed Problems. Winston, 1977
13. Engl H W, Hanke M, Neubauer A. Regularization of Inverse Problems. Springer Science & Business Media, July 1996
14. Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. The Annals of Statistics, 2006, 1436–1462
15. Boyd S, Vandenberghe L. Convex Optimization. Cambridge University Press, 2004
16. Guo B, Wang Z, Yu Z, Wang Y, Yen N Y, Huang R, Zhou X. Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm. ACM Computer Survey, 2015, 48(1): 1–31
17. Froehlich J, Neumann J, Oliver N. Sensing and Predicting the Pulse of the City through Shared Bicycling. In: Proceedings of the International Joint Conference on Artificial Intelligence. 2009, 1420–1426
18. Zhao Y, Chen L, Teng C, Li S, Pan G. GreenBicycling: A Smartphone-Based Public Bicycle Sharing System for Healthy Life. In: Proceedings of the IEEE International Conference on and IEEE Cyber, Physical and Social Computing. August 2013, 1335–1340
19. Randriamanamihaga A N, Côme E, Oukhellou L, Govaert G. Clustering the Vélib dynamic Origin/Destination flows using a family of Poisson mixture models. Neurocomputing, 2014, 141: 124–138
20. Combal B, Baret F, Weiss M, Trubuil A, Macé D, Pragnère A, Myneni R, Knyazikhin Y, Wang L. Retrieval of canopy biophysical variables from bidirectional reflectance: Using prior information to solve the ill-posed inverse problem. Remote Sensing of Environment, 2003, 84(1): 1–15
21. Baraniuk R. Compressive sensing. IEEE signal processing magazine, 2007, 24(4)
22. Vardi Y. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. Journal of the American Statistical Association, 1996, 91(433): 365–377
23. Wang L, Zhang D, Pathak A, Chen C, Xiong H, Yang D, Wang Y. CCS-TA: Quality-guaranteed Online Task Allocation in Compressive Crowdsensing. In: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, 683–694
24. Chawla S, Zheng Y, Hu J. Inferring the Root Cause in Road Traffic Anomalies. In: Proceedings of the IEEE International Conference on Data Mining. 2012, 141–150
25. Burden A M, Barth R. Bike-Share Opportunities in New York City. New York: Department of City Planning, 2009
26. Zabreyko P P, Koshelev A I, Krasnosel'skii M A, Mikhlin S G, Rakovshchik L S, Stet'senko V Y. Integral Equations: A Reference Text. Noordhoff International Publishing Leyden, 1975
27. Candes E, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 2006, 52(2): 489–509
28. Grant M C, Boyd S P. Graph Implementations for Nonsmooth Convex Programs. In: Blondel V D, Boyd S P, Kimura H, eds, Recent Advances in Learning and Control, number 371 in Lecture Notes in Control and Information Sciences, 95–110. Springer London, 2008
29. Powers D M. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. 2011

Longbiao Chen received the B.Sc. degrees in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently pursuing the Ph.D. degree in Department of Computer Science, Zhejiang University, and visiting Institut Mines-TELECOM / TELECOM SudParis in France. His research interests are mainly in urban computing, big data applications, and ubiquitous computing.

Xiaojuan Ma is an assistant professor of Human-Computer Interaction (HCI) at the Department of Computer Science and Engineering (CSE), Hong Kong University of Science and Technology (HKUST). She received the Ph.D. degree in Computer Science at Princeton University. She was a post-doctoral researcher at the Human-Computer Interaction Institute (HCII) of Carnegie Mellon University (CMU), and before that a research fellow in the National University of Singapore (NUS) in the Information Systems department. Before joining HKUST, she was a researcher of Human-Computer Interaction at Noah's Ark Lab, Huawei Tech. Investment Co., Ltd. in Hong Kong.

Thi-Mai-Trang Nguyen is an associate professor at University Pierre and Marie Curie (Paris 6) and doing research at Laboratoire d'Informatique de Paris 6 (LIP6), France. She received the PhD Degree in Computer Science from University of Paris 6, France, in 2003. The PhD thesis was co-supervised and carried-out at Ecole Nationale Superieure des Telecommunications (ENST-Paris). From 2004 to 2006, She was postdoctoral researcher at France Telecom in Rennes, France and at University of Lausanne, Switzerland. Her research interests include network architecture, network protocol design, and network data analytics.

Gang Pan received the B.Sc. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University. He has published more than 90 refereed papers. He visited the University of California, Los Angeles, Los Angeles, during 2007-2008. His research interests include pervasive computing, computer vision, and pattern recognition.

Jérémie Jakubowicz received the M.S. and Ph.D. degrees in applied mathematics from the Ecole Normale Supérieure de Cachan, Cachan, France, in 2004 and 2007, respectively. He was an Assistant Professor with Télécom ParisTech. Since 2011, he has been an Assistant Professor with Télécom SudParis, Evry, France, and an Associate Researcher with the CNRS. His current research interests include distributed statistical signal processing, image processing and data mining.