Inferring Bike Trip Patterns from Bike Sharing System Open Data

Longbiao Chen, Jérémie Jakubowicz
Institut Mines-Télécom; Télécom SudParis; UMR CNRS SAMOVAR
Paris, France
{longbiao.chen, jeremie.jakubowicz}@telecom-sudparis.eu

***Abstract*—Understanding bike trip patterns in a bike sharing system is important for researchers designing models for station placement and bike scheduling. By bike trip patterns, we refer to the large number of bike trips observed between two stations. However, due to privacy and operational concerns, bike trip data are usually not made publicly available. In this paper, instead of relying on time-consuming surveys and inaccurate simulations, we attempt to infer bike trip patterns directly from station status data, which are usually public to help riders find nearby stations and bikes. However, the station status data do not contain information about where the bikes come from and go to, therefore the same observations on stations might correspond to different underlying bike trips. To address this challenge, We conduct an empirical study on a sample bike trip dataset to gain insights about the inner structure of bike trips. We then formulate the trip inference problem as an ill-posed inverse problem, and propose a regularization technique to incorporate the *a priori* information about bike trips to solve the problem. We evaluate our method using real-world bike sharing datasets from Washington, D.C. Results show that our method effectively infers bike trip patterns.**

***Keywords*-Sparse regularization; bike sharing system; open data; urban computing**

I. Introduction

Bike sharing systems have been deployed in many cities to promote greener transportation and healthier life style [1], [2]. By studying the *bike trip patterns*, one can better understand the social dynamics and human mobility in the city [3]. However, *bike trip data* are usually not publicly available due to privacy and operational concerns. Instead, the authorities usually publish real-time *station status data*, *i.e.*, the number of bikes and docks available in stations at the time of query.

In this paper, we attempt to infer bike trip patterns directly from the public station status data. To this end, we first extract the station usage data from the station status data over a period of time, and then infer bike trip patterns from the station usage data. The first step is a simple sampling and aggregation task, while the second step, which corresponds to solving an *ill-posed inverse problem*, is the most challenging step. Fortunately, ill-posed inverse problems have been investigated for decades [4], [5] and it is now well known that our main challenge can be properly tackled provided we have enough *a priori* information to constrain the solution in a lower dimensional space than its original formulation [5]. This *a priori* information is usually injected through *regularization* [4]. We impose the *sparsity* and *locality* properties to the problem by assigning a *weight* to each node pair in the regularization term, penalizing node pairs with geographically-distant components. Using a real-world dataset, we show that our solution leads to a good approximation of the real-world bike trip patterns. The contributions of this work include:

1) We present a first attempt to infer the latent bike trip patterns from the public station status data in bike sharing systems.
2) We conduct an empirical study on the bike sharing system of Washington, D.C. to identify the sparsity and locality properties of bike trip patterns.
3) We cast the bike trip pattern inference problem as an ill-posed inverse problem, and propose a weighted-sparse regularization method that exploits the sparsity and locality of bike trip patterns to solve the problem.
4) We evaluate our approach using a real-world bike sharing system data from Washington, D.C. The results show that the proposed method can effectively infer the common bike trip patterns in the city.

II. Problem Formulation

A. Notations

We denote by $\mathcal{V}$ the set of stations and by $\mathcal{L} = \mathcal{V}^2$ the set of directed links between any station pair $(u, v) \in \mathcal{V}^2$, self loops included. We denote by n the cardinality of set $\mathcal{V}$, *i.e.*, the number of stations. We also denote by $d(u, v)$ the geographic-distance between two stations, 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 flux* $g_{in} : \mathcal{V} \to \mathbb{R}_+$ and the *outgoing flux* $g_{out} : \mathcal{V} \to \mathbb{R}_+$, representing 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 *flux*.

B. Problem

With the above-mentioned definitions, we then formulate our problem as a *flow inference* problem.

978-1-4799-9926-2/15/$31.00 ©2015 IEEE

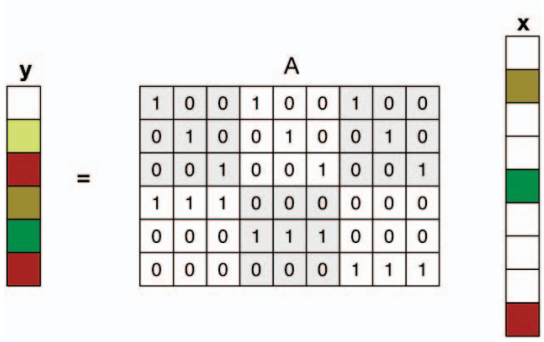


Figure 1: An illustrative example of the use of the incidence matrix. Colored blocks in $\boldsymbol{x}$ denote non-zero flow entries, indicating that $\boldsymbol{x}$ is a sparse signal.

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

We note that such a problem is an *ill-posed inverse* problem. since there are only $2n$ station-to-station fluxes, but we need to infer n^2 unknown flows between stations. In order to solve this problem, we propose a weighted-sparse regularization method to exploit the sparsity and locality structure of the bike network, as detailed in the following.

III. Methodology

In this section, we first elaborate on the modeling of the relationship between the flux and flow in the network, and then present our method for inferring flow from flux.

A. Modeling the Relationship Between Flux and Flow

We define the flux vector $\boldsymbol{y} = g(v), v = 1, 2, \ldots, n$, and the flow vector $\boldsymbol{x} = f(u, v), u, v = 1, 2, \ldots, n$. As the flux vector $\boldsymbol{y}$ is a measurement of the flow vector $\boldsymbol{x}$, we define the *incidence matrix* $\boldsymbol{A}$ to represent their relationship. More specifically, 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

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

In this way, the incidence matrix $\boldsymbol{A}$ transforms the station-to-station flows into the corresponding incoming and outgoing fluxes of stations. We give an example to illustrate the use of the incidence matrix in Figure 1. This bike network contains 3 nodes ($n = 3$), and thus having $n^2 = 9$ node-to-node flow entries and $2n = 6$ fluxes, which is produced by the 6×9 incidence matrix. For instance, the first flux 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. By arrange the flux and flow in different orders, the values of the binary entries in the incidence matrix might be different.

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{2}$$

B. Inferring Flow from Flux

We note that since $|\boldsymbol{x}| = n^2$ and $|\boldsymbol{y}| = 2n$, A is a $2n \times n^2$ matrix. Hence, as soon as $n > 2$, A necessarily admits a *non-degenerate kernel* [6]. In other words, the system of equation (2) is *under-constrained*, which implies that by itself there are infinitely many solutions to the system. This problem can be addressed by specifying the type of the solution that is required by the application. For instance, we can require the solution to have small element values or be sparse. Such additional constraints on the solution are usually injected by *regularization*. Based on the *a priori* information about the bike trip patterns, we impose the sparsity and locality properties in the solution.

To address this issue, we propose a *weighted-sparse regularization* method. The basic idea is to give larger regularization weights for geographically-distant node pairs, and smaller weights for close pairs. To this end, we rewrite the sparse regularization term by multiplying each entry of $\boldsymbol{x}$ by a weight based on the geographic-distance of the corresponding node-pair. More specifically, 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, we rewrite the regularization term as follows

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}(\|\boldsymbol{A}\boldsymbol{x} - \mathbf{y}\|_2^2 + \lambda \sum_{i=1}^{n^2} |w_i x_i|) \tag{3}$$

We note that (3) is still a convex optimization problem, as has been studied in [7]. The weighted-sparse regularization is essentially an ℓ_1 regularization method, and thus we can use the current efficient algorithms (*e.g.* convex optimization) to search for a solution. Finally, we solve the problem using convex optimization with bounds [8]. We use the well known Matlab convex solver **cvx** [9] to efficiently search for the solution.

IV. Evaluation

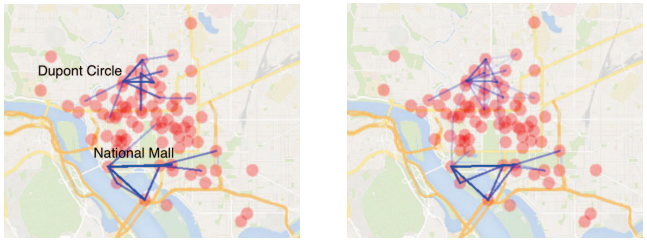
A. Dataset Summary

We collect the station status data of the Washington, D.C. Capital Bikeshare System[1] by automatically querying the API at a frequency of one time per minute. After a data processing step, we obtain the *daily station usage data* (*i.e.*, number of bikes rented and returned in a day) over a consecutive days.

[1] http://www.capitalbikeshare.com/data/stations/bikeStations.xml

Table I: Summary of the Washington, D.C. dataset.

Data collection Period	2012–2013
Selected bike stations	109
Average daily usage number per station	16
Maximum daily usage number per station	156



(a) Real-world flow. **(b)** Flow inference results.

Figure 2: The real-world significant flow and the inferred flow using the proposed methods. Each light blue line denotes a directed flow, while a dark line corresponds to the the superposition of two directed flows between the same node pair.

In order to evaluate the flow inference results, we also compile a bike trip dataset from Washington, D.C. from 2012–2013, as described in Section III(B).

B. Flow Inference Results

Figure 2 shows the inferred significant flow of the proposed method as well as the real-world flow in Washington, D.C. in a typical day. We observe that the real-world bike flows concentrate on two areas: (1) *Dupont Circle*, a historic neighborhood with many embassies and restaurants, and (2) *National Mall*, the central park of the city with numerous monuments and museums. In Figure 2b, we see that our proposed method (*WSR*) recovers most of the significant flow in these areas well.

V. Conclusion

While enabling a green transportation means and a healthy lifestyle, the emerging bike sharing systems have also generated a large volume of usage data, providing invaluable resource for researchers to understand the social dynamics and human mobility in urban areas. In this paper, we propose a method to infer bike trip patterns directly from the public station status data, enabling better understanding of the bike sharing systems and their users. First, by conducting an empirical study on a real-world bike sharing system, we identify the sparsity and locality properties of bike trip patterns. Then, we formulate the bike trip pattern inference problem as an ill-posed inverse problem, and propose a weighted-sparse regularization method to solve the problem by incorporating the *a priori* information about the bike trip patterns. Finally, we evaluate our method using real-world bike sharing datasets collected from Washington, D.C. The results show that our method can effectively infers the common bike trip patterns in the city.

In the future, we plan to apply the flow inference technique to more urban transportation systems, such as bus and metro, to study a wider variety of human mobilities in the cities. We also plan to evaluate our work in different temporal groups (*e.g.* morning and afternoon) and different cities (*e.g.* New York City and Paris).

References

[1] 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 *In Proc UbiComp'15*, pp. 571–575.

[2] L. C. Yifan Zhao, "GreenBicycling: A Smartphone-Based Public Bicycle Sharing System for Healthy Life," pp. 1335–1340, 2013.

[3] D. Zhang, B. Guo, and Z. Yu, "The emergence of social and community intelligence," *Computer*, vol. 44, no. 7, pp. 21–28, 2011.

[4] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*. Springer Science & Business Media, Jul. 1996.

[5] N. Tikhonov and A. Vasilii, *Solutions of ill-posed problems*. Vh Winston, 1977.

[6] P. P. Zabreyko, A. I. Koshelev, M. A. Krasnosel'skii, S. G. Mikhlin, L. S. Rakovshchik, and V. Y. Stet'senko, *Integral equations: A reference text*. Noordhoff International Publishing Leyden, 1975.

[7] H. Zou, "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.

[8] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[9] M. Grant, S. Boyd, and Y. Ye, *CVX: Matlab software for disciplined convex programming*, 2008. http://cvxr.com/cvx/download/