OD-HyperNet: A Data-Driven Hyper-Network Model for Origin-Destination Matrices Completion Using Partially Observed Data

Yuxuan Xiu, Wanda Li, Jing Yang (Sunny) Xi, Wai Kin Victor Chan*
Shenzhen Environmental Science and New Energy Technology Engineering Laboratory, Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, P.R. China
yuxuanxiu@gmail.com, wdli10@outlook.com, sunnyx@berkeley.edu, chanw@sz.tsinghua.edu.cn

***Abstract*—Estimating the inter-city population flow is critical for modeling the spread of COVID-19. However, for most cities, it is difficult to extract accurate population numbers for inflow and outflow. On the other hand, mobile carriers and Internet companies can estimate the distribution of population flow by tracking their users; but their data only cover part of the travelers. In this paper, we present a data-driven hyper-network model to aggregate these two types of data and complete the inter-city OD matrix. We first propose a cross-layer breadth-first traversal algorithm to estimate the inflow and outflow population of each city, then complete the OD matrix with an optimization model. Our experiments on a real-world dataset prove the accuracy and efficiency of our model.**

Keywords—hyper-network, data-driven modeling, origin-destination matrix

I. INTRODUCTION

The current COVID-19 outbreak is one of the greatest challenges that humanity has ever faced, and consequently, many new research questions have surfaced. Population movement between regions, on which many epidemic models rely, is an important factor for modeling the spread of COVID-19 [1, 2]. For the first wave of the COVID-19 epidemic that started in Wuhan, China, it has been found that the geographic distribution of COVID-19 infections can be rather accurately predicted by the population outflow from Wuhan [3]. For the new clusters of COVID-19 cases emerging in Jilin and Beijing recently, population movement data are also fundamental in estimating the risk of the epidemic spreading to other cities.

The number of people traveling between different places can be specified using the Origin Destination (OD) matrix, where the element n_{ij} represents the number of people moving from region i to region j. As an important method in the field of transportation and logistics, OD matrix estimation has long been studied. A characteristic of the OD matrix estimation problem is that it relies heavily on the available data. Researchers have proposed a variety of OD matrix estimation algorithms to handle different types of input data. Before pervasive computing devices (e.g. mobile phone and GPS devices) are widely used, the most commonly used data source is traffic counts. For this type of input data, only the traffic flows at specific positions are observed, while the trajectories of travelers are missing. A commonly used approach is to impose some assumptions (e.g., the route choice pattern) to the OD matrix. Transport demand models, such as the gravity model [4] and the opportunity model [5], are calibrated based on the traffic counts to perform the estimation. With the development of mobile devices, digital footprints are available nowadays, enabling data-driven OD estimation algorithms. Data-driven algorithms usually leverage cell phone data [6] or GPS data [7] to estimate the OD matrix, focusing on tracking travelers based on the raw positioning data.

This paper aims to tackle a different OD matrix completion problem, which is encountered in modeling the inter-city spread of COVID-19 in China. On one hand, only a small number of cities can provide accurate traffic counts (i.e., the total number of inflow and outflow population), as most local governments are unable to track all of the origins or destinations of the population flow. On the other hand, mobile carriers (e.g., China Mobile) and Internet companies (e.g., Baidu) can track their users based on the positioning data, thus calculating the distribution of population inflow and outflow of each city. However, they may fail to obtain the total number of people moving in and out of cities, since the positioning data can only cover their own users.

In this paper, we assume that the following two types of data are available: (1) the accurate total population outflow (or inflow) of one or a few cities, and (2) the distribution of population inflow and outflow of each city. This paper aggregates such information to complete the inter-city OD matrices, which is rather different from the two typical OD matrix estimation problems mentioned earlier. On one hand, extracting trajectories of travelers is beyond the scope of this paper. Rather, we assume that such data is already provided by mobile carriers or Internet companies. On the other hand, traffic counts-based models are not very suitable for our problem, where the aggregation of existing information alone is sufficient to complete the OD matrix. Additional inferences and assumptions are not necessary.

One powerful model to aggregate information from multiple data sources is the hyper-network model [8-11]. Its hyper-network structure could transform and combine complicated mathematical relationships between data into one structure. A standard hyper-network model is composed of several single layers and inter-layer connections between them. Data from the same information source form a single-layer network, where each data element serves as a node, and the relations between them form intra-layer edges. The other kind of edge, the inter-layer connections, are set by the relationship between data from different information sources.

In this paper, we propose OD-HyperNet, a data-driven hyper-network model, to integrate the hyper-network model

into the task of completing OD matrix. In real-world migration population estimating tasks, it works well in obtaining the inflows and outflows in most of the cities when such data of one or several cities are known. OD-HyperNet is composed of two single-layer networks, namely the population inflow network and the population outflow network. We use the inter-layer connections to link the nodes belonging to the same cities in the two-layer hyper-network, thereby connecting and aligning the two single-layer networks. Within OD-HyperNet, we utilize a cross-layer breadth-first traversal algorithm to estimate the population inflows and outflows. We also propose a practical completion procedure to lower the sparsity of the OD matrix.

Our experiment is based on the Baidu Migration dataset. It includes two parts: (1) the total number of people moving in and out of each region during a specific period, and (2) the inter-regional traffic flow, i.e., the number of people moving from one region to another. We evaluate the overall estimation accuracy base on an R-Squared metric, then show the influence of time and starting node. Results suggest that our model can complete the OD matrix very well and provide precise estimations with an accuracy of at least 0.975. To enhance the performance, we recommend to start traversal from cities with relatively high traffic and avoid tremendous fluctuations in data sequence.

II. Problem Definition

In this section, we stress the urgency of completing the origin-destination (OD) matrix, then present the assumptions and mathematical formulation for this problem.

A. Partially Observed Origin-Destination Matrix

A high-quality transit OD matrix acts as a solid foundation in many application problems, such as transportation planning [12], market evaluation [13], and epidemiological estimation [14]. Unfortunately, it is not always easy for people to obtain enough suitable data to build it.

There are two standard data providers for OD flow, but both of them have drawbacks. First, mobile network operators or cellphone APPs (e.g. China Mobile or Baidu Map) could generate real-time data by tracking their users. These data suffer from two aspects: they could only access their own users' moving trace, while the market share of them in different places is hard to analyze. Another provider, the government, could acquire more reliable data by surveys. However, such methods are expensive and sometimes result in old-fashioned data. Such a condition urges us to discover an appropriate method to complete the partially observed OD matrix.

In this paper, we focus on migration data from mobile carriers and cellphone APPs, for real-time information is more valuable to in driving decisions. To overcome the typical drawback of missing OD flows, we propose a data-driven modeling method, which can be applied to any dataset of this kind.

B. Basic Assumptions and Mathematical Formulation

Migration data provided by mobile carriers and cellphone APPs mainly fall into two categories: (1) the traffic flow between OD pairs, and (2) the total moving-in/out population of selected cities. Without loss of generality, we process these two types of data following three assumptions:

1) Assumption 1: The market occupancy of a particular data provider in city i is α_i. Normally, when $i \neq j$, $\alpha_i \neq \alpha_j$.

2) Assumption 2: The users of one data provider are uniformly distributed in the moving-in/out population. i.e., in city i, if its moving-in population is N_i^{in} and moving-out population is N_i^{out}, then the inflow/outflow users of the data provider are $\alpha_i N_i^{in}$ and $\alpha_i N_i^{out}$.

3) Assumption 3: The inflow/outflow population occupies a relatively small portion of the total population in a city.

The analysis process runs as following: For the first type of data, we suppose that the number of people moving out of city i to city j is n_{ij}^{out}, the total number of people moving out of city i is N_i^{out}, then the proportion moving out of city i to city j is defined as $p_{ij}^{out} = n_{ij}^{out} / N_i^{out}$. Similarly, the proportion of newcomers of city j who come from city i is defined as $p_{ji}^{in} = n_{ji}^{in} / N_j^{in}$, where N_j^{in} represents the total number of migrants in city j and n_{ji}^{in} among them are from city i. Since $n_{ij}^{out} = n_{ji}^{in}$, $p_{ij}^{out} \bullet N_i^{out} = p_{ji}^{in} \bullet N_j^{in}$. Therefore, once the moving-in/out population in a city is known, the data of its related cities can be inferred. The first type of data is usually released as the outflow proportion matrix $\mathbf{P}^{out} = \{p_{ij}^{out}\}$ and the inflow proportion matrix $\mathbf{P}^{in} = \{p_{ji}^{in}\}$, which are estimated based on the user positioning data. In practice, these matrices are usually rather sparse, because there may not be any users travelling between for most of the city pairs.

The second type of data can be classified into two categories: (1) Total inflow and outflow population, i.e., N_i^{in} and N_i^{out}, which are released by the government. Such data is scarce but somewhat accurate; (2) Migration Index estimated by mobile network operators or Internet companies, which is a function of the moving-in/out users of a data provider. We indicate the moving-in and moving-out migration index of city i by $I_i^{in} = f(\alpha_i N_i^{in})$ and $I_i^{out} = f(\alpha_i N_i^{out})$ respectively.

The input of our model is the partially observed transport probability matrix $\mathbf{P}^{out}$ and $\mathbf{P}^{in}$. Without loss of generality, we assume that only the *Top-K* entities in each row of these two matrices are observed. That is, we only know the origins/destinations with the *Top-K* transport probabilities of each city. Besides, we also need at least one city whose accurate inflow or outflow population is known. That is, at least one N_i^{out} or N_i^{in} is needed.

The output of our model is the completed OD matrix $\{OD_{ij}\}_{|\mathbf{N}|\times|\mathbf{N}|}$, where $OD_{ij}=n_{ij}^{out}$ is the number of people who transport from city i to city j.

III. Model

A. OD-HyperNet: the Origin-Destination Hyper-Network Model

Based on the original definition of hyper-network [8, 10, 11], we propose the OD-HyperNet model to describe OD flow data.

A hyper-network $\mathbf{H}(\mathbf{B}(\mathbf{N},\mathbf{E}),\mathbf{M},\mathbf{R})$ is constructed from a base network $\mathbf{B}(\mathbf{N},\mathbf{E})$, in which $\mathbf{N}$ represents the set of all

members (nodes) and $\mathbf{E}$ represents the set of connections (edges). The set of single layer networks is denoted as $\mathbf{M}$, and the number of layers in the hyper-network is $|\mathbf{M}|$. In the behavior matrix $\mathbf{R}$, each element $\mathbf{R}(i,M)$ describes the connection pattern of node i in layer M.

Intuitively, the cities and the migration flows between them constitute a network, where cities are nodes, and the migration flows between the cities are edges. We build two migration networks, an inflow network, and an outflow network based on the previous definition.

In the outflow network, the weight of node i is set to N_i^{out}, which is the total number of people moving out of city i. The weight of directed edge (i,j) represents p_{ij}^{out}, which is the proportion of migrants from city i who move to city j. Similarly, in the inflow network, the weight of node j, N_j^{in}, is the total number of people moving into city j; the weight of directed edge (j,i), p_{ji}^{in}, is the proportion of the inflow population of city j coming from city i.

Notice that directed edge (i,j) in the outflow layer means that people move from city i to city j, while an edge (t,k) in the inflow layer shows the opposite direction, i.e., people move to city t from city k. Therefore, out-neighbors in the *outflow* layer represent the *origins* of inter-city trips, while the out-neighbors in the *inflow* layer are the *destinations*. Similarly, the *in-neighbors* in the *outflow* layer denote *destinations*, while *in-neighbors* in the *inflow* layer are *origins*. As available public data are often restricted to the *Top-K* origins/destinations of each city, the out-degree of each vertex in both layers is a constant, K. However, the in-degrees of each vertex in both layers can differ significantly. A high in-degree in the outflow layer denotes that the city acts as a frequent destination of the inter-city migration, while higher in-degree in the inflow layer means the city is more likely to be a common destination.

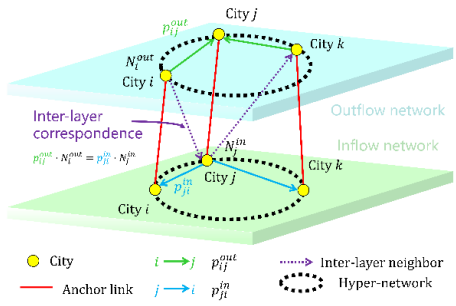

Fig. 1. An illustrative example for the inflow-outflow hyper-network

To construct the anchor links, we refer to the inter-layer correspondence between nodes (i.e., N_i^{out} and N_i^{in} belong to the same city i) and edge ($p_{ij}^{out} \bullet N_i^{out} = p_{ji}^{in} \bullet N_j^{in}$). Figure 1 gives an example of our proposed migration inflow-outflow hyper-network.

B. *Inferring Inflow/Outflow Population based on Cross-Layer Breadth-First Search*

In most of the cases, we can calculate the proportion p_{ij}^{out} and p_{ji}^{in} from the data collected by mobile phone network operators. However, the precise numbers of inflow/outflow population (i.e., N_i^{out} and N_j^{in}) are difficult to obtain. To conquer the problem, we first start from one city i with known inflow/outflow population, then iteratively derive the data of most cities based on $p_{ij}^{out} \bullet N_i^{out} = p_{ji}^{in} \bullet N_j^{in}$.

We apply the notion of interlayer neighbor to represent such relationship. City i and city j are interlayer neighbors if there is a pair of edges with opposite directions in two layers (e.g., directed edges $(i,j) \in \mathrm{E}_{out}$ and $(j,i) \in \mathrm{E}_{in}$, where E_{out} and E_{in} represent the set of edges of the outflow and inflow network, respectively). For one pair of inter-layer neighbors, if the total outflow population of city i is known, then the total inflow population of city j can be expressed as $N_j^{in}=p_{ij}^{out} \bullet N_i^{out} / p_{ji}^{in}$. Thus, the knowledge of only one or a few cities' inflows/outflows is enough to power the whole research. In Table I, we propose a cross-layer breadth-first search (BFS) algorithm for a two-layer hyper-network as an example. Two critical catches here are (1) cross-layer BFS needs two queues to store the nodes of inflow and outflow layer, and (2) cross-layer BFS visits inter-layer neighbors iteratively rather than intra-layer neighbors.

TABLE I. CROSS-LAYER BREADTH-FIRST SEARCH ALGORITHM

Input: Weights of all the edges in each layer. Weights of one or several nodes. **Output**: Weights of all the nodes in each layer.
(1) Establish two queues: Q_{in} and Q_{out}, enqueue nodes with known inflow data into Q_{in}, while nodes with known outflow data are loaded into Q_{out}. Here we suppose the weight of node i in the outflow layer N_i^{out} is known, thus we enqueue node i into Q_{out}.
(2) While Q_{out} is not empty, dequeue a node i and visit its out-neighbors of the outflow layer, i.e., find the set of node j where there exists $(i,j) \in E_{out}$.
(3) For each out-neighbor j, check whether edge $(j,i) \in E_{in}$ exists in the inflow layer. If such edge exists, we calculate the weight of node j in the inflow layer by $N_j^{in}=p_{ij}^{out} \bullet N_i^{out} / p_{ji}^{in}$ and load node j into Q_{in}.
(4) While Q_{in} is not empty, dequeue a node j and visit its out-neighbors of the inflow layer, i.e., find the set of node k where there exists $(j,k) \in E_{in}$.
(5) For each out-neighbor k, check whether edge $(k,j) \in E_{out}$ exists in the outflow layer. If so, we calculate the weight of node in the outflow layer by $N_k^{out}=p_{jk}^{in} \bullet N_j^{in} / p_{kj}^{out}$ and load node k into Q_{out}.
(6) Iteratively perform step (2) to step (5) until Q_{in} and Q_{out} are both empty.

C. *OD Matrices Completion*

In the previous section, we obtain the weights (N_i^{out} and N_i^{in}) of each node i in both layers. Moreover, the weights of directed edges in each layer can be expressed as

$p_{ij}^{out}=n_{ij}^{out}/N_i^{out}$ and $p_{ji}^{in}=n_{ji}^{in}/N_j^{in}$, where $n_{ij}^{out}=n_{ji}^{in}$ is the OD flow from city i to city j. It enable us to calculate the OD matrix $\{OD_{ij}\}_{|\mathbf{N}|\times|\mathbf{N}|}$, where $OD_{ij}=n_{ij}^{out}=n_{ji}^{in}$ and $|\mathbf{N}|$ is the number of the cities.

We can first calculate the OD matrix using either the inflow layer or the outflow layer. Since only the *Top-K* origins/destinations and their proportion are known, each row of the OD matrix only have K non-zero elements. Therefore, the number of the non-zero elements in the OD matrix is $\|\mathbf{OD}\|_0=|\mathbf{N}|K$, where $\|\bullet\|_0$ denotes the L0-norm. Our goal is to complete the sparse matrix as much as possible, which can be regarded as a matrix completion problem [15]. Existing literatures reveal the spatial affinity feature of the OD matrix, that is, there are some rows similar to each other [16]. Therefore, we assume the OD matrices are low-rank.

However, compared with the general low-rank matrix completion problem, the OD matrix completion problem has a lot of intrinsic characteristics that can be leveraged. Much information can be provided by aggregating data from both layers in our OD-HyperNet model. Our OD matrix completion method has three steps.

1) Step 1: We calculate two OD matrices based on the inflow layer and the outflow layer respectively. For the outflow layer, the OD matrix is calculated by $OD_{ij}^{out}=n_{ij}^{out}=p_{ij}^{out}\bullet N_i^{out}$. For the inflow layer, the OD matrix is calculated by $OD_{ij}^{in}=n_{ji}^{in}=p_{ji}^{in}\bullet N_j^{in}$.

2) Step 2: We aggregate these two OD matrices based on the following equation,

$$OD_{ij}=\begin{cases}(OD_{ij}^{out}+OD_{ij}^{in})/2, \text{if } p_{ij}^{out}\bullet p_{ji}^{in}\neq 0\\ \max(OD_{ij}^{out},OD_{ij}^{in}), \text{if } p_{ij}^{out}\bullet p_{ji}^{in}=0 \text{ and } p_{ij}^{out}\neq p_{ji}^{in}\\ 0, \text{if } p_{ij}^{out}=p_{ji}^{in}=0\end{cases}. \quad (1)$$

3) Step 3: To further reduce the sparsity of the OD matrix obtained by Equation (1), we apply the following low-rank matrix completion formulation,

$$\begin{aligned}&\text{minimize} \quad \|\mathbf{X}\|_*\\ &\text{subject to } X_{ij}=OD_{ij} \quad (i,j)\in\Omega\\ &\sum_{i=1}^{|\mathbf{N}|}X_{ij}=N_j^{in} \quad j=1,2,\cdots|\mathbf{N}|,\\ &\sum_{j=1}^{|\mathbf{N}|}X_{ij}=N_i^{out} \quad i=1,2,\cdots|\mathbf{N}|\\ &X_{ij}\geq 0 \quad i,j=1,2,\cdots|\mathbf{N}|\end{aligned} \quad (2)$$

where $\mathbf{X}$ is the decision variable (i.e., the completed matrix) and $\|\mathbf{X}\|_*$ is the nuclear norm of $\mathbf{X}$, which is applied to approximate the rank of $\mathbf{X}$. The set Ω contains the observed data. Program (2) aims at seeking the matrix $\mathbf{X}$ with the lowest rank that fits the observed data and the total inflow and outflow population.

Theoretically, Program (2) provides a proper formulation of the OD matrix completion problem. However, it may not be feasible in practice due to the bias in data. Thus, we provide an alternative formulation, Program (3), by relaxing some of the constraints.

$$\begin{aligned}&\text{minimize} \quad \|\mathbf{X}\|_*\\ &\text{subject to } X_{ij}=p_{ij}^{out} \quad (i,j)\in\Omega\\ &\sum_{j=1}^{|\mathbf{N}|}X_{ij}=1 \quad i=1,2,\cdots|\mathbf{N}|\\ &X_{ij}\geq 0 \quad i,j=1,2,\cdots|\mathbf{N}|\end{aligned}. \quad (3)$$

In Program (3), $p_{ij}^{out}=OD_{ij}/N_i^{out}$, that is ,we normalize each row of the OD matrix by N_i^{out}. Therefore, the decision variable X_{ij} is the proportion rather than the number of travelers. Therefore, the elements of the completed OD matrix is calculated X_{ij}. Program (3) is a constrained convex optimization problem, which can be solved by solvers such as CVXPY [17, 18].

IV. Experiments

In this section, we apply the OD-HyperNet model to a case study using the Baidu Migration dataset, then validate the robustness and the accuracy of the OD flow estimation.

A. Data Description

The experiments are carried out based on the Baidu Migration dataset from Jan 1, 2020 to Jan 31, 2020. We collect the migration data of 352 cities. For each city, the dataset contains the following information: (1) *Top-100* origins or destinations with the proportion of daily population moving in/out of the given city, and (2) Baidu Migration Index that reflects the size of the population moving in or out the given city.

TABLE II. Examples of the Origin-Destination Data

City A	City B	Date	Migration Type	Proportion
Wuhan	Xiaogan	2020/01/23	Move-Out	16.91%
Wuhan	Huanggang	2020/01/23	Move-Out	14.12%
…	…	…	…	…
Shenzhen	Dongguan	2020/01/23	Move-In	16.77%
Shenzhen	Huizhou	2020/01/23	Move-In	11.84%
…	…	…	…	…

TABLE III. Examples of the Migration Index Data

City A	Date	Migration Type	Migration Index
Wuhan	2020/01/23	Move-Out	11.14
Wuhan	2020/01/23	Move-In	1.75
…	…	…	…
Shenzhen	2020/01/23	Move-Out	17.78
Shenzhen	2020/01/23	Move-In	3.37
…	…	…	…

Table II shows the format of the origin-destination (OD) data. For example, the population who migrated from Wuhan to Xiaogan on Jan 23, 2020 accounts for 16.91% of the total outflow population of Wuhan. Likewise, the population that flows into Shenzhen from Dongguan on that day accounts for 16.77% of Shenzhen's total inflow population.

Table III lists the migration scale index data. It reads that the outflow population scale indexes of Wuhan and Shenzhen on Jan 23 were 11.14 and 17.78, respectively, which means that the total outflow population of Shenzhen on Jan 23 is 1.6 times higher than Wuhan.

In this paper, we first use the OD data to construct OD-HyperNet models for each day, then infer Baidu Migration Index based on the OD-HyperNet models. Assuming we only know one city's Baidu Migration Index, we validate the performance of our proposed inference method. The ground truth is the Baidu Migration Index of all the other cities.

B. *Robustness and Accuracy of Inflow/Outflow Population Estimation*

As is illustrated in Section II, the accuracy of the OD matrix depends only on the accuracy of the estimated inflow and outflow population. Therefore, we evaluate the accuracy of the inflow/outflow population estimation. We also evaluate the robustness when starting from different vertices to perform cross-layer BFS. We demonstrate that only one starting vertex is needed, and our proposed method is not sensitive to the starting vertex. This finding is rather important because the set of candidates for the starting vertex (i.e., the cities whose inflow or outflow population is known) is rather limited in most of the practical applications.

Since the Baidu Migration dataset covers the *Top-100* origins/destinations of each city, the inter-layer and the intra-layer connections are rather dense in the OD-HyperNet models. However, this question remains to be determined: How many starting points does the cross-layer BFS need at least to infer the inflow and outflow population of all the other cities?

Since one city corresponds to two nodes (i.e., city i corresponds to N_i^{out} in the outflow layer and N_i^{in} in the inflow layer), there are 704 nodes in our proposed hyper-network model. We first choose each node as the starting point to perform the cross-layer BFS algorithm. It turns out that the cross-layer BFS starting from any given node in the OD-HyperNet can reach all the other nodes. Therefore, we can select only one city, rather than a set of cities, as the starting point for each OD-HyperNet model to perform cross-layer BFS.

We choose 10 different nodes as the starting point of the cross-layer BFS algorithm and evaluate the overall estimation accuracy based on the R-Squared metric.

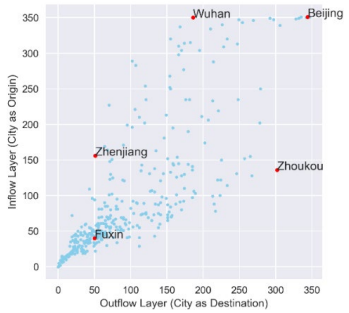

Fig. 2. The in-degrees in both layers and the starting points

The starting points are chosen based on the in-degrees of the corresponding cities in both layers. Figure 2 shows a scatter plot, selecting the in-degrees of outflow layer and inflow layer to appear on the x-axis and y-axis, respectively. We choose 5 typical cities for our study and mark them in red in Figure 3. For each city, we separately assume the inflow population N_i^{in} or outflow population N_i^{out} is known. We use the outflow migration index to represent N_i^{out}, and set N_i^{in} to the inflow migration index, thus obtaining 10 starting points to perform the cross-layer BFS.

We evaluate the overall estimation accuracy base on the R-Squared metric defined as Equation (4). This metric is commonly used to evaluate the accuracy of the OD flow estimation [19]. The ground-truth of the inflow/outflow population of city i is denoted as N_i^{in} and N_i^{out}, the corresponding average values are expressed as $\overline{N_i^{in}}$ and $\overline{N_i^{in}}$. The estimation results are $\hat{N}_i^{in}$ and $\hat{N}_i^{out}$. Better estimations are indicated by higher R-Squared values [20].

$$\theta = 1 - \frac{\sum_{i=1}^{n}\left(N_i^{out} - \hat{N}_i^{out}\right)^2 + \sum_{i=1}^{n}\left(N_i^{in} - \hat{N}_i^{in}\right)^2}{\sum_{i=1}^{n}\left(N_i^{out} - \overline{N_i^{out}}\right)^2 + \sum_{i=1}^{n}\left(N_i^{in} - \overline{N_i^{in}}\right)^2}. \tag{4}$$

Figure 3 illustrates the R-Squared values for the inference results of the cross-layer BFS starting from N_s^{out} or N_s^{in}, where s is one of the 5 cities.

There are three key observations of Figure 3. First, the overall inference results are rather accurate, and the performance is relatively robust to different choices of the starting point. Second, choosing vertex with higher in-degrees improves the accuracy of the inference regardless of the layer. This indicates that transportation centers are more effective as the starting points. Third, the inference accuracy is rather stable from Jan 1 to Jan 22. However, the overall inference accuracy dramatically fluctuates after Jan 23. The reason might be many cities have had issued traffic restricting policies after Jan 23. Since Baidu may calculate the original data (e.g., N_i^{out} and p_{ij}^{out}) based on specific traveling pattern, significant changes in traveling pattern can lead to larger error and instability.

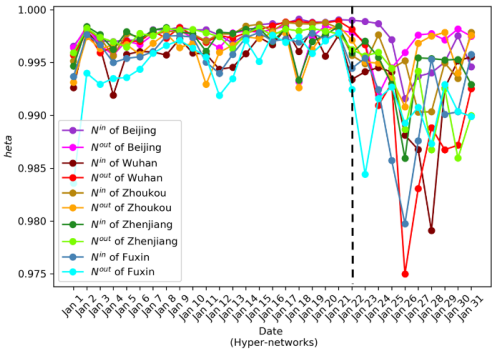

Fig. 3. Inference accuracy at varying starting points

We further calculate the estimation error of each value based on $E_i^{io} = (N_i^{io} - \hat{N}_i^{io}) / N_i^{io}$, where N_i^{io} represents either N_i^{in} or N_i^{out} of the city i. Figure 4 shows that most of $E_i^{io} \in (-0.2, 0.2)$, that is, most of the inference error is within the range of $(-0.2N_i^{io}, 0.2N_i^{io})$. We compare the distribution of E_i^{io} for the cross-layer BFS algorithm starting from 4

different vertices (i.e., the inflow/outflow population of Beijing and Fuxin) in Figure 5. This agrees with our previous results that starting from N_i^{in} or N_i^{out} of big cities can be more accurate.

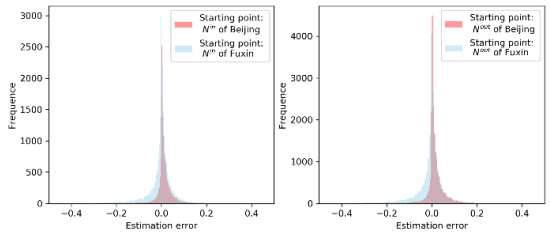

Fig. 4. Distribution of error at varying starting points

C. Completing OD Matrix

In this part, we evaluate the model's performance in completing the OD matrices. Since we do not have the ground truth of the missing data, we only compare the sparsity of the matrix after Step (1) ~ (3).

Figure 5 gives examples of the OD matrices after each step. Each row represents the OD matrices at one day. The three columns are the OD matrices after Step (1) ~ (3), respectively. For better visualization, we normalize each column of the OD matrices by the total outflow population of the corresponding city, that is, we obtain $p_{ij}^{out}=OD_{ij} / N_i^{out}$ and visualize the matrix $\left\{p_{ij}^{out}\right\}_{|\mathbb{N}|\times|\mathbb{N}|}$.

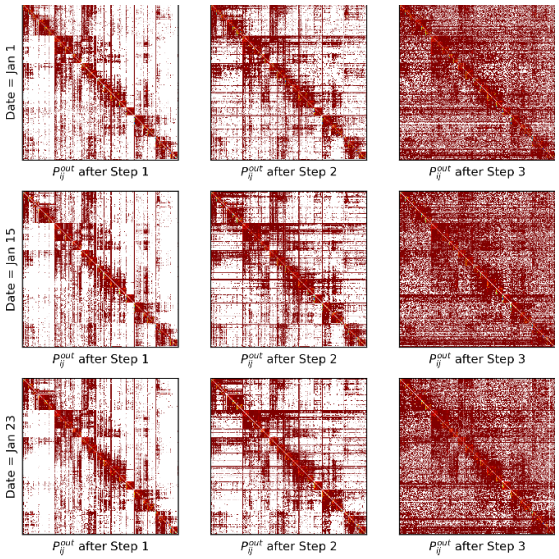

Fig. 5. The matrix completion results of three steps

Figure 5 demonstrates that our OD matrix completion procedure gradually decreases the sparsity of the OD matrices step by step to ultimately form a rather dense OD matrix. Interestingly, it can also be noticed that there are some time-invariant patterns of the matrices in each column.

V. Conclusion

Accurate tracking of population flow is crucial in the prediction and containment of possible infections during the current COVID-19 epidemic. However, real-world data are not accessible in many cases, which may result in sparse OD matrices. This paper solves the problem by incorporating hyper-network model with data-driven method. Our model is able to interpolate and infer missing pieces of data and provides high estimation accuracy and matrix completeness.

Acknowledgement

This research was funded by the National Natural Science Foundation of China (Grant No. 71971127) and the Hylink Digital Solutions Co., Ltd. (120500002).

References

[1] D. Brockmann and D. Helbing, "The hidden geometry of complex, network-driven contagion phenomena," Science, vol. 342, no. 6164, pp. 1337-1342, 2013.

[2] D. Taylor et al., "Topological data analysis of contagion maps for examining spreading processes on networks," Nature Communications, vol. 6, no. 1, pp. 1-11, 2015.

[3] J. S. Jia, X. Lu, Y. Yuan, G. Xu, J. Jia, and N. A. Christakis, "Population flow drives spatio-temporal distribution of COVID-19 in China," Nature, pp. 1-5, 2020.

[4] D. E. Low, "New approach to transportation systems modeling," Traffic Quarterly, vol. 26, no. 3, 1972.

[5] O. Tamin and L. Willumsen, "Transport demand model estimation from traffic counts," Transportation, vol. 16, no. 1, pp. 3-26, 1989.

[6] M.-H. Wang, S. D. Schrock, N. Vander Broek, and T. Mulinazzi, "Estimating dynamic origin-destination data and travel demand using cell phone network data," International Journal of Intelligent Transportation Systems Research, vol. 11, no. 2, pp. 76-86, 2013.

[7] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Time-evolving OD matrix estimation using high-speed GPS data streams," Expert Systems with Applications, vol. 44, pp. 275-288, 2016.

[8] W. K. V. Chan and C. Hsu, "Service scaling on hyper-networks," Service Science, vol. 1, no. 1, pp. 17-31, 2009.

[9] W. K. V. Chan and C. Hsu, "How hyper-network analysis helps understand human networks?," Service Science, vol. 2, no. 4, pp. 270-280, 2010.

[10] W. K. V. Chan and C. Hsu, "Service value networks: Humans hypernetwork to cocreate value," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 42, no. 4, pp. 802-813, 2012.

[11] W. K. V. Chan and C. Hsu, "When human networks collide: the degree distributions of hyper-networks," IIE Transactions, vol. 47, no. 9, pp. 929-942, 2015/09/02 2015.

[12] R. Borndörfer, M. Grötschel, and M. E. Pfetsch, "A column-generation approach to line planning in public transport," Transportation Science, vol. 41, no. 1, pp. 123-132, 2007.

[13] T. Li, "A demand estimator based on a nested logit model," Transportation Science, vol. 51, no. 3, pp. 918-930, 2017.

[14] Z. Cao et al., "Incorporating Human Movement Data to Improve Epidemiological Estimates for 2019-nCoV," medRxiv, 2020.

[15] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," Foundations of Computational Mathematics, vol. 9, no. 6, p. 717, 2009.

[16] H. Zhou, D. Zhang, and K. Xie, "Accurate traffic matrix completion based on multi-Gaussian models," Computer Communications, vol. 102, pp. 165-176, 2017.

[17] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," The Journal of Machine Learning Research, vol. 17, no. 1, pp. 2909-2913, 2016.

[18] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," Journal of Control and Decision, vol. 5, no. 1, pp. 42-60, 2018.

[19] A. Tavassoli, A. Alsger, M. Hickman, and M. Mesbah, "How close the models are to the reality? Comparison of transit origin-destination estimates with automatic fare collection data," in Proc. 38th Australas. Transp. Res. Forum (ATRF), 2016, pp. 1-15.

[20] X. Liu, P. Van Hentenryck, and X. Zhao, "Optimization Models for Estimating Transit Network Origin-Destination Flows with AVL/APC Data," arXiv preprint arXiv:1911.05777, 2019.