

# CITY PATH TOMOGRAPHY: RECONSTRUCTING SQUARE ROAD NETWORK FROM ARTIFICIAL USERS MOBILE PHONE DATA

## TOMOGRAFÍA DE TRAYECTORIAS: RECONSTRUYENDO UNA RED CUADRADA DE CALLES A PARTIR DE USUARIOS ARTIFICIALES DE TELEFONÍA

A. RODRÍGUEZ AND A. LAGE-CASTELLANOS<sup>†</sup>

Center for Complex Systems, Faculty of Physics, University of Havana, ale.lage@gmail.com<sup>†</sup>

<sup>†</sup> corresponding author

Recibido 30/10/2023; Aceptado 14/02/2023

Population mobility can be studied readily and cheaply using cellphone data, since people's mobility can be approximately mapped into tower-mobile registries. We model people moving in a grid-like city, where edges of the grid are weighted and paths are chosen according to overall weights between origin and destination. Cellphone users leave sparse signals in random nodes of the grid as they move by, mimicking the type of data collected from the tower-cellphone interactions. From this noisy data we seek to build a model of the city, *i.e.* to predict probabilities of paths from origin to destination. We focus on the simplest case where users move along shortest paths (no loops, no going backwards). In this simplified setting, we are able to infer the underlying weights of the edges (akin to road transitivity) with an inverse statistical mechanic model.

La movilidad de la población se puede estudiar de manera fácil y económica utilizando datos de la telefonía celular, mapeando la movilidad de las personas en registros de torres móviles. Se modela el movimiento de las personas en una ciudad en forma de red cuadrada, donde los nodos y sus conexiones se ponderan y las rutas se eligen de acuerdo con los pesos generales entre el origen y el destino. Los usuarios dejan señales dispersas en nodos de la red a medida que avanzan, imitando el tipo de datos recopilados de las interacciones entre la torre y el teléfono celular. Utilizando estos datos buscamos construir un modelo de la ciudad, es decir, predecir probabilidades de caminos desde el origen hasta el destino. Se estudia el caso donde los usuarios se mueven a lo largo de los caminos más cortos (sin lazos, sin retroceder). En esta configuración simplificada, podemos inferir los pesos del modelo (similar a la transitabilidad de la carretera) con un modelo físico estadístico inverso.

PACS: Statistical physics and nonlinear dynamics (física estadística y sistemas no lineales), 05.10.-a; computer modeling and simulation (modelado computacional y simulaciones), 07.05.Tp; Telecommunications (telecomunicaciones), 84.40.Ua.

### I. INTRODUCTION

Population size, concentration and mobility have been constantly growing during most of human history, and especially so in the last 200 years. These changes have impacted geography, demography and social dynamics in all too many ways, sometimes at a pace faster than the adaptability of the systems. Understanding human mobility patterns at city, country, and international scales is useful for decision-making related with urban planning, transport and dealing with infectious diseases [1].

Many different sources have been used to obtain information about the population mobility, a considerable part of those are expensive or slow. It is possible to find human mobility investigations carried out with the information of a national census [2] or bank notes [3]. Also GPS devices have been used [4]. GPS data provides accurate measures of position in outdoors with the drawback of less accuracy because of weak signals in indoor scenarios and the tendency of users to turn off the service when isn't needed due to battery consumption.

The use of telecommunication data allows researchers to perform these studies in a faster and cheaper way [6]. Large telecommunications companies, private applications, and network providers collect and store enormous quantities of data on users of their products and services [6]. A massive

number of phone calls can be processed and arranged in records that might include information of the user (usually anonymized), the position of the device or at least the radio base to what the user's device is connected and a time stamp [7].

The use of telecommunication data opens a new era in the human mobility studies [1]. This kind of data is now available to researchers in many countries, even in developing countries [8]. Due to its vast amount it is possible to improve the results in comparison with traditional sources as surveys and census and start new trends in human mobility fields.

In this paper we formalize the problem of understanding city connections (road usability or transitivity) from analyzing noisy and sparse cellphone data in an artificial city. We model the city as a weighted 2D grid describing the connections among parts of the city. Some papers already have taken graph based approaches. For instance in [10, 11] a transportation network graph is presented to calculate origin destination matrices using Markov models. Instead, we map our reconstruction problem, that we call **city path tomography**, as an inverse statistical mechanics model. We develop a procedure to infer weights for edges of the city from the sparse cellphone data using a gradient descent algorithm. The main limitation of the proposal is related to the capability of fast computation of the partition functions that we are able

to solve for 2D-grids, but remain a complicated issue for more heteromorphic cities.

Origin-destination (O-D) matrices are the standard information taken from mobility studies. Notwithstanding its relevance, it provides no clue on how people move between the given O-D points. City path tomography is somehow a complementary approach. It aims to produce a probabilistic model of the usage of different paths between given O-D points in a city.

The paper is organized as follows: section PATH TOMOGRAPHY describe the city and its connection to the artificial paths of users. In section MAX-LIKELIHOOD we discuss an efficient procedure to infer the weights of the city based on the incomplete path data. In section RESULTS we apply this method to some artificial datasets and discuss its performance. Finally CONCLUSIONS are drawn.

## II. PATH TOMOGRAPHY

We consider a simplified version of city as a squared lattice graph  $C = (V, E, W)$ , a 2D grid of weighted nodes and/or edges. Let  $L$  be the side of the square city, we label the nodes in the city as  $V = 1, \dots, L^2$ , and define paths, from start  $h_0$  to end  $h_n$  as a sequence of contiguous edges in the graph:

$$\mathcal{P} = \{h_0, h_1, \dots, h_{n-1}, h_n\} \text{ where } \forall_i (h_i, h_{i+1}) \in E(C). \quad (1)$$

Furthermore, we will only consider paths that take the traveler closer to destination at every step, for example if  $h_n$  is above/(to the right) of node  $h_0$ , then  $h_{i+1}$  is always above or to the right of node  $h_i$  in the sequence. This also means that if  $h_0$  and  $h_n$  are  $p$  steps distant in the y-axis and  $q$  steps in the x-axis, then the total number of nodes visited in every non returning path is  $n = p + q$ , and the amount of such paths is equal to:

$$N(h_0, h_n) = \binom{n}{p} \quad (2)$$

In figure 1 is shown an example of this kind of 2D grid using  $L = 5$ .

In order to build a statistical model for the trajectories of users in the city, we define the Hamiltonian (cost function) as the total weight of a path as follows:

$$H(\mathcal{P}) = \sum_{i \in \mathcal{P} \setminus \{h_0 \cup h_n\}} a_i + \sum_{(ij) \in \mathcal{P}} \omega_{ij} \quad (3)$$

where the first sum runs over nodes weights while the second over the links weights.

Be noticed that the first and last node weight wasn't added in the Hamiltonian. The probability of a given path between points  $h$  and  $m$  in the grid is therefore given by

$$P(\mathcal{P}) = \frac{1}{Z_{hm}} \exp(-\beta C(\mathcal{P})) \quad (4)$$

where the normalization  $Z_{hm}$  ensures that, constrained to the same origin  $h$  and destination  $m$ , the set of all non returning paths  $\mathcal{P}(h, m)$  has probability 1:

$$\sum_{\mathcal{P} \in \mathcal{P}(h, m)} P(\mathcal{P}) = 1 \Rightarrow Z_{hm} = \sum_{\mathcal{P} \in \mathcal{P}(h, m)} \exp(-\beta C(\mathcal{P})). \quad (5)$$

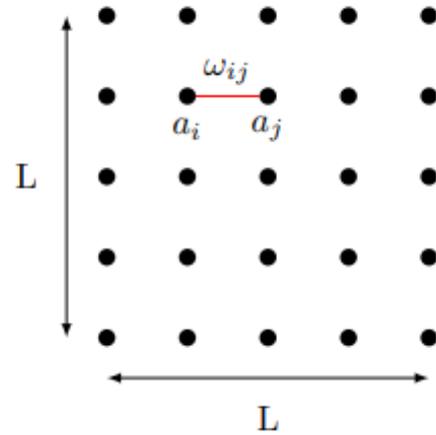


Figure 1. Example of 2D city grid, with nodes and links weighted.

### II.1. Telecom data

For the sake of our model, assume that towers exist at every node of our grid-like city, and that travelers with a mobile phone can leave registries of phone-tower communication in the nodes along their path. However, registry data from these paths is usually incomplete, and they usually consist on only but a few of the nodes visited. For instance, paths could look like

$$\mathcal{D} = \begin{cases} [95, 46, 48, 14, 42, 128, 116, 117] \\ [14, 61, 68] \\ [95, 128, 112, 117] \\ \dots \end{cases}$$

where each row correspond to a single user, and numbers correspond to tower ID's. As you can see from the first and third lines, the amount of signals left from a trip could vary widely, even between same origin and destination. We break down every multi-tower path to its minimum informative value, which is a set of triplets

$$\left. \begin{array}{l} [95, 46, 117] \\ [95, 48, 117] \\ [95, 14, 117] \\ [95, 42, 117] \\ [95, 128, 117] \\ [95, 116, 117] \end{array} \right\} 1/6$$

$$[14, 61, 68] \quad \} \quad 1$$

$$\left. \begin{array}{l} [95, 128, 117] \\ [95, 112, 117] \end{array} \right\} 1/2$$

When a given path has more than one (say  $k$ ) intermediate points, each of its triplets is weighted as  $1/k$  to avoid an statistical bias on the path due to its length.

We consider the travelers decisions around the city to be given by the model (4), and we assume that given a path as (1), every visited node leaves a signal in the mobile-phone records with a probability  $\eta < 1$ .

The path tomography problem (PTP) that we are interested in is the following:

**Definition [PTP]:** given a large set of incomplete paths  $\mathcal{D}$  of users in the city, infer the set of weights in  $W$  that define model (4).

Although we can consider node-weighted or edge-weighted cities, or even both, it can be shown that the probabilistic model given by (4) is invariant to local transformations of the type

$$a_i \rightarrow a_i + \Delta \quad \forall_{j \in \partial j} \omega_{ij} \rightarrow \omega_{ij} - \frac{\Delta}{2}. \quad (6)$$

This implies that a model with edge-weights is equivalent to the more general model with both edge and node weights, since every node weight can be freely set to zero by taking  $\Delta_i = -a_i$ . However, we will keep the model in full generality.

We will proceed with the node-weight graph first, and extensions to edge-weighted graphs is immediate. Given that someone is known to have moved from (non-adjacent) nodes  $i$  to  $j$ , the probability that it has passed through node  $k$  is given by:

$$P(k|i, j) = \frac{Z_{ik} e^{-\beta a_k} Z_{kj}}{Z_{ij}}. \quad (7)$$

### III. MAX-LIKELIHOOD INFERENCE

The model (4) considers probabilities of different paths between a given origin  $O$  and destination, but does not take into account (nor care) the frequency of each  $O - D$  pairs. Therefore we build a likelihood not in terms of full probabilities  $p(i, k, j) = P(k|i, j)P(O = i, D = j)$ , but rather in terms of the conditionals (7):

$$L(\mathcal{D}) = \sum_{(ikj) \in \mathcal{D}} \log P(k|i, j) \quad (8)$$

$$= \sum_{(ikj) \in \mathcal{D}} -\beta a_k + \log Z_{ik} + \log Z_{kj} - \log Z_{ij} \quad (9)$$

If  $n_{ij}^O$ ,  $n_{ij}^D$  and  $n_{ij}^m$  are the number of elements in  $\mathcal{D}$  of the type  $(i, j, *)$ ,  $(*, i, j)$  and  $(i, *, j)$  respectively, then we define the  $n_{ij} = n_{ij}^O + n_{ij}^D - n_{ij}^m$  and

$$L(\mathcal{D}) = \sum_{(ij)} n_{ij} \log Z_{ij} - \beta \sum_k m_k a_k \quad (10)$$

where all information from the data is subsumed in the coefficients  $n_{ij}$ . Maximization of the log-likelihood is achieved when its derivatives with respect to the cost function parameters is zero.

Considering that weight  $a_k$  appears in the exponential of paths that contain  $k$  as an intermediate point, we have:

$$\frac{1}{\beta} \frac{\partial L(\mathcal{D})}{\partial a_k} = - \sum_{i-j \ni k} n_{ij} \left( \frac{1}{Z_{ij}} Z_{ik} e^{-\beta a_k} Z_{kj} \right) - m_k \quad (11)$$

The gradient respect to the edges weights  $w_{kk'j}$  is:

$$\frac{1}{\beta} \frac{\partial L(\mathcal{D})}{\partial w_{kk'}} = - \sum_{i-j \ni (k, k')} n_{ij} \times \left( \frac{1}{Z_{ij}} Z_{ik} e^{-\beta(a_k + w_{kk'} + a_{k'})} Z_{k'j} \right) \quad (12)$$

A fast computation of this gradient, in order to implement gradient descent method, requires a clever way to evaluate all the  $Z_{ij}$  functions for every pair of possible origin and destination in the city.

#### III.1. Gradient fast computation

As explained in the previous section from a data set and a graph we aim to find the weights that maximize the log-likelihood expressed in (8). Then, the log-likelihood gradient should be computed respect to each type of weight used, let say nodes (11) and links (12).

The first step from the gradient computation is to extract the information from the data. This information is condensed in the values of  $n_{ij}$  and  $m_i$  as explained in the text before (10).

The computation of the gradient in terms of a node weight in (11) (link weight in (12)), requires a sum over all  $O - D$  pairs that include that node (link) as a passing point of at least one non returning path from  $O$  to  $D$ . Considering that the city has  $n$  nodes, the number of  $O - D$  pairs grows as  $n^2$ , and those sums are not extremely large. The list of  $O - D$ 's corresponding to every parameter to be extremized, is computed only once at the beginning.

Starting from a random initialization of the weights we apply a gradient descent as is described in algorithm 1.

In the algorithm, these quantities are computed at initialization time before the epoch loop and do not change over the gradient steps:

- $\forall_{i,j} n_{ij}$  and  $\forall_k m_k$  coefficients computed from data.
- possible\_nodes\_OD: The set of possibles origin-destination for paths including each node.
- possible\_links\_OD: The set of possibles origin-destination for paths including each link.
- Initial values of the weights  $\omega^0$  and/or  $a^0$ , chosen randomly.

Algorithm 1. Gradient descent for city path tomography

**Input:**  $n, m, \omega^0, a^0$

**Output:**  $\omega, a$

**foreach** epoch **do**

/\* Repeat until gradient convergence \*/

$Z = \text{Calc\_Part\_Func}(\omega^i, a^i)$  /\*  $i$ : epoch index \*/

**for**  $k$  in nodes **do**

temp=0

**for**  $i-j$  in possible\_nodes\_OD[ $k$ ] **do**

temp +=  $\frac{n_{ij}}{Z_{ij}} Z_{ik} e^{-\beta a_k} Z_{kj}$  /\* Summation term in equation (11) \*/

**end**

gradient\_a[ $k$ ] =  $-1 \times \text{temp} - m[k]$

**end**

/\* At this point log-likelihood gradient respect to  $a$  was computed \*/

**for** ( $k, k'$ ) in links **do**

temp=0

**for**  $i-j$  in possible\_links\_OD( $k, k'$ ) **do**

temp +=  $\frac{n_{ij}}{Z_{ij}} Z_{ik} e^{-\beta(a_k + \omega_{kk'} + a_{k'})} Z_{k'j}$  /\* Summation term in equation (12) \*/

**end**

gradient\_omega[( $k, k'$ )] = temp

**end**

/\* At this point log-likelihood gradient respect to  $\omega$  was computed \*/

Update weights:  $a^{i+1} = a^i + \text{learning\_rate} \times \text{gradient\_a}$

$\omega^{i+1} = \omega^i + \text{learning\_rate} \times \text{gradient\_omega}$

/\* Alternatively, an adaptative gradient descent algorithm can be use to compute  $a_{i+1}$  and  $\omega_{i+1}$  \*/

**end**

However, changing the weights  $a$  and  $\omega$  do affect the partition functions used in the computation, and it is carried by the function  $\text{Calc\_Part\_Func}(\omega^i, a^i)$  at the beginning every loop of the gradient descent algorithm. This procedure is described in algorithm 2.

Updating the partition functions is the time consuming step in this algorithm. Generally speaking, partition functions are hard, since they typically imply sums over a combinatorial number of states in a model. However, the regular grid-like geometry of the artificial city used and the assumption that only non-returning paths are considered, allow for an efficient computation of the partition function. In particular, the partition function for an origin-destination pair  $i - j$  can be split into the partition functions of a decomposition of the paths,  $i - k$  and  $k - j$  for example. This leads to:

$$Z_{ij} = \sum_{k \in S} Z_{ik} \times e^{-\beta a_k} \times Z_{kj} \quad (13)$$

in this equation  $S$  is a set of points with the same city block distance to  $i$  and the same city block distance to  $j$ . Fig. 2 shows a possible selection of  $k$  values to create a set  $S$  and the areas of the paths which each partition function in the summation represent.

This results brings two interesting facts. Considering the set  $S$  with points only one step away from  $j$  and a fixed number of steps away for  $i$ , any value  $Z_{ij}$  can be calculated using:

$$Z_{ij} = Z_{ij'} \times e^{-\beta(a_{j'} + \omega_{j'i})} + Z_{ij''} \times e^{-\beta(a_{j''} + \omega_{j''i})} \quad (14)$$

where  $j'$  and  $j''$  are the nodes closer to  $j$  in the directions to approach to  $i$ . With this in mind, recursivity can be used. Note that in (14)  $Z_{ij'}$  and  $Z_{ij''}$  are unknown but can be computed with the same idea. Using this recursive strategy implies that due to the calculation of  $Z_{ij}$  are computed partition functions with origin in  $i$  and destination each time closer to  $i$ . Then, the calculation of  $Z_{ij}$  leads to the computation of every  $Z_{ij'}$ , where  $\gamma$  is every node in the rectangle delimited by  $i$  and  $j$  in the city grid.

Algorithm 2. Procedure to calculate partition functions

**procedure**  $\text{Calc\_Part\_Func}(\omega^i, a^i)$

$Z$  initialization

**for**  $i$  in  $V \setminus \{(L-1) \times L + 1\} \cup L^2$  **do**

$Z_{i[(L-1) \times L + 1]} = \text{Calc\_Z}_{ij}(i, (L-1) \times L + 1, a, \omega)$

$Z_{iL^2} = \text{Calc\_Z}_{ij}(i, L^2, a, \omega)$

**end**

$Z_{[(L-1) \times L + 1]L^2} = \text{Calc\_Z}_{ij}((L-1) \times L + 1, L^2, a, \omega)$

/\* and  $\text{Calc\_Z}_{ij}(i, j, a, \omega)$  is described next \*/

**procedure**  $\text{Calc\_Z}_{ij}(i, j, a, \omega)$

**if**  $Z_{ij}$  is already in  $Z$  **then**

return  $Z_{ij}$

**else**

$Z_{ji} = 1$

return  $Z_{ij}$

$up, down, left, right = \text{calculate\_direction}(i, j)$  /\* Each value is true if  $i$  is in that direction respect  $j$  \*/

$Z_{ij} = up \times e^{-\beta(a_{j\uparrow} + \omega_{j\uparrow i})} \times \text{Calc\_Z}_{ij}(i, j_{\uparrow}, a, \omega)$

+  $down \times e^{-\beta(a_{j\downarrow} + \omega_{j\downarrow i})} \times \text{Calc\_Z}_{ij}(i, j_{\downarrow}, a, \omega)$

+  $left \times e^{-\beta(a_{j\leftarrow} + \omega_{j\leftarrow i})} \times \text{Calc\_Z}_{ij}(i, j_{\leftarrow}, a, \omega)$

+  $right \times e^{-\beta(a_{j\rightarrow} + \omega_{j\rightarrow i})} \times \text{Calc\_Z}_{ij}(i, j_{\rightarrow}, a, \omega)$

$Z_{ji} = Z_{ij}$

return  $Z_{ij}$

/\* Notice that  $Z$  is a data structure and  $Z_{ij}$  are values stored inside  $Z$  using the adequate indexes \*/

**end**

This approach is more efficient that obtaining every partition function following equation (5). Instead of adding over every possible path the already calculated partition functions are used to avoid redundant computations. This reduction in calculations becomes more significant as city side grows. For instance, if a partition function is calculated for points distant  $q = K$  steps horizontally and  $p = K$  steps vertically, the brute force computation results in  $\sim 2^{2K}$  from an Stirling's approximation of (2). Exploiting the our recursive procedure

this results in  $\sim K^2$  operations.

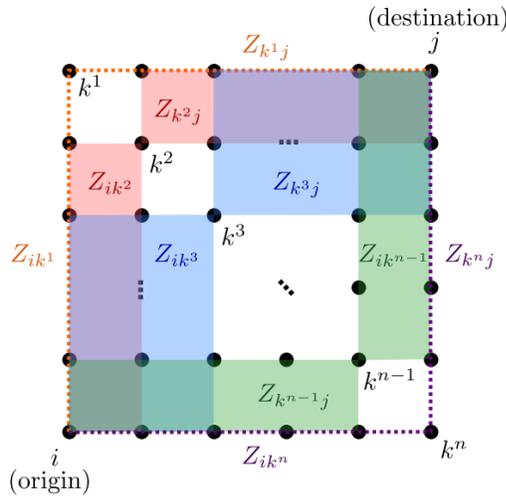


Figure 2. A possible selection of  $S$  and description of partition functions involved.

The final step in the gradient descent is the updating of the weights in the direction of the gradient in order to maximize the log likelihood. This can be done with a fixed learning rate like the stochastic learning rate or with an adaptative step, like Adam gradient descent algorithm or similar [12]. The gradient steps are carried up to when a given tolerance is achieved or a maximum number of loops are reached. Typically the tolerance is fixed as a small value for the norm of the gradient.

#### IV. RESULTS

We now show some experiments to test the precision of our inference methodology with synthetic data from an artificial city. We will consider a grid-like city of  $N = L \times L$  nodes ( $L = 12$ ) and a link weights extracted from a scaled Gaussian distribution

$$w_{i,j} = \beta \omega_{i,j}^0 \quad \text{where } \omega_{i,j}^0 \sim N(\mu = 0, \sigma^2 = 1).$$

We do not consider node weights, since they are equivalent to a gauge transformation of the links.

In artificial city it is possible to take several paths between each origin and destination. The probability of each path can be computed using (4) given a set of weights. From this artificial city we generate a data set of triplets  $\mathcal{D}$ . Each triplets is composed by an origin, an intermediate point and destination. The length of the set,  $|\mathcal{D}|$ , corresponds with the total of triplets. This set of triplets are similar to the data set of sparse trajectories of mobile phone registers after be split.

At this point, we have a set of paths for each origin destination pair and the probability to take each path. The probability to use  $k$  in a  $i - j$  path is easily obtainable summing over every  $i - j$  path that includes  $k$ . Then, we can pick origin-destination pairs and sample the data according the distribution created.

There are two main parameters to consider in order to perform a proper evaluation. The first is  $|\mathcal{D}|$ , the length of the set. If

our procedure is correct, the quality of our inference should grow towards perfection when the number of samples in  $\mathcal{D}$  is large. The second parameter is  $\beta$ , this value represents the inverse of the temperature, taking a direct effect over the shape of the distribution of the paths probabilities. For  $\beta = 0$  (infinite temperature) every path has equal probability with independence of the values of  $\omega$  and  $a$ . With an increase of  $\beta$  (temperature decrease) it is expected that the path with the lower Hamiltonian (3) becomes more probable relative to the others with same origin and destination.

Firstly, we explore the influence of  $|\mathcal{D}|$  over mean square error of the probabilities generated with the inferred weights. For that, we set  $\beta = 0$  and perform the procedure presented in the previous section to obtain a set of weights from a random ones.

Once the new weights are inferred we compute:

$$mse = \frac{1}{N} \sum_{(ikj) \in \mathcal{D}} \sqrt{(p_{r(ikj)})^2 - (p_{i(ikj)})^2} \quad (15)$$

where  $N$  is the total of  $ikj$  in the graphs and  $p_{r(ikj)}$  and  $p_{i(ikj)}$  stands for the real and inferred probability for the trio  $ikj$ , respectively.

Figure 3 shows the results of the logarithm of mean square error with the grow of  $|\mathcal{D}|$ . As expected the error decreases with an increases of length  $\mathcal{D}$ .

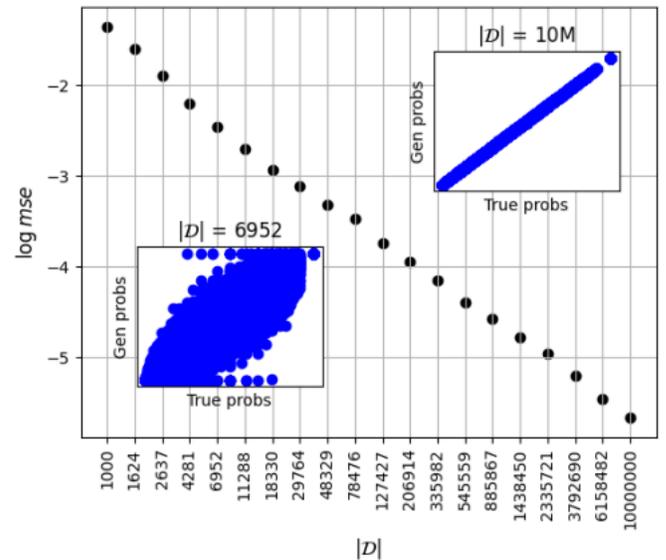


Figure 3. Log mean square error for several length of  $\mathcal{D}$ , in an homogeneous city,  $\beta = 0$ . True probabilities vs model generated probabilities for particular values of  $\mathcal{D}$  in inner plots.

In the inner plots of Fig. 3 it is shown the behavior of  $p_{r(ikj)}$  against  $p_{i(ikj)}$  for every  $ikj$ . Notice that with a bigger number of trios the distribution of dots became similar to the curve  $x = y$ , as expected. This is shown with more detail in Fig. 4, when also is considered the case with length of  $\mathcal{D}$  infinite. Notice, that to assume this case is needed to set  $n$  and  $m$  with the value of its convergence with the increase of  $|\mathcal{D}|$ .

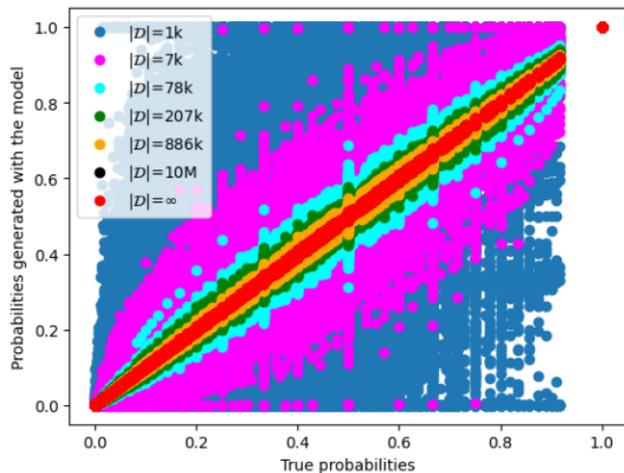


Figure 4. True probabilities against probabilities generated using model's parameters.

The analysis done in Fig. 3 can be extended to other values of  $\beta$  to prove that the same behavior is obtained in a non-homogeneous city. This is shown in Fig. 5, where using the same set of discrete  $|D|$  the values of the logarithm of mse are plotted from different values of  $\beta$ . Particularly,  $\beta = 0, \beta = 1, \beta = 2, \beta = 4$  and  $\beta = 8$ .

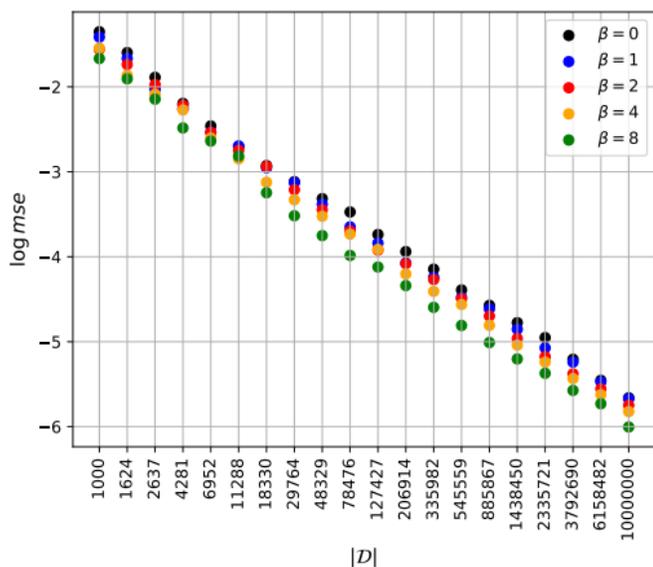


Figure 5. Mean square error varying length of  $\mathcal{D}$ , using different values of  $\beta$ .

## V. CONCLUSIONS

We introduce the City Path Tomography problem and show that it can be solved in a very simplified toy model of phone

users moving in a grid-like city. The likelihood of the observed data is maximized in an efficient way, thanks, mostly to the simplicity of the city and the assumption of users moving along non returning paths. As expected, the inference is more accurate as the amount of data grows.

The success of this methodology acts as a proof of concept. It is a first step to attempt the more challenging situation of realistic cities and travelers. The challenge in that case is two fold: first describe a model that is consistent with real human behavior in a city, and second, solve it (probably resorting to approximate stat mech methods).

**Acknowledgments** The research presented in this publication received funds from the Office of International Funds and Projects Management under the code PN223LH006-007.

## REFERENCES

- [1] F. Asgari, V. Gauthier, and M. Becker, arXiv: Social and Information Networks (2013).
- [2] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Vars-havsky, and C. Volinsky, *Comm. ACM* **56**, 74 (2013).
- [3] D. Brockmann, L. Hufnagel, and T. Geisel, *Nature* **439**, 462 (2006).
- [4] H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, *Environ. Urban Syst.* **36**, 131 (2012).
- [5] H. Huang, Y. Chen, and R. Weibel, *Transp. Res. Pt. C-Emerg. Technol.* **101**, 297 (2019).
- [6] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evsukoff, and M. C. González, *Transp. Res. Pt. C-Emerg. Technol.* **58**, 162 (2015).
- [7] C. Chen J. Ma, Y. Susilo, Y. Liu, and M. Wang, *Transp. Res. Pt. C-Emerg. Technol.* **68**, 285 (2016).
- [8] F. Calabrese, L. Ferrari, and V. D. Blondel, *ACM Comput. Surv.* **47**, 25 (2014).
- [9] K. G. Shin, X. Ju, Z. Chen, and X. Hu, *IEEE Wirel. Commun.* **19**, 30 (2012).
- [10] A. Tesselkin and V. Khabarov, *Procedia Eng.* **178**, 107 (2017).
- [11] M. Pourmoradnasseri, K. Khoshkhah, A. Lind, and A. Hadachi, 2019 International Conference on Wireless and Mobile Computing, Networkin and Communications (WiMob), 1 (2019).
- [12] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747 (2016).

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0, <http://creativecommons.org/licenses/by-nc/4.0>) license.

