

TELL ME WHERE YOU ARE AND I TELL YOU WHERE YOU ARE GOING: ESTIMATION OF DYNAMIC MOBILITY GRAPHS

Marcelo Fiori, Pablo Musé

Universidad de la República, Uruguay

Mariano Tepper, Guillermo Sapiro

Duke University, NC, USA

ABSTRACT

The interest in problems related to graph inference has been increasing significantly during the last decade. However, the vast majority of the problems addressed are either static, or systems where changes in one node are immediately reflected in other nodes. In this paper we address the problem of mobility graph estimation, when the available dataset has an asynchronous and time-variant nature. We present a formulation for this problem consisting on an optimization of a cost function having a fitting term to explain the observations with the dynamics of the system, and a sparsity promoting penalty term, in order to select the paths actually used. The formulation is tested on two publicly available real datasets on US aviation and NY taxi traffic, showing the importance of the problem and the applicability of the proposed framework.

Index Terms— Graph inference, Asynchronous Dynamic Mobility Graphs.

1. INTRODUCTION

The significant growth of available data, both in quantity and diversity, has motivated an increased interest in problems related with graph inference or network estimation, from gene regulatory networks and brain connectivity graphs using fMRI data to social networks and micro-blog data.

Several graph inference algorithms have been recently introduced [1–6], showing that sparse models provide useful formulation for addressing the problem, and introducing a significant number of applications. For instance, in [7] the authors study the problem of inferring the “online news” network topology and dynamics from the spread of blog posts and news articles. Another common application is the estimation of brain connectivity from fMRI data [8], or gene regulatory networks from micro-array data [9].

However, all these works address the network inference problem either for static graphs, or for graphs that exhibit very particular dynamics, i.e., where the interactions between nodes are instantaneous, and once the information arrives to a certain node, it gets “infected,” meaning that the node cannot go back to its previous state. This is the case, for instance, of social networks or blogs and micro-blogs data.

We study the problem of estimating the mobility pattern of entities in a more general setting. Let us assume that we can count the number of entities at different sites along time. For example, we may know how many people are on each track of a subway station connecting several lines, at every time. The goal is to infer the general mobility pattern within the station, to infer how connections are taken by the passengers.

The main difference with the other graph inference problems previously mentioned is the timing aspect: in this problem, the time it takes to go from one site to another depends on the sites and is *unknown*. This simple modification adds a whole layer of complexity to the problem, rendering very challenging. To the best of our knowledge, this problem is here studied for the first time.

As described throughout this paper, this problem is extremely ill-conditioned in general, since there might be several ways to explain certain observations. Thus, the selection of the right type of regularizers plays a critical role, even more so than in other related formulations.

2. PROBLEM FORMULATION

Let us suppose that we are observing entities moving through different sites over time. Given n such sites, we observe (exactly or approximately) the number of entities in each site, at discretized time intervals $t = 1 \dots T$. Our goal is to infer, from this information, the mobility pattern of the entities.

This problem, although simple to describe, presents several difficulties. First, we cannot observe an entity while it is moving from one site to another. If an entity is traveling from site i to site j with a travel time d , we can only observe it in node i at time t , and then in node j at time $t + d$: in the interval $(t, t + d)$, the entity becomes unobservable. Additionally, these travel times are unknown, and they might depend on the path and also on the particular entity itself. On the other hand, each movement might have several possible explanations (i.e., there are many ways of traveling from site i to site j); these uncertainties make the problem ill-posed in general.

We will formally model the desired mobility pattern of the entities as a graph of transitions between n nodes, where each node corresponds to a site. We first represent the given/observed information as an $n \times T$ matrix \mathbf{U} , where the entry u_{it} contains the number of entities at node i during time

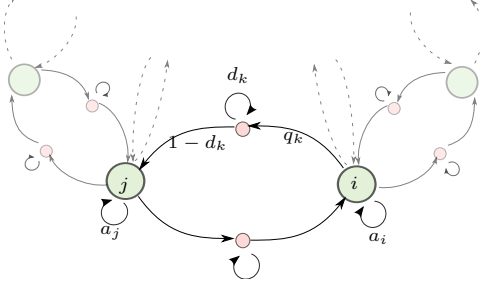


Fig. 1. Example graph of the formulation. The big nodes are the *original nodes*, where we can observe the number of entities at every time. The smaller nodes are the *transition nodes*, which represent the “in transit” state, and are not observable.

interval t . In order to capture the described mobility problem, we augment the graph with $n(n-1)$ extra nodes, that model the transition between every (ordered) pair of original nodes. Observe that each transition node is associated with a directed path (say, from node i to node j), and represents an “in transit” site where the entities virtually stay for the travel duration between node i and node j . Of course, the number of entities in each transition node is not directly observable. We refer to the observable nodes as *original* and the unobservable transition nodes as *transition* nodes.

Transitions from one node to another are modeled stochastically. An entity present in node i at a given time can either stay at the same site i with probability a_i , or choose an outgoing path k , connecting the node i with one of the remaining $n-1$ nodes, with probability q_k . The probability d_k of staying in the transition node k models the travel duration from i to j . See Figure 1 for a visual representation of these quantities.

Thus, for each transition node, linking original nodes i and j , we have the following associated unknowns: (1) the probability of going from i to j , (2) the probability of staying in the transition node, and (3) the amount of entities at the transition node at each time interval.

These unknowns are globally represented by a vector \mathbf{q} with $n(n-1)$ entries, containing the probabilities of going from one original node to another (i.e., the probability of going from one original node to the transition node associated with that path); a vector \mathbf{d} with $n(n-1)$ entries, containing the probabilities of staying in the corresponding transition node; and a $n(n-1) \times T$ matrix \mathbf{V} containing the number of entities at each transition node (or path) at each time.

Each one of the $n(n-1)$ entries of these unknowns (\mathbf{q} , \mathbf{d} and the rows of \mathbf{V}) are associated with an ordered pair of the original nodes. We order these variables according to the destination node first, and then according to the source node (i.e., co-lexicographic order).

Additionally, we have an n dimensional vector \mathbf{a} , with the probability of staying at each original node.

In order to write the equations, let us begin by analyzing

a single node in particular at certain time.

The number of entities at node i at a given time $t+1$ is equal to the number of entities that were at node i at time t and stayed, plus the number of entities that were traveling towards node i and arrived at time $t+1$. This is

$$u_{i,t+1} = a_i u_{i,t} + \sum_k (1 - d_k) v_{k,t},$$

where a_i is the probability of staying (fraction of entities staying) at node i , and k indexes the transition node associated with the path $j \rightarrow i$. Hence, $v_{k,t}$ is the number of entities on their way from i to j at time t , and $1 - d_k$ is the probability of leaving the transition node and arriving to j .

This can be re-written for all nodes and for all time intervals, in the matrix form

$$\mathbf{U}_2 = \mathbf{A}\mathbf{U}_1 + \mathbf{M}(\mathbf{I} - \mathbf{D})\mathbf{V}_1, \quad (1)$$

where \mathbf{A} and \mathbf{D} are diagonal matrices with the elements of \mathbf{a} and \mathbf{d} in the diagonal, respectively, \mathbf{I} is the identity matrix, \mathbf{U}_2 and \mathbf{U}_1 are the $n \times (T-1)$ matrices formed by taking the original matrix \mathbf{U} and removing the first and the last column respectively, and \mathbf{M} is the $n \times n(n-1)$ matrix having $n-1$ ones per row. In the first row, the ones are in the first block of $n-1$ columns, in the second row they are in the following block of $n-1$ columns and so on. This way, by left multiplying by \mathbf{M} we are adding up through all the transition nodes with the same destination node.

In a similar way, we can describe the number of entities at every transition node at each time, and an equation similar to (1) can be stated

$$\mathbf{V}_2 = \mathbf{D}\mathbf{V}_1 + \mathbf{QPM}^T\mathbf{U}_1, \quad (2)$$

where \mathbf{Q} is a diagonal matrix with the elements of \mathbf{q} in the diagonal, and \mathbf{P} is the permutation matrix transforming from co-lexicographic to lexicographic order.

We can now introduce a fitting function, to be minimized, that penalizes deviations from the model given by (1) and (2). A suitable choice is, for example,

$$f(\mathbf{a}, \mathbf{q}, \mathbf{d}, \mathbf{V}) = \frac{1}{2} \|\mathbf{A}\mathbf{U}_1 + \mathbf{M}(\mathbf{I} - \mathbf{D})\mathbf{V}_1 - \mathbf{U}_2\|_F^2 + \frac{1}{2} \|\mathbf{D}\mathbf{V}_1 + \mathbf{QPM}^T\mathbf{U}_1 - \mathbf{V}_2\|_F^2. \quad (3)$$

Let us now add several constraints and priors which help to better solve this ill-posed problem.

First, the variables \mathbf{a} , \mathbf{q} , and \mathbf{d} are vectors representing probabilities, so each entry of these three unknowns must be in $[0, 1]$. The matrix \mathbf{V} contains the number of entities in each path, so it must be $v_{k,t} \geq 0$ for all k, t . In addition, for each node the probabilities associated with outgoing edges should add up one. For the original nodes, the probability of staying in the node a_i plus the probability of leaving to any path (given by q_k) should add up one. This can be written,

for all the nodes, as: $\mathbf{a} + \mathbf{M}\mathbf{P}\mathbf{q} = \mathbf{1}$, where $\mathbf{1}$ is a vector of ones. Here the vector \mathbf{a} can be written in terms of \mathbf{q} as $\mathbf{a} = \mathbf{1} - \mathbf{M}\mathbf{P}\mathbf{q}$, and therefore this constraint can be directly incorporated in the formulation.

Finally, we assume that the number of entities is constant over time, and therefore each column of \mathbf{U} plus the corresponding column of \mathbf{V} must be constant. This assumption can be easily removed by adding or subtracting the number of entities entering or leaving the system at each time; it is reasonable to assume that this number can be observed.

Note that the function f in Equation (3) is biconvex (the variables \mathbf{d} and \mathbf{V} are multiplying each other), and all the restrictions are convex sets. A common way to address the minimization of a biconvex function is to alternatively fix one variable and solve for the other (\mathbf{V} and (\mathbf{q}, \mathbf{d}) in this case).

Additionally, as mentioned above, some solutions are indistinguishable from each other, making the problem extremely ill-posed. Therefore, additional prior information is still needed to regularize this inverse problem.

A very reasonable assumption is that not every original node is connected to all the others (meaning that the graph is not complete), but on the contrary, that most of the paths are unused (non-existent in the physical world). This can be incorporated to the formulation by means of a sparsity promoting norm, such as ℓ_0 or ℓ_1 , used as a penalty term for the unknowns we want to make sparse.

Moreover, observe that if a certain entry of \mathbf{q} is non-zero, this means that the corresponding path is active. Thus, the corresponding entry of \mathbf{d} and row of \mathbf{V} should be also active. On the contrary, if a path is inactive, then all the corresponding entries of \mathbf{q} , \mathbf{d} , and \mathbf{V} should be zero. This suggest a group lasso type of approach [11], which is known to promote either active or inactive groups. In this case, we consider $n(n-1)$ groups (one per transition node), each one formed by an entry of \mathbf{q} , of \mathbf{d} , and the corresponding row of \mathbf{V} .

This group lasso approach has both effects at the same time: promoting a sparse number of active paths, and enforcing that if a path is inactive, then all the corresponding variables should be zero. Then, the resulting formulation is

$$\min_{\substack{\mathbf{q} \geq 0, \mathbf{d} \geq 0, \mathbf{V} \geq 0 \\ \mathbf{U}^T \mathbf{1} + \mathbf{V}^T \mathbf{1} = N \mathbf{1}}} f(\mathbf{q}, \mathbf{d}, \mathbf{V}) + \lambda \sum_{k=1}^{n(n-1)} \|\mathbf{w}_k\|_2, \quad (4)$$

where N is the total number of entities, λ is a parameter controlling the sparsity of the solution, and \mathbf{w}_k is the k -th row of the matrix \mathbf{W} , formed by taking the k -th row of matrix \mathbf{V} concatenated with q_k and d_k .

The second term is the ℓ_1 norm of the vector containing the norms of the rows of \mathbf{W} , which is the convex relaxation of the ℓ_0 pseudo-norm of that vector.

Since this problem is non-convex, a good initialization of the optimization is crucial. In the next section, we provide an independent formulation to estimate \mathbf{V} , and use it as initialization for the optimization of (4).

3. OPTIMIZATION

Since f is biconvex, we proceed by alternate minimization for \mathbf{B} and (\mathbf{q}, \mathbf{d}) , with the other variable fixed.

For each subproblem, the constraint $\mathbf{U}^T \mathbf{1} + \mathbf{V}^T \mathbf{1} = N \mathbf{1}$ is added to the optimization by means of the augmented Lagrangian method, which consists in adding a smooth term (or two terms that can be combined into one, as done here) with a new auxiliary variable \mathbf{h} . For instance, when \mathbf{q} and \mathbf{d} are fixed, the problem to solve is

$$\min_{\mathbf{V} \geq 0} f(\mathbf{q}, \mathbf{d}, \mathbf{V}) + \lambda \sum_{k=1}^{n(n-1)} \|\mathbf{w}_k\|_2 + \frac{\mu}{2} \|\mathbf{U}^T \mathbf{1} + \mathbf{V}^T \mathbf{1} - N \mathbf{1} + \mathbf{h}\|_F^2, \quad (5)$$

where μ is a parameter which does not affect the convergence, and it was set to $\mu = 2$ in the experiments.

The procedure is iterative. At each iteration l , the objective function in (5) is minimized, and then the auxiliary variable is updated as $\mathbf{h}^{l+1} = \mathbf{h}^l + \mathbf{U}^T \mathbf{1} + \mathbf{V}^T \mathbf{1} - N \mathbf{1}$.

At each iteration, the minimization of the non-smooth function in (5) is solved by standard techniques, consisting of gradient descent combined with the vector soft-thresholding operator (for more details, see [14, 15]).

The optimization for (\mathbf{q}, \mathbf{d}) with \mathbf{V} fixed is analogous.

4. EXPERIMENTAL RESULTS

We now present experimental results with two publicly available datasets containing real transportation data.

Airports routes

Suppose that we are given the number of airplanes at a given set of airports at every time, is it possible to recover the airplane routes and the trips durations? As mentioned in the introduction, this is an interesting and challenging problem.

We apply the proposed formulation to the analysis of a dataset containing all the US internal flights from 1987 to 2014. We analyze the most recent available month (which is November 2014), then select 11 important airports in the US, and finally considered all the airplanes which have departed from or landed in any of those airports during that month.

Since every airplane has a unique identifier (the *Tail Number*), it can be tracked to determine at which airport it was at every time, or if it was flying (i.e., present in no airport). With this information, and dividing the complete month into 15 minutes intervals, we construct the matrix \mathbf{U} and the ground truth matrix \mathbf{V} , and then infer the transitions with the presented formulation.

The inferred graph is very similar to the ground truth graph at first sight. Although some edges disappeared, and some other routes appeared, the general topology of the network is recovered. Also, it can be observed that the main circuits also have high probability in the estimated network: for instance, flights from LAX to JFK and the way back.

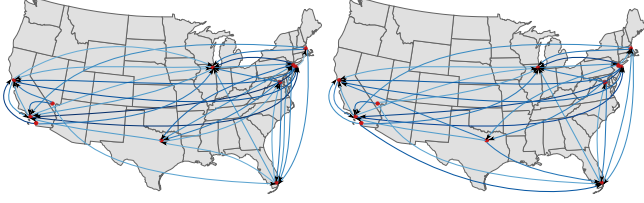


Fig. 2. Results of the graph estimation for the airplane routes dataset. Left: ground truth. Right: inferred network. The color indicates the probability q_k of the corresponding path.

Since the two airports in New York are very close to each other in the figure, it cannot be observed, but there are no edges between LGA and JFK in the estimated graph (and of course neither in the ground truth graph). The same happens with the airports of San Diego and Los Angeles.

In terms of numerical results, we computed the difference (in absolute value) of the ground truth and estimated probabilities. The average and standard deviation of these differences is 0.023 and 0.029 respectively.

New York taxis

A very interesting dataset has been recently released, containing all the trips of New York yellow taxis during 2013. The dataset consists of millions of records, each one corresponding to a certain trip. Each record contains the vehicle identifier, the pickup and dropoff date and times, trip time in seconds, and the GPS coordinates of the pickup and dropoff locations. We chose to limit the data to the trips within Manhattan, and we divided it into 10 regions.

The experiments described below were carried out with the data corresponding to one day, from 8:00am to 9:00pm, dividing the time interval into intervals of one minute.

Let us suppose that we can observe the taxis location only when they are picking up or dropping off a passenger, which is actually the available data, as described above. However, we will only make use of this information, forgetting which point is a pick up or a drop off, the taxi identifier, and the trip time. All this information will be used as ground truth, in order to compare the results obtained by estimating the mobility pattern solely from the described data: time and GPS coordinates for the pickup and drop off locations.

For each time interval, we compute the number of taxis picking up or dropping off a passenger at every zone, in order to construct the matrix \mathbf{U} , and then we run the algorithm.

The results are shown in Figure 3. The ground truth graph is computed with the complete dataset by simply counting the number of trips between regions, and then computing directly the probability. The inferred mobility pattern is very similar to the ground truth. The estimated graph has some extra arrows, but with the exception of one of them, the width of the arrows is the thinnest in the figure, which means that the associated probability is not significant.

In some cases where a path is present in both graphs, the

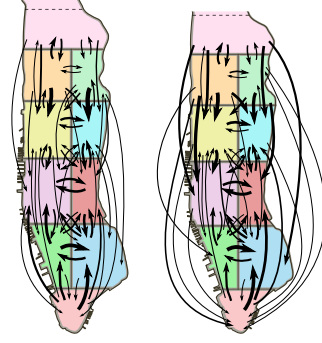


Fig. 3. Results of the graph inference for the New York taxis dataset. Left: ground truth. Right: inferred graph. The width of the arrows is proportional to the probability q_k of the corresponding path.

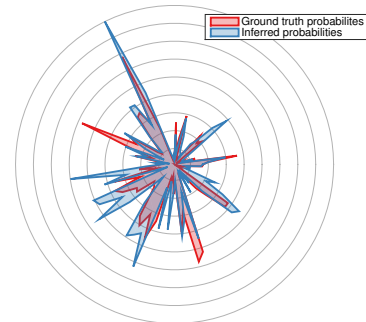


Fig. 4. Radar plot of the probabilities (vector \mathbf{q}) for the New York traffic dataset.

width of the arrow may differ. However, the general “large scale” pattern is the same: most of the trips are from one region to an adjacent region (observe that the formulation does not include any geographical information, nor any relation between the regions).

In order to compare directly the results, we also include a radar plot of the ground truth and inferred probabilities, shown in Figure 4, and the numerical results as in the previous example: in this case, the average and standard deviation of the errors is 0.0519 and 0.0470.

5. CONCLUSION

In this work we introduced a framework to address the problem of mobility graph estimation, when only counting information on some nodes is available, the movements are asynchronous, and the time it takes to an entity to go from one site to another depends on the origin and destination. Due to these particularities, this is a very challenging problem.

We solved the proposed optimization problem for two real datasets. The results show that the general topology of the mobility pattern can be recovered, and therefore the system can be analyzed from this inferred network. This suggests that this is a promising line of work for a very interesting and challenging problem, which can be further improved.

6. REFERENCES

- [1] O. Banerjee, L. El Ghaoui, and A. D'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.
- [2] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the PC-Algorithm," *Journal of Machine Learning Research*, vol. 8, pp. 613–636, 2007.
- [3] J. Peng, P. Wang, N. Zhou, and J. Zhu, "Partial correlation estimation by joint sparse regression models," *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 735–746, June 2009.
- [4] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, Feb. 2007.
- [5] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the Lasso," *The Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, June 2006.
- [6] P. Loh and M. Wainwright, "Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses," in *Advances in Neural Information Processing Systems 25*, pp. 2096–2104, 2012.
- [7] B. Baingana, G. Mateos, and G.B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 8, no. 4, pp. 563–575, Aug 2014.
- [8] Gal Varoquaux, Alexandre Gramfort, Jean-Baptiste Poline, and Bertrand Thirion, "Brain covariance selection: better individual functional connectivity models using population prior," in *NIPS'10*, 2010, pp. 2334–2342.
- [9] Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu, "Partial correlation estimation by joint sparse regression models," *Journal of the American Statistical Association*, vol. 104, no. 486, 2009.
- [10] John P Formby, W James Smith, and Buhong Zheng, "Mobility measurement, transition matrices and statistical inference," *Journal of Econometrics*, vol. 120, no. 1, pp. 181–205, 2004.
- [11] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 1, pp. 49–67, 2006.
- [12] David L Donoho and Jared Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9446–9451, 2005.
- [13] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.
- [14] Pablo Sprechmann, Ignacio Ramírez, Guillermo Sapiro, and Yonina C Eldar, "C-hilasso: A collaborative hierarchical sparse modeling framework," *Signal Processing, IEEE Transactions on*, vol. 59, no. 9, pp. 4183–4198, 2011.
- [15] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.
- [16] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2nd edition, 2006.