Inferring Networks From Marginals Using Iterative Proportional Fitting

Serina Chang*,[1], Zhaonan Qu*,[2], Jure Leskovec[1], Johan Ugander[3]

[1] Department of Computer Science, Stanford University
[2] Department of Economics, Stanford University
[3] Department of Management Science & Engineering, Stanford University
* These authors contributed equally to this work.

Abstract

When collecting dynamic network data, it is often more admissible—either due to privacy concerns or real-time feasibility—to collect the marginals of a network than its time-varying interiors. This reality, arising in classic and recent studies of human mobility, transportation, and migration networks, results in a natural and increasingly common network inference problem, where the goal is to infer a dynamic network from its 3-dimensional marginals, i.e., its time-varying rows, time-varying columns, and time-aggregated interaction matrix. Prior works on this problem have repurposed the popular iterative proportional fitting (IPF) procedure, also widely known as Sinkhorn's algorithm, to infer dynamic networks from aggregate data; these resulting networks have been employed in several downstream tasks, including building tools for COVID-19 policymakers. Despite these high-impact applications, the behavior and assumptions of using IPF in this setting are not well understood. In this work, we fill in the missing theory, rigorously motivating the use of IPF for this network inference problem. Our main contribution is a statistical justification of the minimization principle of IPF for network inference, by formulating an instructive, generative network model whose maximum likelihood objective is dual to the Kullback-Leibler divergence minimization problem implied by IPF. Conveniently, the marginal observations form the sufficient statistics of the network model, aligning with problem constraints. We also run computational experiments with real-world mobility data, to demonstrate the effectiveness of IPF to infer networks in practice and to show how our new methods of analysis make it possible to inspect previously unstated assumptions.

1 Introduction

Modern computing platforms, which enable the collection of large-scale but incomplete network data, have given rise to new network inference problems. In this work we study a central network inference problem: how to infer a dynamic network (or graph) from its 3-dimensional marginals, i.e., its time-varying rows, time-varying columns, and time-aggregated interaction matrix. This problem appears across domains, such as human mobility [1], traffic and transportation flow [2, 3], or migration [4], where it is often more feasible—either due to privacy concerns or real-time constraints—to collect a network's marginals than its time-varying interiors. A notable example appears in Chang et al. [1], where the authors sought to infer hourly mobility networks between neighborhoods and points-of-interest (POIs) from aggregated location data, which contained hourly total visits from each neighborhood, hourly total visits to each POI, and monthly estimates of visits from neighborhoods to POIs. These networks were integrated into epidemiological models to simulate the spread of COVID-19, leading to widely cited results in *Nature* [5–8]. The inferred networks were also used in other studies, including to build tools for public health officials [9], to study airbone transmission of COVID-19 [10], and to develop network-based interventions to reduce transmission [11].

S. Chang, Z. Qu, J. Leskovec, J. Ugander, Inferring Networks From Marginals Using Iterative Proportional Fitting (Extended Abstract). Presented at the Second Learning on Graphs Conference (LoG 2023), Virtual Event, November 27–30, 2023.

To solve the network inference problem, Chang et al. [1] repurposed the well-known iterative proportional fitting (IPF) procedure [12, 13], also known as Sinkhorn's algorithm [14, 15]. IPF is a classic algorithm that aims to scale the rows and columns of a given matrix X to match target marginals p and q. In Chang et al. [1], IPF demonstrates promising empirical results: the algorithm is computationally lightweight, allowing the authors to infer networks with billions of edges, and the IPF-inferred networks enable epidemiological models to accurately fit COVID-19 case curves.

Despite the empirical success of IPF for this problem, the *theoretical* basis of its use for network inference has been lacking. While it is well-known that IPF solves a Kullback-Leibler (KL) divergence minimization problem, a formal connection to statistical theory is limited. What are we implicitly assuming about the dynamic network and its observed marginals by using IPF to infer it? This work answers the question by formulating an instructive, generative network model whose parameters are properly estimated by IPF, since its maximum likelihood objective is dual to the KL divergence minimization problem solved by IPF (Theorem 1). Conveniently, the marginal observations form the sufficient statistics of this network model, aligning with the problem constraints. We conduct computational experiments with real-world mobility data and synthetic data, which confirm our analytical results and show how our new methods of analysis make it possible to inspect previously unstated assumptions. Our results provide much-needed theory to justify high-impact applications of IPF [1, 9–11] and to rigorously motivate future uses of IPF for this problem. Given the vast literature on IPF and Sinkhorn's algorithm, connecting network inference to IPF also opens up future avenues for research, creating a bridge from the graph learning community to decades of statistical theory.

2 Optimization principles of IPF for network inference

Background on IPF. Consider the following classic *matrix balancing problem* [12, 16]:

> Given positive vectors $p \in \mathbb{R}^m_{++}, q \in \mathbb{R}^n_{++}$ with $\sum p_i = \sum q_j$ and non-negative matrix $X \in \mathbb{R}^{m\times n}_{+}$, find positive diagonal matrices D^0, D^1 satisfying the conditions $D^0 X D^1 \cdot \mathbf{1}_n = p$ and $D^1 X^T D^0 \cdot \mathbf{1}_m = q$.

In network applications [1, 9], X is an aggregate or historical network, and our task is to infer an up-to-date network Y observing only its marginals p, q, and X, with the estimate $D^0 X D^1$. IPF learns the scaling factors d^0 and d^1, which are diagonals of D^0, D^1, by alternating between scaling the rows to match p, then scaling the columns to match q:

$$d_i^0(k+1) = \frac{p_i}{\sum_j X_{ij} d_j^1(k)}, \quad d_j^1(k+1) = \frac{q_j}{\sum_i X_{ij} d_i^0(k)}. \tag{1}$$

We denote by $M^{\mathrm{IPF}}(k) := D^0(k) X D^1(k)$ the scaled matrix after the k-th iteration. The convergence behavior depends on the problem structure: $(D^0(k), D^1(k))$ can converge to a solution of the matrix balancing problem; $(D^0(k), D^1(k))$ can diverge but $M^{\mathrm{IPF}}(k)$ converges; or $M^{\mathrm{IPF}}(k)$ oscillates between accumulation points [17]. IPF is more prone to non-convergence when the inputs are *sparser*, since it becomes harder to reconcile the marginals; this issue is particularly relevant given high levels of sparsity in real-world network data, which we demonstrate in Section 3. Prior works have extensively studied IPF and the associated matrix balancing problem, and derived several equivalent conditions that characterize exactly when IPF converges [14, 18–21]. A particularly useful perspective, which we focus on in this paper, comes from the principle of entropy minimization [22].

Connection to KL divergence. It is well-known that IPF iterations $M^{\mathrm{IPF}}(k)$ converge to $\hat{Y}$, the solution to the following minimization problem, as long as it is feasible and bounded [18, 23]:

$$\min_{\hat{Y}} D_{\mathrm{KL}}(\hat{Y} \| X), \tag{2}$$

subject to $\hat{Y}_{ij} \geq 0$, $\hat{Y}\mathbf{1}_n = p$, and $\hat{Y}^T \mathbf{1}_m = q$. Recall that for discrete distributions,

$$D_{\mathrm{KL}}(\hat{Y} \| X) = \sum_{ij} \hat{Y}_{ij} \log \frac{\hat{Y}_{ij}}{X_{ij}}, \tag{3}$$

so (2) is feasible and bounded if and only if there exists $\hat{Y}$ with the desired marginals p and q, and $\hat{Y}_{ij} = 0$ *whenever* $X_{ij} = 0$. In other words, $\hat{Y}$ must inherit all the zeros of X. If no such feasible solution exists for (2), IPF does not converge. Furthermore, the dual problem of KL minimization is

$$g(u,v) = \sum_{ij} X_{ij} e^{u_i - v_j} - \sum_i p_i u_i + \sum_j q_j v_j, \tag{4}$$

which is jointly convex in the dual variables $u \in \mathbb{R}^m, v \in \mathbb{R}^n$. IPF is a coordinate descent type algorithm for this problem [24], and (d^0, d^1) is a solution to the matrix balancing problem if and only if $u = \log d^0, v = -\log d^1$ is a minimizer of (4) [24]. For completeness, we provide details of the duality result in Appendix A.1. The main result of our work is showing that $-g(u,v)$ is the log-likelihood function of a Poisson generative network model.

Our network model. Several intuitions guide our construction of the network statistical model. There is a general correspondence between KL divergence minimization and maximum likelihood, although in (2) the ordering between empirical and model distributions has been reversed. Recently, Qu et al. [25] established connections between matrix balancing and choice modeling, and observed that (4) reduces to the maximum likelihood objective of a choice model. We therefore also construct a statistical framework for the network inference problem via the dual representation (4). Lastly, the choice of Poisson distribution is natural given its close connections with KL divergence, similar to the associations between ℓ_1 and Laplace distribution, as well as ℓ_2 and Gaussian distribution.

Based on these intuitions, we formulate the following generative network model:

$$Y_{ij} \sim \text{Poisson}(e^{u_i} X_{ij} e^{-v_j}) \text{ for } X_{ij} > 0, \tag{5}$$

$$p_i = \sum_{j, X_{ij}>0} Y_{ij}, \quad q_j = \sum_{i, X_{ij}>0} Y_{ij}. \tag{6}$$

Here $\{Y_{ij}\}_{i,j:X_{ij}>0}$ are a collection of independent Poisson random variables with parameters $\lambda_{ij} = e^{u_i} X_{ij} e^{-v_j}$. Now, we are ready to formally state the *network inference problem*:

> Given X, find parameters u and v that maximize the likelihood under this network model of Y, only the marginals p and q of which are observed.

Our main theorem connects IPF to this network inference problem. Its proof is in Appendix A.2. Notably, our theorem reveals that even though only X, p, and q are observable, maximum likelihood estimation of this model is still feasible, since the marginals of Y (p and q) are sufficient statistics.

Theorem 1. *Assume that the matrix balancing problem has a finite solution (D^0, D^1). Then d^0 and d^1 are limits of the IPF iterations if and only if $u = \log d^0$ and $v = -\log d^1$ are solutions to the network inference problem. Moreover, the network inference problem is equivalent to the maximum likelihood estimation of a Poisson regression model, and p, q are the sufficient statistics.*

Implications of our result. Connecting IPF to this network model allows us to interpret IPF through the lens of a dynamic network process. In the original matrix balancing problem, where IPF has typically been applied, the goal is to learn a scaling of X that satisfies marginals p and q, but there is no explicit network that we are trying to infer. Now, given our network model, we are explicitly trying to infer an up-to-date network Y, given its marginals p and q, and X, a historical or aggregate network. For example, in the mobility setting from Chang et al. [1], X serves as the time-aggregated mobility network and p and q are the observed hourly marginals for POIs and neighborhoods. Then, Y is the (unobserved) hourly network traffic, and we can view u and v as each POI's and neighborhood's *time-varying* mobility dynamics, e.g., if schools are visited more during the day or if younger populations go out more after work.

Moreover, since $d_i^0 X_{ij} d_j^1$ from IPF corresponds to $e^{u_i} X_{ij} e^{-v_j}$, the expectation of the Poisson variable Y_{ij}, we can interpret the matrix inferred by IPF as the *expected* values of the network-generating process. This also explains our choice of notation in (2), where we use $\hat{Y}$ to emphasize the fact that it is the predicted (expected) value of the Poisson network variables. Our model also clarifies previously implicit assumptions. For example, this model assumes that each Y_{ij} is sampled independently, and that the estimated network $\hat{Y}$ is an (entry-wise) rank-1 modification of the given aggregate network X. Making these assumptions explicit allows practitioners to evaluate how reasonable these assumptions are given their domain and data, when deciding whether to use IPF to infer dynamic networks. In the following section, we also show how the network model enables new empirical analyses of IPF, such as quantifying uncertainty, evaluating estimation error, and assessing the impact of sparsity.

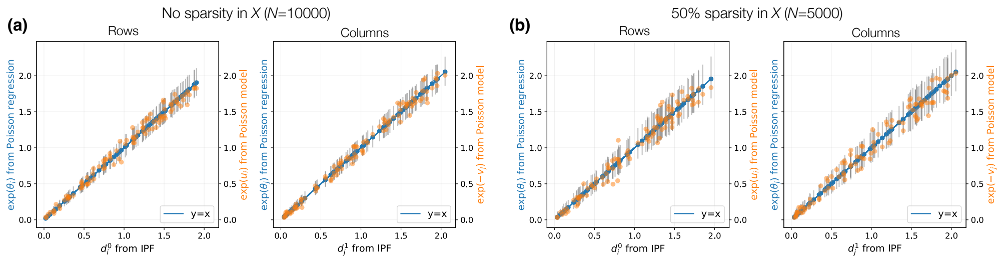


Figure 1: Comparing inferred parameters from IPF (x-axis) against inferred parameters from Poisson regression (left y-axis, blue) and true parameters from Poisson model (right y-axis, orange). Grey bars indicate 95% CIs from Poisson regression.

3 Experiments with real-world and synthetic data

Synthetic data and Poisson regression. In our first set of experiments, we use synthetic data generated from our network model to confirm the correspondence between IPF and Poisson regression (Appendix A.5). In Figure 1, we show that the parameters inferred by IPF and Poisson regression align perfectly, validating Theorem 1. Furthermore, using our network model, we can evaluate IPF's performance in terms of its estimation error on the network model's true parameters, which is distinct from its error on the marginals, which is how IPF is typically analyzed. We find that, despite matching the marginals in all cases, the ℓ_2 distance between IPF's inferred parameters and the true parameters increases quadratically as we increase the sparsity in X (Figure A.1). Second, since we mapped IPF to a maximum likelihood problem, we can now quantify uncertainty in its estimates. In Figure 1, we display 95% confidence intervals (CIs); while these CIs are only asymptotically valid, they provide measures of uncertainty where prior uses of IPF for this problem had none.

Mobility networks, sparsity, and convergence. We test our methods on real-world mobility networks, using aggregated location data from Chang et al. [1, 9]. We focus on the Richmond metropolitan statistical area in Virginia, which has 9917 POIs and 1098 neighborhoods. In this setting, $p(t)$ represents hourly visits to POIs, $q(t)$ represents hourly visits from neighborhoods, and X represents time-aggregated visits from neighborhoods to POIs, aggregated from January to October 2020 (Appendix A.6.1). We find that, despite aggregating over 10 months, X remains quite sparse: only 8% of its entries are non-zero. Furthermore, in nighttime hours, up to 90% of POIs have zero marginals. From running IPF on these two days, we find that IPF is decently robust to sparsity, converging for 45 out of the 48 hours. However, IPF gets stuck in oscillation for 3 hours during nighttime, when POI marginals are particularly sparse. We can evaluate convergence by running IPF or directly test for convergence from the input data X, p, q by running a max-flow-based algorithm that checks one of the IPF convergence conditions (Appendix A.4) [15].

We also plot the total ℓ_1 error between the marginals of M^{IPF} and the target marginals, which is known to decrease monotonically with each iteration [21]. When IPF converges, we find empirically that the error decreases exponentially (Figure A.3). This intriguing observation is worthy of further study given that IPF only converges exponentially in certain settings [17]. Finally, we compare IPF to Poisson regression on this mobility data. Since we are using real-world data, we do not have access to Y, as defined in our network model. Instead, we construct Y (using the same max-flow algorithm that we use to test convergence) such that Y inherits the zeros of X and satisfies the target marginals. We find that on this data as well, the parameters inferred by IPF and Poisson regression are perfectly matched (Figure A.4), providing further confirmation of Theorem 1.

4 Discussion

In this work, we have established an optimization principle of using IPF for network inference, by formulating a network model whose maximum likelihood objective corresponds to IPF. In Appendix A.3, we also discuss the necessary and sufficient conditions for IPF to recover the true network exactly (Theorem 2). Our empirics both confirm our theoretical results and demonstrate how this model enables new principled analyses of IPF. We have many future directions we hope to explore, including analyzing the robustness of IPF to noise and developing methods for IPF convergence.

References

[1] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589(7840):82–87, 2021. 1, 2, 3, 4, 9, 10

[2] J Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52:15–25, 1937. 1

[3] Malachy Carey, Chris Hendrickson, and Krishnaswami Siddharthan. A method for direct estimation of origin/destination trip matrices. *Transportation Science*, 15(1):32–49, 1981. 1

[4] David A Plane. An information theoretic approach to the estimation of migration flows. *Journal of Regional Science*, 22(4):441–456, 1982. 1

[5] Yaryna Serkez. The magic number for reducing infections and keeping businesses open. *The New York Times Opinion*, 2020. Available at `https://www.nytimes.com/interactive/2020/12/16/opinion/coronavirus-shutdown-strategies.html`. 1

[6] Kevin C. Ma and Marc Lipsitch. Big data and simple models used to track the spread of covid-19 in cities. *Nature News and Views*, 2020. Available at `https://www.nature.com/articles/d41586-020-02964-4`.

[7] US Joint Economic Committee. Further support for restaurants and restaurant workers is needed. 2021. Available at `https://www.jec.senate.gov/public/_cache/files/35bd8cec-1017-45e4-ba6f-31bb74660609/restaurant-report-1-12-2021-final.pdf`.

[8] Joe R. Whatley Jr., Edith M. Kallas, and Henry C. Quillen. Brief of the american medical association and the medical society of the state of new york as amici curiae in support of respondent. In *Agudath Israel of America et al. v. Andrew Cuomo*, 2020. Available at `https://www.supremecourt.gov/DocketPDF/20/20A90/161168/20201119102026299_20A90acTheAmericanMedicalAssociation.pdf`. 1

[9] Serina Chang, Mandy L. Wilson, Bryan Lewis, Zakaria Mehrab, Komal K. Dudakiya, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, Madhav Marathe, and Jure Leskovec. Supporting covid-19 policy response with large-scale mobility-based modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, 2021. 1, 2, 4, 9

[10] Swetaprovo Chaudhuri, Prasad Kasibhatla, Arnab Mukherjee, William Pan, Glenn Morrison, Sharmistha Mishra, and Vijaya Kumar Murty. Analysis of overdispersion in airborne transmission of covid-19. *Physics of Fluids*, 34(051914), 2022. 1

[11] Dongyue Li, Tina Eliassi-Rad, and Hongyang R. Zhang. Optimal intervention on weighted networks via edge centrality. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM '23')*, 2023. 1, 2, 9

[12] W. Edwards Deming and Frederick F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Ann. Math. Statist.*, 11(4): 427–444, 1940. 2

[13] Stephen E. Fienberg. An iterative procedure for estimation in contingency tables. *Ann. Math. Statist.*, 41(3):907–917, 1970. 2

[14] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. ii. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974. 2

[15] Martin Idel. A review of matrix scaling and sinkhorn's normal form for matrices and positive maps. *arXiv*, 2016. 2, 4, 7, 8

[16] Michael H Schneider and Stavros A Zenios. A comparative study of algorithms for matrix balancing. *Operations research*, 38(3):439–455, 1990. 2

[17] Friedrich Pukelsheim and Bruno Simeone. On the iterative proportional fitting procedure: Structure of accumulation points and l1-error analysis. *Preprint*, 2009. 2, 4

[18] Lev M Bregman. Proof of the convergence of sheleikhovskii's method for a problem with transportation constraints. *USSR Computational Mathematics and Mathematical Physics*, 7(1): 191–204, 1967. 2

[19] Michael Bacharach. Estimating nonnegative matrices from marginal data. *International Economic Review*, 6(3):294–310, 1965.

[20] Hamsa Balakrishnan, Inseok Hwang, and Claire J Tomlin. Polynomial approximation algorithms for belief matrix maintenance in identity management. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 5, pages 4874–4879. IEEE, 2004.

[21] Friedrich Pukelsheim. Biproportional scaling of matrices and the iterative proportional fitting procedure. *Ann. Oper. Res.*, 215:269–283, 2014. 2, 4, 7, 8, 12

[22] C Terrance Ireland and Solomon Kullback. Contingency tables with given marginals. *Biometrika*, 55(1):179–188, 1968. 2

[23] Flavien Léger. A gradient descent perspective on sinkhorn. *Applied Mathematics & Optimization*, 84(2):1843–1855, 2021. 2

[24] Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992. 3

[25] Zhaonan Qu, Alfred Galichon, and Johan Ugander. On sinkhorn's algorithm and choice modeling. *arXiv preprint arXiv:2310.00260*, 2023. 3

[26] Ravi Kumar, Andrew Tomkins, Sergei Vassilvitskii, and Erik Vee. Inverting a steady-state. In *WSDM*, 2015. 8

A Appendix

A.1 Duality result for KL divergence minimization

For completeness, we provide the details for the duality result. Let u and v be the multipliers of the constraints $\hat{Y}\mathbf{1}_n = p, \hat{Y}^T\mathbf{1}_m = q$, respectively. The problem is equivalent to

$$\min_{\hat{Y}} \max_{u,v} \sum_{ij} \hat{Y}_{ij} \log \frac{\hat{Y}_{ij}}{X_{ij}} - \sum_i u_i(\hat{Y}\mathbf{1}_n - p)_i + \sum_j v_j(\hat{Y}^T\mathbf{1}_m - q)_j =$$

$$\max_{u,v} \min_{\hat{Y}} \sum_{ij} \hat{Y}_{ij} \log \frac{\hat{Y}_{ij}}{X_{ij}} - \sum_i u_i(\hat{Y}\mathbf{1}_n - p)_i + \sum_j v_j(\hat{Y}^T\mathbf{1}_m - q)_j$$

where strong duality holds because both problems are feasible and bounded. Taking the first order condition with respect to $\hat{Y}_{ij}$, we obtain

$$\log \hat{Y}_{ij} = \log X_{ij} - 1 + u_i - v_j,$$

and substituting this back into the objective, we obtain

$$\max_{u,v} \sum_{ij} X_{ij} e^{-1+u_i-v_j}(-1 + u_i - v_j) - \sum_i u_i(\sum_j X_{ij} e^{-1+u_i-v_j} - p_i) + \sum_j v_j(\sum_i X_{ij} e^{-1+u_i-v_j} - q_j)$$

$$= \max_{u,v} - \sum_{ij} X_{ij} e^{-1+u_i-v_j} + \sum_i u_i p_i - \sum_j v_j q_j.$$

Finally, using the change of variable $u_i = u_i - \frac{1}{2}$ and $v_j = v_j + \frac{1}{2}$, we obtain

$$\max_{u,v} - \sum_{ij} X_{ij} e^{u_i - v_j} + \sum_i p_i u_i - \sum_j q_j v_j - \frac{\sum_i p_i + \sum_j q_j}{2} \Leftrightarrow$$

$$\min_{u,v} \sum_{ij} X_{ij} e^{u_i - v_j} - \sum_i p_i u_i + \sum_j q_j v_j,$$

which we recognize as $g(x, y)$.

A.2 Proof of Theorem 1

Proof. We start by showing that the log-likelihood of our Poisson network model is $-g(u, v)$ from (4). The Poisson network's likelihood is given by

$$\mathcal{L}(Y|X, u, v) = \Pi_{i,j;X_{ij}>0} \frac{(e^{u_i} X_{ij} e^{-v_j})^{Y_{ij}} \exp(-(e^{u_i} X_{ij} e^{-v_j}))}{Y_{ij}!}. \tag{7}$$

When maximizing the likelihood with respect to u and v, we can drop the denominator, which is constant in the parameters. Maximizing the log-likelihood yields the following problem:

$$\max_{u,v} \sum_{i,j;X_{ij}>0} Y_{ij} \cdot (u_i + \log X_{ij} - v_j) - e^{u_i} X_{ij} e^{-v_j}. \tag{8}$$

We can also drop $Y_{ij} \log(X_{ij})$ since it does not depend on u, v. The resulting problem is

$$\begin{aligned} &\max_{u,v} \sum_{i,j;X_{ij}>0} Y_{ij}u_i - Y_{ij}v_j - e^{u_i} X_{ij} e^{-v_j} && (9)\\ &= \max_{u,v} \sum_i u_i (\sum_{j;X_{ij}>0} Y_{ij}) - \sum_j v_j (\sum_{i;X_{ij}>0} Y_{ij}) - \sum_{i,j;X_{ij}>0} e^{u_i} X_{i,j} e^{-v_j} && (10)\\ &= \max_{u,v} \sum_i u_i p_i - \sum_j v_j q_j - \sum_{ij} e^{u_i} X_{ij} e^{-v_j}. && (11) \end{aligned}$$

Thus, our model's log-likelihood in (11) is indeed $-g(u, v)$ from (4). Furthermore, observe that (11) implies that the marginals of Y are sufficient statistics for our network model, conveniently aligning with the problem constraints from IPF.

Next, we show that the network inference problem is equivalent to a Poisson *regression* problem. Although our Poisson network model and Poisson regression are, not surprisingly, close relatives, it is instructive to precisely illustrate their connections and distinctions. Poisson regression is a generalized linear model (GLM) which defines the *logarithm* of a Poisson variable's expected value as a linear model of input features. Generically, let $\theta \in \mathbb{R}^d$ represent the parameters of a Poisson regression model, $\mathbf{x} \in \mathbb{R}^d$ represent input features, and y represent the observed non-negative count data. Ignoring constants, the log-likelihood of observing y under the Poisson regression model is

$$y \cdot \theta^T \mathbf{x} - e^{\theta^T \mathbf{x}}. \tag{12}$$

To match this to the log-likelihood of our network model in (8), for each sample indexed by i, j with $X_{ij} > 0$, we set the input features $\mathbf{x}$ to be $[\mathbf{e}_i, \mathbf{e}_j, \log X_{ij}]$, where $\mathbf{e}_i \in \{0, 1\}^m$ is a vector of all zeros aside from a 1 in the i-th position and $\mathbf{e}_j \in \{0, 1\}^n$ is a vector of all zeros aside from a 1 in the j-th position. Observe again that since this construction relies on $\log X_{ij}$, the Poisson regression model only applies to Y_{ij} where $X_{ij} > 0$. We then set the dependent variable to be $y = Y_{ij}$. Lastly, it is obvious that we should set the parameter $\theta \in \mathbb{R}^d$ with $d = m + n + 1$ as $\theta = [u, -v, 1]$. We can verify that the log-likelihood of this Poisson regression model is equal to the objective in (8), and that p, q are the sufficient statistics of the model. To perform Poisson regression, we may need a set of values Y_{ij} for $X_{ij} > 0$ that are consistent with the marginals. This can be achieved exactly by running the max flow algorithm on the bipartite graph defined by X [15]. For details on the maximum flow algorithm, see Section A.4. □

A.3 When IPF recovers the true network exactly

If we swap out the Poisson distribution for an identity function in (5), we find that this condition is necessary and sufficient for IPF to exactly recover the true network, Y, given X and Y's marginals p and q.

Theorem 2. *IPF returns the true network Y if and only if $Y = AXB$, for some positive diagonal matrices A and B.*

Proof. The first statement to prove is, if IPF returns Y, then $Y = AXB$. All IPF solutions take the form $D^0 A D^1$, where D^0 and D^1 are positive diagonal matrices. If IPF returns Y, then Y can be written as AXB, with $A = D^0$ and $B = D^1$.

The second statement to prove is, if $Y = AXB$, then IPF will return Y. To prove this, we can use *direct biproportional scalings* from Pukelsheim [21]. Matrix M^1 is a direct biproportional scaling of matrix M^0 if, for all i, j, $M^1_{ij} = M^0_{ij}/(\mu_i \nu_j)$, for positive divisors μ_i and ν_j. If Y is AXB, then Y is a direct biproportional scaling of X with the diagonal of A set to $1/\mu$ and the diagonal of B set to $1/\nu$. Second, biproportional scalings are unique with respect to marginals, meaning if two biproportional scalings M^1 and M^2 of M^0 have the same marginals, then $M^1 = M^2$ [21]. This means that if IPF can perfectly match the marginals p and q of Y by scaling X, then it must have found Y (since Y is a biproportional scaling of X). Finally, we know that if it is possible to perfectly match the marginals by scaling X, then IPF will return a solution that matches the marginals [21]. Thus, IPF will return Y, since IPF will find a biproportional scaling of X that matches marginals p and q, and Y is the unique scaling of X that achieves that.

While this statement about Y ($Y = AXB$) is rarely true in practice, this result allows us to pin down when IPF works "perfectly" for network inference. Notably, this result implies that the scaling matrix from X to Y, i.e., Y/X, must be a rank-1 matrix with entries $a_i b_j$ (the diagonals of A and B). If we interpret Y as a dynamic network and X as its time-aggregated form, then we are essentially constraining the complexity of the network's temporal variation. □

A.4 Algorithm to test for convergence

For completeness, we describe the algorithm from Idel [15] for testing whether IPF will converge based on input data X, p, and q. This algorithm is also closely related to the max-flow-based algorithm from Kumar et al. [26] to test their concept of graph consistency. First, one of the conditions that exactly characterizes when IPF will converge is whether there exists a weight matrix $W \in \mathbb{R}^{m \times n}$ that inherits the zeros of X and matches the marginals p and q. Note that this matrix is *more general* than the set of possible solutions to the matrix balancing problem, since W does not have to be a biproportional scaling of X. Now, the following algorithm will check for the existence of W.

Create a new directed graph G that has a source node s connected to one node n_i for each row and set the capacity of the edge $s \to n_i$ to p_i. Create a sink node t connected to one node n_j for each column and set the capacity of the edge $n_j \to t$ to q_j. Finally, include an edge $n_i \to n_j$, with capacity ∞, wherever $X_{ij} > 0$. Compute the maximum flow f_G on the resulting graph. If the maximum flow f_G is equal to $\sum_i p_i = \sum_j q_j$, then the weight matrix W exists, meaning IPF converges for X, p, and q; otherwise, it does not converge.

A.5 Experiments with synthetic data

Generating synthetic data. We generate synthetic data based on our generative network model in (5)-(6). In these experiments, we set $m = n = 100$, and generate data in the following order:

1. We sample the row scaling factors $e^u \in \mathbb{R}^m$ and column scaling factors $e^{-v} \in \mathbb{R}^n$ from Uniform(0, 4).
2. We sample $X \in \mathbb{R}^{m \times n}$ from Uniform(0, 1).
3. For a given sparsity level $r \in [0, 1)$, we randomly select $r \cdot mn$ entries from X (without replacement) and set them to 0.
4. We sample each Y_{ij} from Poisson($e^{u_i} X_{ij} e^{-v_j}$).
5. We set $p = Y\mathbf{1}_m$ and $q = Y^T\mathbf{1}_n$.

Comparing IPF and Poisson regression. In our IPF experiments, we run IPF on X, p and q, producing parameters $d^0 \in \mathbb{R}^m$ and $d^1 \in \mathbb{R}^n$. In our Poisson regression experiments, we fit a Poisson regression model on all observations where $X_{ij} > 0$, following the construction in Section A.2. Poisson regression returns parameters $\{\theta_i\}_{i=1}^m$ corresponding to rows and parameters $\{\theta_j\}_{j=1}^n$ corresponding to columns. Based on Theorem 1, we expect $d_i^0 = \exp(\theta_i)$, for all $i \in [m]$, and $d_j^1 = \exp(\theta_j)$, for all $j \in [n]$, subject to arbitrary scaling between rows and columns (i.e., scaling row factors by k and scaling column factors by $1/k$). To control for such scaling, we normalize both sets of parameters by dividing by their means. In Figures 1 and A.4, we plot the normalized IPF parameters versus the Poisson regression parameters. We find that they lie perfectly on the $y = x$ line, validating Theorem 1. We also plot the 95% confidence intervals, as provided by Python's statsmodels package for fitting generalized linear models.[1] Finally, we also plot the true parameter values e^u and e^{-v} in Figure 1, which we can only include in our synthetic setting since we know the true network model's parameters.

[1] https://www.statsmodels.org/stable/glm.html

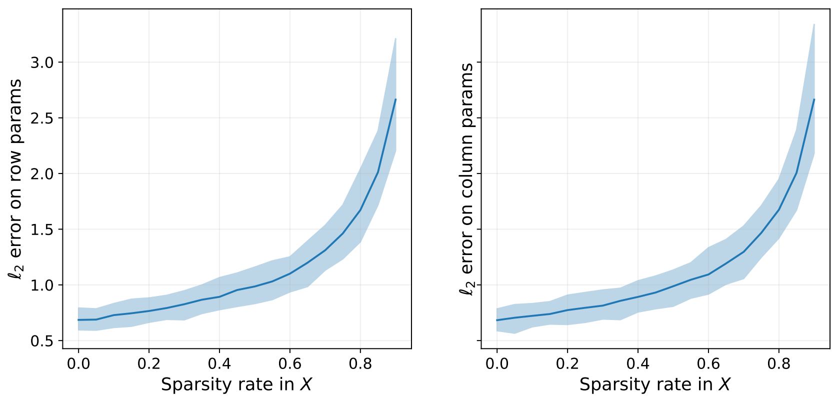


Figure A.1: Comparing sparsity rate in X to IPF's ℓ_2 error, i.e., the ℓ_2 distance between IPF's inferred parameters and the true model parameters.

ℓ_2 estimation error vs. sparsity in X. As an additional experiment, we evaluate IPF's ℓ_2 error on the true parameters:

$$\ell_2^0 = ||d^0 - e^u||_2 = \sqrt{\sum_i (d_i^0 - e^{u_i})^2} \tag{13}$$

$$\ell_2^1 = ||d^1 - e^{-v}||_2 = \sqrt{\sum_j (d_j^1 - e^{-v_j})^2}, \tag{14}$$

where we used the mean-normalized version of all parameters. For each sparsity rate in $r \in \{0, 0.05, \cdots, 0.9\}$, we run 100 random trials, where in each trial, we repeat steps 2-5 of our generative process with that sparsity rate in X (i.e., we fix e^u and e^{-v} but resample all other variables), run IPF on the newly generated X, p, and q, and compute its ℓ_2 error. In Figure A.1, we visualize our results, showing the ℓ_2 mean and 95% CIs (from 2.5th to 97.5th percentiles over random trials) over sparsity rates. We find that ℓ_2 error increases quadratically with greater sparsity in X.

A.6 Experiments with real-world mobility data

We conduct experiments with real-world mobility data, motivated by past applications that used IPF to infer mobility networks from aggregated cell phone location data, then integrated those networks into epidemiological models to simulate the spread of infectious diseases, like SARS-CoV-2 [1, 9–11].

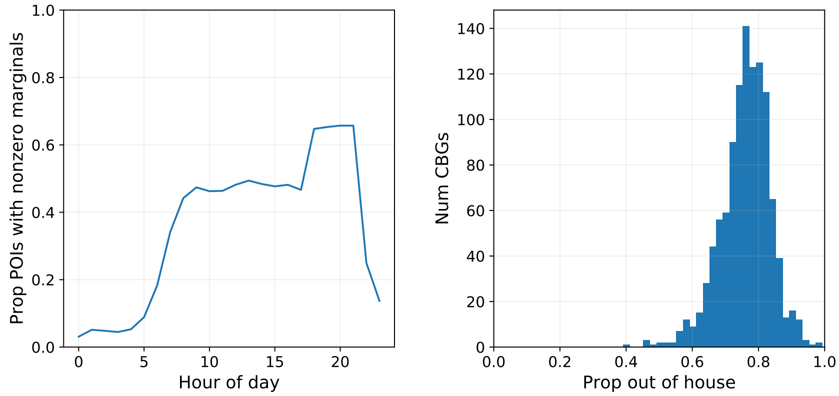
A.6.1 Constructing marginals from mobility data

In the mobility network inference setting, our goal is to infer the hourly network at hour t from n census block groups (CBGs), which are neighborhoods, to m points-of-interest (POIs), which are individual locations such as restaurants, grocery stores, or gas stations. To do this, we construct:

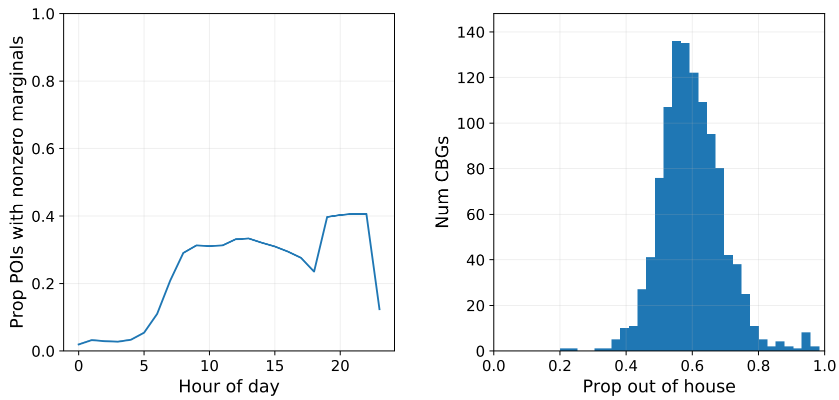
- $X \in \mathbb{R}_+^{m \times n}$, a time-aggregated visit matrix,
- $p(t) \in \mathbb{R}_+^m$, the hourly number of visitors to each POI,
- $q(t) \in \mathbb{R}_+^n$, the hourly number of visitors from each CBG.

We construct these quantities from SafeGraph data in the same way as the authors did in Chang et al. [1, 9]. We summarize this procedure below, highlighting a few important facts, and point to the original text for details.

Constructing matrix X. SafeGraph provides summaries of the home CBGs of each POI's visitors, per month (before March 2020) or week (after March 2020). To account for non-uniform sampling from different CBGs, we weight the number of SafeGraph visitors from each CBG by the ratio of the CBG population (from US Census) and the number of SafeGraph devices with homes in that CBG. Following the original text, let $\hat{W}(r)$ represent the reweighted matrix for period r (we use r instead



(a) Marginals on Monday, March 2, 2020.



(b) Marginals on Monday, April 6, 2020.

Figure A.2: POI and CBG marginals from mobility data for Richmond, Virginia MSA.

of t to denote time periods longer than an hour). Since these visit matrices are sparse, we aggregate over R time periods:

$$\bar{W} = \frac{1}{R}\sum_{r=1}^{R} \hat{W}(r) \tag{15}$$

$$X_{ij} = \frac{\bar{W}_{ij}}{\sum_k W_{kj}}. \tag{16}$$

So, X_{ij} represents the time-aggregated *proportion* of visits to POI i that come from CBG j. Note that SafeGraph's visit matrices include all possible home CBGs, but when we construct X, we only include the n CBGs for the metropolitan statistical area. So, the rows of X typically do not sum to 1 and are usually around 0.9-0.97.

Constructing visitors to POIs, $p(t)$. SafeGraph provides the hourly number of *visits*, not visitors, so first we apply corrections to the SafeGraph counts based on the POI's median dwell time to estimate the hourly number of visitors (see Supplementary Information from Chang et al. [1]). To account for SafeGraph undersampling, we also multiply each POI's visit count by a uniform correction factor which is the ratio of the US population to the total number of SafeGraph devices; this factor is around 7. Finally, since not all of the POI's visits are captured by the n CBGs in X, we multiply the POI's visits by its row sum in X, i.e., its total proportion of visits kept.

In Figure A.2, we visualize the proportion of POI marginals that are nonzero, over 24 hours in the day on March 2, 2020 and April 6, 2020. We see that only a small proportion of POIs have nonzero marginals at nighttime, e.g., less than 10% from 12-5am. For both days, the proportion peaks from around 6-10pm, likely when people are visiting POIs after work. We also see considerably more sparsity in POI marginals on April 6, compared to March 2, which reflects the onset of the COVID-19 pandemic in the US. In Section A.6.2, we discuss how to run and interpret IPF with row or column marginals that contain zeros, since IPF is typically defined with strictly positive marginals.

Constructing visitors from CBGs, $q(t)$. Using SafeGraph data, we can estimate $\hat{h}_j(t)$, the fraction of each CBG that has left their home. Then, we estimate the number of people who left their home by multiplying these fractions by the CBG population N_j (from US Census). Finally, we scale these estimates so that $q(t)$ and $p(t)$ sum to the same totals. We do this for two reasons: first, we want to ensure that the sum of the POI and CBG marginals match; second, since the number of people who are not at home may not exactly match the number of people who are visiting POIs. So, we have

$$q_j(t) = \hat{h}_j(t) N_j \cdot \frac{\sum_i p_i(t)}{\sum_k \hat{h}_k(t) N_k}. \tag{17}$$

In Figure A.2, we also visualize the proportions out of the house per CBG. We only have these quantities at a daily granularity from SafeGraph, so we plot a histogram over CBGs instead of an hourly measure. We can see that, in this setting, CBG marginals are always positive, even after the pandemic onset.

A.6.2 IPF experiments on mobility data

IPF with zeros in marginals. IPF returns a matrix of the form $D^0 X D^1$, where D^0 and D^1 are positive diagonal matrices. So, unless the entire row or column of X is 0, IPF solutions cannot naturally match zeros in the target row or column marginals. However, we can modify IPF slightly to allow for non-negative, instead of strictly positive, marginals by setting $d^0_i = 0$, for all $p_i = 0$, and setting $d^1_j = 0$, for all $q_j = 0$, then updating all other entries in d^0 and d^1 as usual, as described in (1). We will show that this is still a valid IPF procedure and all guarantees of IPF hold, because this procedure is *equivalent* to running the original IPF procedure on $\tilde{X}$, $\tilde{p}$, and $\tilde{q}$, where $\tilde{X}$ is a submatrix of X that leaves out the rows and columns with zero marginals, $\tilde{p}$ contains the nonzero entries in p, and $\tilde{q}$ contains the nonzero entries in q.

For some row i where $p_i > 0$, let d^0_i represent IPF's inferred parameter under the modified IPF procedure on X, p, q, and let $\tilde{d}^0_i$ represent IPF's inferred parameter under the original IPF procedure on $\tilde{X}$, $\tilde{p}$, and $\tilde{q}$. Let d^1_j and $\tilde{d}^1_j$ be defined analogously. We will prove by induction that, for all iterations k, $d^0_i(k) = \tilde{d}^0_i(k)$, $\forall i$ s.t. $p_i > 0$, and $d^1_j(k) = \tilde{d}^1_j(k)$, $\forall j$ s.t. $q_j > 0$. First, in the base case, $d^0_i(0)$, $\tilde{d}^0_i(0)$, $d^1_j(0)$, and $\tilde{d}^1_j(0)$ are all initialized to 1. Now, assuming the statement holds up to iteration k, the next IPF update is

$$\begin{aligned} d^0_i(k+1) &= \frac{p_i}{\sum_j X_{ij} d^1_j(k)} && (18)\\ &= \frac{p_i}{\sum_{j;q_j>0} X_{ij} \tilde{d}^1_j(k)} && (19)\\ &= \tilde{d}^0_i(k+1). && (20) \end{aligned}$$

In (19), we can drop all the terms where $q_j = 0$, since in our modified algorithm, we set $d^1_j = 0$ if $q_j = 0$. Furthermore, since we are only considering j where $q_j > 0$, then we can replace $d^1_j(k)$ with $\tilde{d}^1_j(k)$, based on the inductive hypothesis. A similar proof follows to show the inductive step for d^1_j and $\tilde{d}^1_j$. Recall that in the connection between IPF and our Poisson network model, the number of nonzero entries in X is our number of Poisson observations. One implication of this equivalence between modified IPF for non-negative marginals and original IPF on the submatrix is that zero marginals can substantially reduce our number of Poisson observations, since the submatrix drops entire rows and columns. In other words, even if $X_{ij} > 0$, we will lose it in our observations if $p_i = 0$ or $q_j = 0$.

IPF convergence on mobility data. We run (modified) IPF for all hours on March 2 and April 6, 2020, and we find that IPF converges for 45 out of the 48 hours. However, it gets stuck in oscillation for 3 hours during nighttime, when POI marginals are particularly sparse (2am on March 2 and 12am

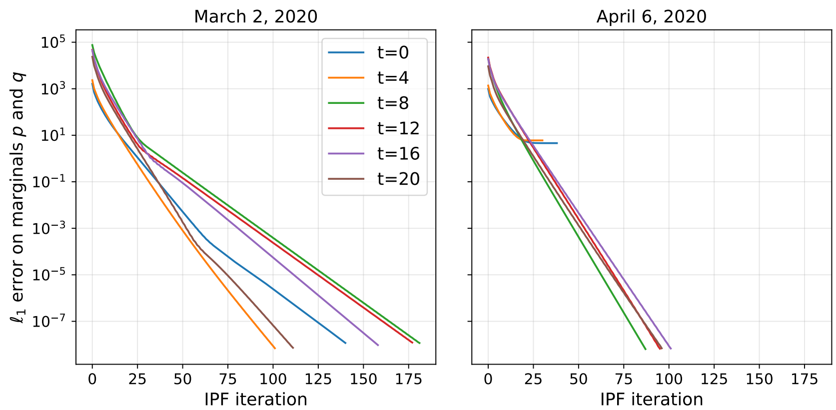


Figure A.3: ℓ_1 error on marginals p and q over IPF iterations on mobility data. We show convergence results from hours $t \in \{0, 4, 8, 12, 16, 20\}$ on March 2, 2020 (left) and April 6, 2020 (right).

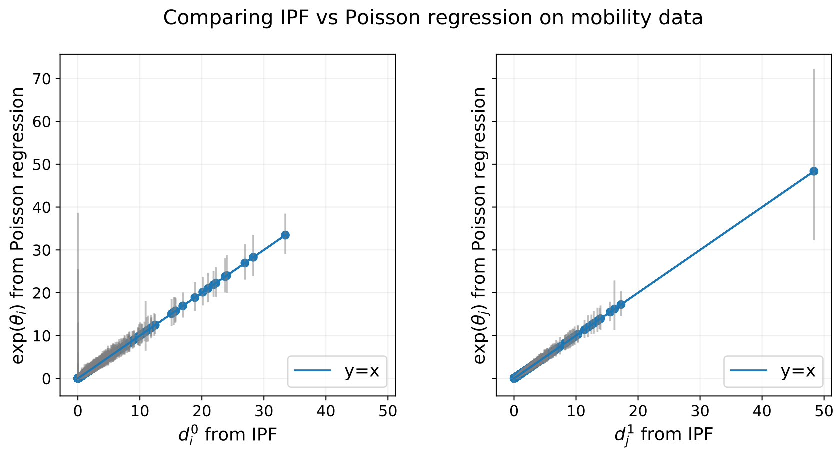


Figure A.4: Comparing inferred parameters from IPF (x-axis) vs. Poisson regression (y-axis) on mobility data.

and 4am on April 6). We evaluate convergence in two ways: first, we check for it in the IPF iterations, where we assume IPF converges if the difference between $M^{\text{IPF}}(k)$ and $M^{\text{IPF}}(k+1)$ is smaller than $\epsilon = 10^{-8}$. We also directly test for convergence from the input data, i.e., X, p, and q, by running a max-flow-based algorithm that checks one of the IPF convergence conditions (Section A.4).

In Figure A.3, we plot the ℓ_1 error between the marginals of M^{IPF} and the target marginals, which is known to decrease monotonically with each iteration [21]. When IPF converges, we find that the error decreases exponentially, although the convergence sometimes demonstrates a one-time "bend" where the exponential rate changes (e.g., for $t = 8$ on March 2). When IPF does not converge, as in the case of $t = 0$ and $t = 4$ on April 6, its ℓ_1 error gets stuck at a fixed value, since that fixed value gets passed back and forth from error on the row marginals to error on the column marginals.

Comparing IPF and Poisson regression on mobility data. In our final experiment, we compare the inferred parameters from IPF to those inferred by Poisson regression, this time on the mobility data instead of synthetic data (Section A.5). Unlike in the case of synthetic data, we do not have access to individual Y_{ij} terms when we are using mobility data; we only have Y's marginals p and q. However, we only need a matrix Y that inherits the zeros of X and matches the target marginals, which we can construct using the max-flow-based convergence algorithm (Section A.4). In Figure A.4, we show that the IPF-inferred parameters and Poisson regression parameters are perfectly aligned (after dividing by their respective means). We also plot the 95% confidence intervals from the Poisson regression estimates. The CIs are mostly small, aside from a few outliers.