Inferring Dynamic Networks from Marginals with Iterative Proportional Fitting

Serina Chang [*1] **Frederic Koehler** [*2] **Zhaonan Qu** [*3 4] **Jure Leskovec** [1] **Johan Ugander** [4]

Abstract

A common network inference problem, arising from real-world data constraints, is how to infer a dynamic network from its time-aggregated adjacency matrix and time-varying marginals (i.e., row and column sums). Prior approaches to this problem have repurposed the classic iterative proportional fitting (IPF) procedure, also known as Sinkhorn's algorithm, with promising empirical results. However, the statistical foundation for using IPF has not been well understood: under what settings does IPF provide principled estimation of a dynamic network from its marginals, and how well does it estimate the network? In this work, we establish such a setting, by identifying a generative network model whose maximum likelihood estimates are recovered by IPF. Our model both reveals implicit assumptions on the use of IPF in such settings and enables new analyses, such as structure-dependent error bounds on IPF's parameter estimates. When IPF fails to converge on sparse network data, we introduce a principled algorithm that guarantees IPF converges under minimal changes to the network structure. Finally, we conduct experiments with synthetic and real-world data, which demonstrate the practical value of our theoretical and algorithmic contributions.

1. Introduction

Dynamic networks of human movements are integral to important societal problems, such as epidemic response and transportation planning, but they are rarely fully observed. Instead, it is often easier to observe the time-varying *marginals* of the network, i.e., the row sums and column sums of its adjacency matrix. In transportation networks, it is easier to observe passengers embarking or disembarking, e.g., at bus stops (Navick & Furth, 1994), but harder to uncover routes between stops. In migration networks, it is easier to observe the number of individuals leaving from and arriving at a region but harder to estimate the network of their movements (Plane, 1982; Pham & Komiyama, 2022). In these settings, we observe the time-varying marginals regularly, but only have occasional access to a time-aggregated network (e.g., from surveys). Thus, a natural *dynamic network inference problem* arises from such data: given a network with adjacency matrices $X^{(t)}$ per time step t, can we reasonably infer $X^{(t)}$ from its time-varying marginals and a time-aggregated version, $\sum_t X^{(t)}$, of the network?

One approach to solving this problem is to repurpose the classic iterative proportional fitting (IPF) procedure (Deming & Stephan, 1940), also widely known as Sinkhorn's algorithm (Sinkhorn, 1974). Given a matrix X, target row marginals p, and target column marginals q, IPF tries to find a *biproportional* scaling of X to match the target marginals. IPF poses an attractive solution for the network inference problem: it matches the marginal constraints of the problem, is computationally lightweight and space efficient, and is supported by decades of literature devoted to its analysis (Bregman, 1967; Fienberg, 1970; Csiszár, 1975; Pukelsheim & Simeone, 2009; Marino & Gerolin, 2020; Léger, 2021; Carlier, 2022; Qu et al., 2023). Furthermore, prior works have shown empirical success with using IPF to infer networks (Liang et al., 2006; McCord et al., 2010; Chang et al., 2021a), enabling important applications such as modeling epidemic spread on mobility networks (Li et al., 2023; Chaudhuri et al., 2022; Alimohammadi et al., 2023) and supporting policymakers (Chang et al., 2021b).

However, despite the appeal and empirical success of using IPF for network inference, what is missing is a firm statistical grounding for *when* and *why* IPF is justified in this setting. While it is well-known that IPF solves a Kullback-Leibler (KL) divergence minimization problem (Ireland & Kullback, 1968), a formal connection to statistical theory is limited, especially in network settings. Moreover, IPF occasionally fails to converge on sparse network data. While tests exist for whether IPF will converge, there is no clear

[*]Equal contribution [1]Department of Computer Science, Stanford University [2]Department of Statistics and Data Science Institute, University of Chicago [3]Department of Economics, Stanford University [4]Department of Management Science & Engineering, Stanford University. Correspondence to: Serina Chang <serinac@stanford.edu>, Frederic Koehler <fkoehler@uchicago.edu>, Zhaonan Qu <zhaonanq@stanford.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

answer to how to interpret failed convergence, or how to repair it. Thus, to rigorously employ IPF to infer networks, we ask: under what model is IPF a principled estimator of a dynamic network from its marginals? Under this model, how well does IPF estimate these networks? What are principled strategies to ensure that IPF converges?

In this work, we first identify a generative network model, which we term the *biproportional Poisson model*, whose maximum likelihood estimates (MLEs) are recovered by IPF, thus establishing a statistical framework under which IPF is a principled estimation procedure of a dynamic network from its marginals (Theorem 3.1). Our model clarifies implicit network assumptions when using IPF and enables the analysis of IPF estimates using statistical theory of maximum likelihood estimation. Next, we provide expectation and tail bounds on estimation errors of the MLEs, and show that finite MLEs exist with high probability under the biproportional Poison model (Theorems 4.1-4.2). To address the issue of IPF non-convergence, we introduce a principled and polynomial time algorithm, `ConvIPF`, that guarantees IPF convergence while making minimal changes to the network structure (Section 5). Finally, we conduct extensive experiments with synthetic data and two real-world datasets: mobility data from SafeGraph and bikeshare data from New York City's CitiBike (Section 6).[1] Our experiments demonstrate IPF's ability to infer ground-truth hourly networks, outperforming several baselines, and tie our theoretical and algorithmic contributions to practical insights.

Our results provide much-needed theoretical foundation to justify recent high-impact applications of IPF to infer dynamic networks and to rigorously motivate future uses of IPF for this and related problems. Given the vast literature on IPF (Sinkhorn's algorithm) and matrix balancing, connecting this network inference problem to IPF also opens up future avenues of research, creating a bridge from modern data-driven problems to decades of statistical theory.

2. Related Work

Iterative proportional fitting. IPF (Deming & Stephan, 1940), also known as Sinkhorn's algorithm, biproportional fitting, raking, and the RAS algorithm, has a long history across disciplines. IPF offers many empirical advantages: it is straightforward to implement (see Algorithm 1 in Appendix A), transparent, reproducible, space efficient, and computationally lightweight (Liang et al., 2006; Lomax & Norman, 2015; Lovelace et al., 2015). The algorithmic properties of IPF have also been extensively studied (Sinkhorn, 1964; Fienberg, 1970; Franklin & Lorenz, 1989; Pukelsheim & Simeone, 2009; Qu et al., 2023). Notably, IPF solves a KL divergence minimization problem (Bregman, 1967; Ireland & Kullback, 1968; Csiszár, 1975; Ruschendorf, 1995), and is a coordinate descent type algorithm for its *dual* problem (Luo & Tseng, 1992). Recently, Qu et al. (2023) further established connections between IPF and ML estimation of choice models. We leverage these observations to identify a Poisson network model whose MLEs are recovered by IPF (Section 3). In Appendix B.3, we discuss other settings where IPF is known to recover MLEs of distinct models, such as contingency tables (Bishop & Fienberg, 1969; Bishop et al., 1974; Little & Wu, 1991; Little, 1993). The connections of IPF to Poisson-type models also have precedence in economics, such as trade (Silva & Tenreyro, 2006) and matching (Galichon & Salanié, 2022a;b). IPF (or Sinkhorn's algorithm) is also closely connected to (discrete) entropy regularized optimal transport and Schrödinger bridges, which can be reformulated as matrix balancing problems with a kernel/reference matrix and fixed marginals, and solved using the IPF procedure (Cuturi, 2013; Marino & Gerolin, 2020; De Bortoli et al., 2021).

Despite extensive literature on IPF's convergence behavior, few works have discussed principled solutions when IPF does *not* converge, which may occur when there are many zeros in the initial matrix or marginals (Bishop et al., 1974; Wong, 1992). The typical solution is to replace *all* zeros with small positive values (Lomax & Norman, 2015; Lovelace et al., 2015). However, this approach may result in unrealistic positive entries or, in our setting, vastly alter the structure of the network. To address this gap in literature, we propose a principled algorithm that guarantees IPF convergence while *minimally* changing the network (Section 5).

Network inference. The problem of inferring dynamic networks from marginals has been central to the impactful guidance of COVID-19 policy from mobility data (Chang et al., 2021a;b; Chaudhuri et al., 2022; Li et al., 2023; Alimohammadi et al., 2023), and also appears across many domains, including transportation (Carey et al., 1981), communication (Kruithof, 1937), and migration (Plane, 1982; Pham & Komiyama, 2022). Prior works have also explored related but distinct network inference problems. When node-level signals are observed but the network is not known at all, temporal or spatial relationships are often used to infer which nodes are likely to be connected (Gomez-Rodriguez et al., 2012; Hallac et al., 2017; Akagi et al., 2018; Chen et al., 2021; Rossi et al., 2022). A closer setting to ours is where the structure of the graph and the node-level marginals are known (Kumar et al., 2015; Maystre & Grossglauser, 2017). Our setting is distinct since we study dynamic networks and have access to the time-aggregated network, not just the binary structure. Nevertheless, we note similarities, as both settings solve matrix balancing problems, with the ChoiceRank algorithm of Maystre & Grossglauser (2017) closely

[1] Our code is available at https://github.com/snap-stanford/ipf-network-inference. Citibike data (CitiBike, 2023) and SafeGraph mobility data (Dewey, 2023) are available online.

connected to IPF as well (Qu et al., 2023).

Our network inference problem is also related to collective graphical models (CGMs), which aim to fit a model of individual behavior from aggregate data (Sheldon & Dietterich, 2011). Like CGMs, we also seek to estimate finer-grained information based on coarser data, but our setting differs in a number of ways: CGMs are typically applied in settings where only the time-varying marginals are known, such as to estimate population flows from counts per region over time (Iwata et al., 2017; Akagi et al., 2018; Iwata & Shimizu, 2019); CGMs try to learn a model of *individual* behavior, which is not our goal; and their mathematical formulations and estimation procedure are distinct from IPF. We compare our work to CGMs in greater detail in Appendix B.3.

3. Biproportional Poisson Model: A Statistical Framework of IPF for Inferring Networks

In this section, we provide background on IPF and define our dynamic network inference problem, then propose a statistical framework—via our generative network model—that establishes IPF as a principled solution to this problem.

Matrix balancing and IPF. IPF is an iterative algorithm that seeks to solve the following *matrix balancing problem*:

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

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:

$$d^0_i(k+1) = \frac{p_i}{\sum_j X_{ij} d^1_j(k)}, \quad d^1_j(k+1) = d^1_j(k), \tag{1}$$
$$d^1_j(k+2) = \frac{q_j}{\sum_i X_{ij} d^0_i(k+1)}, \quad d^0_i(k+2) = d^0_i(k+1).$$

We denote by $M^{\text{IPF}}(k)$ the scaled matrix after the k-th iteration: $M^{\text{IPF}}(k) := D^0(k) X D^1(k)$. The convergence behavior depends on the problem structure: if the matrix balancing problem has a finite solution, $(D^0(k), D^1(k))$ will converge to it; $(D^0(k), D^1(k))$ can diverge but $M^{\text{IPF}}(k)$ converges; or $M^{\text{IPF}}(k)$ does not converge and instead oscillates between accumulation points (Pukelsheim & Simeone, 2009). It is well-known that IPF solves the following KL divergence minimization problem, and IPF converges as long as the KL problem is feasible and bounded (Bregman, 1967; Ireland & Kullback, 1968; Léger, 2021):

$$\min_{\hat{Y}} D_{\text{KL}}(\hat{Y} \| X) \Leftrightarrow \min_{\hat{Y}} \sum_{ij} \hat{Y}_{ij} \log \frac{\hat{Y}_{ij}}{X_{ij}}, \tag{2}$$

subject to $\hat{Y}_{ij} \geq 0$, $\hat{Y}\mathbf{1}_n = p$, and $\hat{Y}^T \mathbf{1}_m = q$.[2] (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 $X_{ij} = 0$. Furthermore, the dual problem of (2) minimizes the potential function

$$g(u, v) := \sum_{ij} X_{ij} e^{u_i - v_j} - \sum_i p_i u_i + \sum_j q_j v_j, \tag{3}$$

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 (3) 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 (3) (Luo & Tseng, 1992). For completeness, we provide details of the duality result in Appendix B.1. The KL minimization problem (2) is also closely related to entropy regularized optimal transport, which we discuss in more detail in Appendix B.2.

Dynamic network inference problem. In our setting, we have a dynamic network with discrete time steps, where $X^{(t)} \in \mathbb{R}^{m\times n}_+$ represents the weighted adjacency matrix at time t. We do not have access to $X^{(t)}$ due to privacy or sampling constraints, but we observe its row sums $p^{(t)} := X^{(t)}\mathbf{1}_n$ and column sums $q^{(t)} := (X^{(t)})^T \mathbf{1}_m$, as well as a time-aggregated network, $\bar{X} := \sum_{t=1}^T X^{(t)}$, for some large T. The goal of the dynamic network inference problem is to provide a reasonable estimate of $X^{(t)}$, given $p^{(t)}$, $q^{(t)}$, and $\bar{X}$. The correspondence between this problem and the matrix balancing problem is natural: we can treat $\bar{X}$ as the initial matrix, $p^{(t)}$ as the target row marginals, and $q^{(t)}$ as the target column marginals. IPF's solution to this matrix balancing problem then serves as an estimate of the hourly network $X^{(t)}$. However, how should we interpret this estimate and why is IPF a justified approach here?

Our proposed network model. We provide a statistical justification of IPF as a solution to the network inference problem by identifying a generative network model under which IPF in fact recovers the MLEs of the network parameters. Our model, which we term the *biproportional Poisson model*, is defined as follows with parameters u, v:

$$X^{(t)}_{ij} \sim \begin{cases} \text{Poisson}(e^{u_i} \bar{X}_{ij} e^{-v_j}), \text{ if } \bar{X}_{ij} > 0, \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$
$$p^{(t)}_i = \sum_j X^{(t)}_{ij}, \quad q^{(t)}_j = \sum_i X^{(t)}_{ij}.$$

Model (4) posits *independent Poisson* samples $X^{(t)}_{ij}$ with expected value $\lambda_{ij} = e^{u_i} \bar{X}_{ij} e^{-v_j}$ wherever $\bar{X}_{ij} > 0$ (we

[2] Technically, the KL divergence $D_{\text{KL}}(P\|Q)$ is well-defined only on probability distributions P, Q. Without loss of generality, we may rescale X so that $\sum_{ij} X_{ij} = \sum_i p_i = \sum_j q_j = c$. Then normalizing $\hat{Y}, X$ by the common constant c yields valid discrete probability distributions. The resulting KL minimization problem is equivalent to (2).

suppress time-indexing on u and v to simplify notation). Note that the model is unique up to normalization, e.g., $(u, v)^T \mathbf{1}_{m+n} = 0$, since adding a constant c to (u, v) yields the *same* model. Since $X^{(t)}$ is not observed, it is *a priori* not obvious if the model parameters can be recovered by maximum likelihood estimation. Our first result reveals that although only $\bar{X}$, $p^{(t)}$, and $q^{(t)}$ are observed, MLEs of u, v are well-defined, since $p^{(t)}$ and $q^{(t)}$ form the sufficient statistics of model (4). More importantly, our result connects IPF to (4), by showing that its MLEs are exactly the IPF solution to the matrix balancing problem, with (3) equivalent to the negative log-likelihood. This connection also partially inspired the name "biproportional Poisson", as an alternative name of IPF is "biproportional fitting" (Bacharach, 1965).

Theorem 3.1. *Assume that the matrix balancing problem with $\bar{X}$, $p^{(t)}$, and $q^{(t)}$ has a finite solution (D^0, D^1). Then d^0 and d^1 are limits of the IPF iterations if and only if $\hat{u} = \log d^0$ and $\hat{v} = -\log d^1$ are solutions to the maximum likelihood estimation problem of (4) given $\bar{X}$, $p^{(t)}$, and $q^{(t)}$, with log-likelihood $\ell \equiv -g(u, v)$ in (3) modulo constants:*

$$\ell(u, v) = \sum_i p_i^{(t)} u_i - \sum_j q_j^{(t)} v_j - \sum_{ij} \bar{X}_{ij} e^{u_i - v_j}. \quad (5)$$

Moreover, maximizing $\ell(u, v)$ is equivalent to the maximum likelihood estimation of a Poisson regression model, with $p^{(t)}, q^{(t)}$ as the sufficient statistics.

We prove Theorem 3.1 in Appendix B.3. The core of our result lies in identifying a model whose log-likelihood is equivalent to $-g(u, v)$, since IPF minimizes $g(u, v)$ in (3). Our result is closely related to Qu et al. (2023), who recently established connections between IPF and choice modeling, and observed that (3) reduces to the maximum likelihood objective of a general class of choice models.

Notably, our theorem does not require assumptions about how the network evolves over time, since our model includes scaling factors u and v per time step, which IPF directly estimates. However, performing network inference in this decoupled fashion potentially leaves out additional information, which is that $\bar{X} = \sum_{t=1}^{T} X^{(t)}$. We show in Appendix B.4 that performing network inference with this constraint reduces to the decoupled problems under the following mild stationarity assumption for some constant c:

$$\sum_{t=1}^{T} e^{u_i(t) - v_j(t)} \approx c, \quad (6)$$

for all i, j where $\bar{X}_{ij} > 0$. We also verify that the stationarity assumption approximately holds on real-world data (Appendix E.3.3), thus justifying the decoupled approach. As two additional results, in Appendix B.5, we prove necessary and sufficient conditions for IPF to recover $X^{(t)}$ exactly, and in Appendix B.6, we prove that among a larger class of generalized linear models, the Poisson model is the *unique* one where the MLE is the IPF solution — so from the perspective of IPF, our generative model is canonical.

Implications of our result. Our model allows us to interpret IPF through the lens of a generative network model. Since, in relevant applications, $X_{ij}^{(t)}$ is often the number of visits from node i to j, (4) is consistent with queuing theory where rare events are modeled with a Poisson process due to the memory-less property. In addition, in the Poisson parameters $e^{u_i} \bar{X}_{ij} e^{-v_j}$, u_i can be interpreted as the emission intensity of node i, and $-v_j$ as the absorption intensity of node j. For example, in mobility networks between residential neighborhoods and public places (Chang et al., 2021a), u captures when each neighborhood is likelier to go out (e.g., younger populations more at night) and $-v$ captures each place's visit propensity (e.g., schools visited more during the day while bars visited more at night). Notably, the biproportional form of the model assumes that there are not time-varying interactions between rows and columns (e.g., if a place offers special discounts for seniors at this time, attracting neighborhoods with large senior populations), making explicit one of the key assumptions of using IPF in this network inference setting. Moreover, since $d_i^0 \bar{X}_{ij} d_j^1$ from IPF corresponds to $e^{u_i} \bar{X}_{ij} e^{-v_j}$, we can interpret the matrix inferred by IPF as estimating the *expected values* of the network-generating process.

We also note connections of our model to (pseudo-)Poisson maximum likelihood regression in economics (Gourieroux et al., 1984), including for models of trade (Silva & Tenreyro, 2006) and matching (Galichon & Salanié, 2022a;b). In particular, Theorem 3.1 implies that estimating the biproportional model is equivalent to solving a pseudo-Poisson maximum likelihood problem. Consequently, the results of Gourieroux et al. (1984) guarantee that the minimizers of Equation (3) are consistent estimators of u, v even under *misspecifications* of the distribution and homoskedasticity of $X_{ij}^{(t)}$ given $\lambda_{ij} = e^{u_i} \bar{X}_{ij} e^{-v_j}$, highlighting the robustness of our biproportional Poisson model.

Defining an explicit model also yields several advantages. First, the equivalence between the IPF solution and this model's MLE enables us to analyze IPF estimates using tools from statistical theory. In Section 4, we develop bounds on the MLE's estimation error and establish that finite MLEs exist with high probability. Second, our model clarifies previously implicit assumptions when using IPF to infer dynamic networks, such as the stationarity assumption (6) or the lack of time-varying interactions. Making such assumptions explicit allows practitioners to evaluate how reasonable the assumptions are given their domain and data; for example, we test several model assumptions on real-world bikeshare data (Appendix E.3.3). Third, defining an explicit model reveals natural ways to extend the model,

such as non-Poisson distributions or interaction terms between rows and columns. These extensions allow us to test IPF under model misspecification (Appendix E.1.3) and create future opportunities for studying how changes in the model map back to changes in IPF. Finally, the model enables new empirical analyses of IPF (Section 6), such as quantifying uncertainty in the parameter estimates and evaluating IPF's estimation of the network parameters, instead of only evaluating IPF's error on the marginals, which is how IPF tends to be evaluated (Lovelace et al., 2015).

4. Statistical Theory of Biproportional Poisson

We have shown in Theorem 3.1 that when the matrix balancing problem has finite solutions, IPF recovers them as MLEs of our biproportional Poisson model. Two important questions remain: how "good" are these MLEs, in terms of their estimation error relative to the true model parameters, and how often can we guarantee that the MLEs, solutions to the matrix balancing problem, are finite? In this section, we develop statistical theory to answer these questions.

4.1. Structure-dependent MLE Error Bounds

Intuitively, the more "well-connected" a biproportional Poisson network, the better quality its MLEs. We quantify network connectivity through the *Fiedler eigenvalue* (Fiedler, 1973), which is the second-smallest eigenvalue $\lambda_{-2}(\mathcal{L})$ of the graph Laplacian $\mathcal{L} := \mathcal{D}(A\mathbf{1}) - A$, where A is the (weighted) adjacency matrix and $\mathcal{D}(\cdot)$ denotes the diagonalization of a vector. In this paper, $\mathcal{L}$ is the graph Laplacian of the *weighted bipartite* graph G_b induced by $\bar{X}$ with $A := \begin{bmatrix} 0 & \bar{X} \\ \bar{X}^T & 0 \end{bmatrix}$. The Fiedler eigenvalue is frequently used in graph theory and distributed optimization to measure graph connectivity (Spielman, 2012). In recent years, its importance for the algorithmic and statistical efficiencies of Luce choice model estimation *vis-à-vis* the topology of comparison structures has been extensively studied (Shah et al., 2015; Vojnovic & Yun, 2016; Seshadri et al., 2020; Vojnovic et al., 2020; Hendrickx et al., 2020; Bong & Rinaldo, 2022). It is important to note that the Fiedler eigenvalue used in our paper is based on a *different* graph than in the choice literature, which are constructed from choice data and are not bipartite. An exception is Qu et al. (2023), who provided linear convergence analyses of IPF quantified by the *same* $\lambda_{-2}(\mathcal{L})$ as in this paper.

We now provide error bounds for normalized MLEs of the biproportional Poisson model quantified by the Fiedler eigenvalue, whenever the true parameters and MLEs are bounded by some constant B.

Theorem 4.1. *Suppose that the biproportional Poisson model* (4) *holds with ground truth parameters* u^*, v^*. *Suppose* $(\hat{u}, \hat{v})$ *is a maximizer of the log-likelihood* (5) *and that we have the normalization condition* $(\hat{u} - u^*, \hat{v} - v^*) \in \mathbf{1}_{m+n}^{\perp}$ *and* $\|(\hat{u}, \hat{v}, u^*, v^*)\|_\infty \leq B$. *Then with* $\kappa = \sum_{ij} e^{u_i^*} \bar{X}_{ij} e^{-v_j^*}$ *the total rate, in expectation we have*

$$\mathbb{E}\left[\|(\hat{u} - u^*, \hat{v} - v^*)\|^2 \mathbb{1}_{\mathcal{B}}\right] \leq \frac{8e^{4B}\kappa}{\lambda_{-2}(\mathcal{L})^2}, \tag{7}$$

where $\mathbb{1}_{\mathcal{B}}$ *is the event that the MLE is bounded above by* B.

A similar tail bound is given and proven in Theorem C.2. The dependence of the error bounds on the Fiedler eigenvalue $\lambda_{-2}(\mathcal{L})$ is clear: the more well-connected the bipartite graph G_b (induced by $\bar{X}$), the larger $\lambda_{-2}(\mathcal{L})$, which in turn improves the estimation quality. Since $\lambda_{-2}(\mathcal{L}) > 0$ if and only if G_b is connected, the bounds blow up when G_b becomes disconnected. Indeed, the biproportional Poisson model is not identified in this case (see Remark C.3). Analytically, $\lambda_{-2}(\mathcal{L})$ also quantifies the *strong concavity* of the log-likelihood function (5), which provides additional justification for its prominence in the bounds.

Given recently established connections between choice modeling and matrix balancing and the well-known structure-dependent estimation error bounds in the choice literature (Shah et al., 2015; Seshadri et al., 2020; Hendrickx et al., 2020), our error bounds in Theorem 4.1 can be viewed as natural analogs of such results for the biproportional Poisson model. As with those results, the assumption that the ground truth parameters and the MLEs are bounded by some B is standard. However, we note that since the dimension of observations $p^{(t)}, q^{(t)}$ is on the same order as the parameter dimension (both $m + n$), the biproportional Poisson model corresponds to a high-dimensional setting by design. Consequently, unlike bounds in the choice literature, there is no explicit dependence on the "sample size" in our bounds. We provide a simple example to illustrate this point.

Example with accurate recovery: complete graph. Suppose that $\bar{X}$ is the $m \times n$ all-ones matrix. Then $\kappa = \Theta(nm)$ and $\lambda_{-2}(\mathcal{L}) = \min(n, m)$ so the right hand side of (7) is of order $\Theta(\max(n/m, m/n))$. So if e.g. $n = m$, then the total error for recovering the *entire vector* (u^*, v^*) is $\mathcal{O}(1)$. Equivalently, the average error *per coordinate* of u^*, v^* is $\frac{1}{2n}\|(\hat{u} - u^*, \hat{v} - v^*)\|^2 = \mathcal{O}(1/n)$.

Bound improves with growing SNR. A key feature of the error bound is that it improves by a factor of $1/c$ when the base matrix $\bar{X}$ is *scaled up* by a constant $c > 1$, since both κ and $\lambda_{-2}(\mathcal{L})$ are scaled by c. This feature can be understood as a result of improved "signal-to-noise" ratio (SNR), since the Poisson rate $e^{u_i^*} \bar{X}_{ij} e^{-v_j^*}$ scales with $\bar{X}$.

Impact of sparsity. As graph connectivity is related to the sparsity of the graph, our results can inform us on how the sparsity of the network impacts estimation quality of IPF. In Appendix E.1.2, we evaluate the quality of MLEs with synthetic data under different sparsity rates $r \in [0, 1)$. One

implication of Theorem 4.1 is that, with high probability,

$$\|(e^{\hat{u}} - e^{u^*}, e^{-\hat{v}} - e^{-v^*})\|_2 = \mathcal{O}\left(\frac{1}{\sqrt{1-r}}\right),$$

which matches the observed deterioration of estimation quality in Figure 2 as sparsity r increases from 0 to 1.

In Remark C.3, we provide more discussion on the error bounds in Theorem 4.1, including optimality of the dependence on κ and necessity of $\lambda_{-2}(\mathcal{L})$ and $e^{\Theta(B)}$. Despite the usefulness of the error bounds we develop in Theorem 4.1, an important technical question remains on the existence of *finite* MLEs. Even under correct specification, a (random) dataset may not yield a well-defined maximum likelihood problem, i.e., no finite maximizer of (5) exists. We next address this issue and identify a sufficient condition that guarantees that finite MLEs exist with high probability.

4.2. Well-posedness of Maximum Likelihood Estimation

Under the biproportional Poisson model (4), the maximum likelihood estimation problem on $\bar{X}, p^{(t)}, q^{(t)}$ may not have a finite solution. As is well-known and discussed in Qu et al. (2023), the corresponding matrix balancing problem has a finite solution if and only if there exists a matrix with *exactly* the same zero patterns as $\bar{X}$ and has marginals $p^{(t)}, q^{(t)}$. Under correct specification of the biproportional Poisson model, the matrix $X^{(t)}$ has the right marginals $p^{(t)}, q^{(t)}$. If all Poisson entries $X_{ij}^{(t)} > 0$, $X^{(t)}$ provides a certificate for the existence of a finite solution to the matrix balancing problem hence finite MLEs. However, as soon as any $X_{ij}^{(t)}$ equals 0, $X^{(t)}$ will have additional zero entries than $\bar{X}$, and the bipartite network induced by $X^{(t)}$ could become disconnected. However, there could still exist *another* matrix that solves the matrix balancing problem, yielding a finite MLE. Our task is to show that this happens often.

A similar challenge exists in the choice setting. When some subset of items are always preferred over its complement, no positive MLE exists in the Luce choice modeling framework. Recently, Bong & Rinaldo (2022) provided a simple sufficient condition on the Fisher information matrix in the Bradley–Terry–Luce model which guarantees that this event rarely happens. Our next theorem can be viewed as an analog of their result for the biproportional Poisson model.

Theorem 4.2. *Let $\mathcal{L}^* := -\nabla^2 \ell(u^*, v^*)$ be the Hessian of the negative log-likelihood in (5) evaluated at the true parameters (u^*, v^*), and suppose its second smallest eigenvalue satisfies*

$$\lambda_{-2}(\mathcal{L}^*) \geq 2\log(m+n). \tag{8}$$

Then with probability at least $1 - 2/\sqrt{m+n}$, the maximum likelihood estimation of the model (4) has a unique normalized finite solution, and IPF converges to this solution.

Note that the sufficient condition (8) is stated in terms of the Hessian evaluated at the *true* parameters (u^*, v^*), which is essentially the Fisher information matrix modulo a constant factor. When is (8) satisfied? As discussed before, for complete graphs $\lambda_{-2}(\mathcal{L}^*) = \mathcal{O}(\min\{m, n\})$, so (8) is satisfied. As another example, an Erdös-Rényi graph with parameter p of order $\Omega(\frac{\log(m+n)}{m+n})$ also satisfies (8) with high probability. See for example Bong & Rinaldo (2022).

Our results in this section provide useful insights about the MLEs of our biproportional Poisson model. However, IPF convergence is a prerequisite for IPF to recover these MLEs. Now, we move onto the next natural and important question: what should one do in practice if IPF does *not* converge?

5. Guaranteeing IPF Convergence

IPF non-convergence tends to occur when there are many zeros in the inputs (Wong, 1992), and such sparsity is very common in real-world network data, partially due to missing values. In fact, we show that, when trying to infer $X^{(t)}$ from $p^{(t)}$, $q^{(t)}$, and $\bar{X}$, for some aggregated time period that includes t, IPF will not converge *only if* there are missing entries in the inputs (Corollary D.1). For example, in mobility data, true visits may be missed due to noisy GPS signals, populations not carrying cell phones (Coston et al., 2021), or data "clipping" where low values are replaced with zeros to preserve privacy (SafeGraph, 2020a). We discuss these mechanisms in depth in Appendix E.2.

Thus, we approach IPF non-convergence from the perspective of missing data: specifically, in initial matrix X, and we resolve non-convergence by adding edges to X. This is a similar view to the typical solution for IPF non-convergence, which is to replace all zeros in X with very small amounts (Lomax & Norman, 2015). However, the typical approach may result in unrealistic positive entries, such as drivers under the age of 16 when inferring joint demographics (Lovelace et al., 2015) or nonexistent routes in transportation networks. Furthermore, in our setting, replacing all zeros with positive entries completely alters the sparsity structure of the inferred network, which can greatly affect downstream results, such as modeling epidemics (Wang et al., 2003). Instead, we introduce a new algorithm, `ConvIPF`, that guarantees IPF convergence by adding edges while minimizing changes to the network structure, where we explore two different definitions of network change below. Even though we focus on network inference, `ConvIPF` can be used in any application of IPF where non-convergence may be caused by missing values in the initial matrix X.

Overview of `ConvIPF`. Two equivalent conditions that define when IPF converges are (Pukelsheim, 2014):

1. There exists a matrix Y with row sums p and column

sums q such that $Y_{ij} = 0$ wherever $X_{ij} = 0$.
2. For all row subsets $S \subseteq [m]$, $\sum_{i \in S} p_i \leq \sum_{j \in N_X(S)} q_j$, where $N_X(S)$ represents the set of columns connected to S in X.

Condition (1) yields an efficient algorithm, which we call MAX-FLOW, for testing whether IPF will converge. The algorithm, as described in Idel (2016) and Appendix D.1, requires one round of max-flow on a graph constructed from the IPF inputs. If the resulting flow is equal to $\sum_i p_i$, then IPF converges. Condition (2) is useful because it allows us to diagnose why IPF is not converging: if IPF does not converge, there must be at least one "blocking set" of rows for which the condition is violated.

The key idea of ConvIPF is that we can unblock a blocking set of rows by adding new edges in X for those rows, but we seek edge additions that modify X *as little as possible*. After we unblock one blocking set, there may be more. So, ConvIPF iteratively identifies a blocking set and modifies X to unblock it, until there are no blocking sets remaining. Our algorithm thus repeats three subroutines, MAX-FLOW, BLOCKING-SET, and MODIFY-X, until IPF converges:

1. Run MAX-FLOW to test for convergence. If IPF converges, then the algorithm is finished. If IPF does not converge, move on to Step 2.
2. Since IPF does not converge, run BLOCKING-SET to identify a blocking set of rows, S.
3. Run MODIFY-X to unblock S by minimally adding edges to X.

ConvIPF must terminate since (i) it is always possible to unblock a row set (by connecting it to all columns) and (ii) an unblocked row set cannot become blocked through subsequent edge additions. Furthermore, even though there are 2^m possible row subsets, the algorithm will terminate within mn iterations, since IPF converges when all entries in X are positive (Pukelsheim, 2014) and each MODIFY-X adds at least one positive entry to X. Thus, as long as each subroutine runs in polytime, the entire algorithm runs in polytime. However, BLOCKING-SET and MODIFY-X both try to solve combinatorial problems, so the challenge is how to efficiently solve each one. We provide brief sketches of each in this section, with details in Appendix D.

BLOCKING-SET: identifying a blocking set. Given inputs X, p, and q, where we know IPF does not converge, this subroutine identifies a blocking set of rows S for which Condition (2) is violated. The naive approach to iterate through all subsets until a violation is found, but this approach is extremely inefficient, as there are 2^m possible subsets. Instead, our subroutine imports ideas from constricted sets in bipartite matching (Hall, 1935; Easley & Kleinberg, 2010) to design a much more efficient algorithm. First, construct a bipartite graph B where the nodes are the rows and columns of X and they are connected wherever $X_{ij} > 0$. From running MAX-FLOW to test for convergence, we have flow values for each row/column. Since IPF does not converge, there must be at least one row i whose flow is less than its capacity, p_i. Run the following variant of breadth-first search (BFS) on B starting from node n_i. When progressing from a column node n_C to its neighboring row node n_R, only include row nodes where $n_R \to n_C$ has non-zero flow in MAX-FLOW. When progressing from row nodes to column nodes, include all (unvisited) neighbors. When BFS terminates, the set of row nodes visited forms a blocking set, which we prove in Appendix D.2.

MODIFY-X: unblocking a blocking set. Given a blocking set S, this subroutine minimally adds edges to X to unblock S. Let X^K represent the modified X after adding new edges K and let $f(X, X^K)$ represent the change in X that we are trying to minimize. Then, our goal is to find K^* that minimizes $f(X, X^{K^*})$, subject to S being unblocked under X^{K^*}, i.e., $\sum_{i \in S} p_i \leq \sum_{j \in N_{X^{K^*}}(S)} q_j$. We consider two natural definitions of $f(X, X^K)$.

1. Number of new edges. Let $\bar{N}_X(S)$ represent the set of columns *not* connected to S in X, and let δ represent the gap in marginals, $\delta := \sum_{j \in N_X(S)} q_j - \sum_{i \in S} p_i$. Take the top-$k$ columns in $\bar{N}_X(S)$, ordered by q_j in descending order, that satisfy $\sum_{j=1}^{k} q_j \geq \delta$. Then, any set of edges between a row in S and these k columns will unblock S, while minimizing the number of new edges added.

2. Change in λ_1. A more nuanced objective minimizes the change in the relevant spectral properties of X. Motivated by the application of epidemic spread, recall that the epidemic threshold of a network is closely related to the largest eigenvalue λ_1 of its adjacency matrix (Wang et al., 2003) and attempts to reduce spreading aim to minimize λ_1 through edge removals (Saha et al., 2015; Li et al., 2023). So, to preserve a "similar" network from a spreading standpoint, we seek to minimize change in λ_1. We show in Appendix D.3 that, with reasonable approximations of change in λ_1 (Tong et al., 2012), unblocking S while minimizing change in λ_1 reduces to the following: first, find the row i^* in S with the smallest $\vec{u}_1(i)$, where $\vec{u}_1$ and $\vec{v}_1$ are the left and right eigenvectors of $\lambda_1(X)$, respectively. Then, solve an integer linear program to find the set of columns $J \subseteq \bar{N}_X(S)$ that minimize $\sum_{j \in J} \vec{v}_1(j)$, subject to $\sum_{j \in J} q_j \geq \delta$. The set of new edges is $\{(i^*, j) | j \in J\}$, which unblocks S and approximately minimizes change in λ_1.

6. Experiments with Data

We now summarize our experiments with synthetic and real-world data. Our experiments reveal the utility of our theoretical and algorithmic contributions, and demonstrate

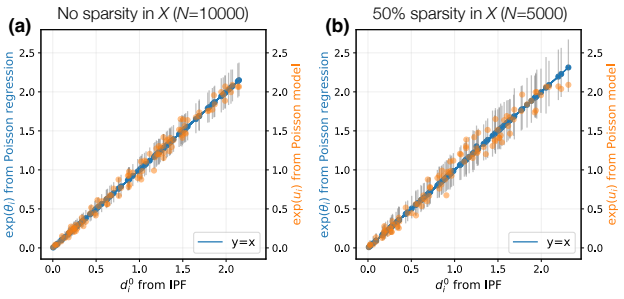


Figure 1. Comparing inferred parameters from IPF (x-axis) against inferred parameters from Poisson regression (left y-axis, blue) and true parameters from Poisson model (right y-axis, orange). Grey bars indicate 95% CIs from Poisson regression. N represents the number of nonzero entries in X, so N is halved with 50% sparsity. Under both networks, estimated parameters from IPF and Poisson regression are perfectly aligned (Theorem 3.1), but their estimation quality worsens with greater sparsity (Theorem 4.1).

IPF's capability to infer networks in practice.

Testing IPF with synthetic data. In our first set of experiments, we use synthetic data generated from our network model (4) to confirm the correspondence between IPF and Poisson regression. In Figure 1, we show that the parameters inferred by IPF and Poisson regression align perfectly, validating Theorem 3.1. Furthermore, our Poisson model enables us to quantify uncertainty in the parameter estimates under the model, adding valuable interpretability missing from IPF output alone. We display 95% confidence intervals (CIs) in Figure 1; while these CIs are only asymptotically valid, they provide useful measures of uncertainty.

We also test IPF in more difficult settings, such as with increased sparsity in $\bar{X}$. We generate $\bar{X}$ with a given sparsity rate r by randomly selecting $r \cdot mn$ entries and setting them to 0, and test IPF on varying sparsity rates. As expected, we find that the widths of the CIs increase with greater sparsity (Figure 1b), since greater sparsity results in fewer Poisson observations to fit our model (4). We also find that, despite matching the target marginals in all cases, the ℓ_2 distance between IPF's inferred parameters and the true parameters increases quickly as we increase the sparsity in $\bar{X}$ (Figure 2, right). These findings align with Theorem 4.1, where we showed that the bound on the MLEs' expected estimation error (7) improves as $\bar{X}$ becomes more well-connected.

We also use synthetic data to test IPF under model misspecification and find that IPF is reasonably robust to model modifications in this network inference setting (Appendix E.1.3). When the model is correctly specified (i.e., data is generated from our biproportional Poisson model), the cosine similarity between the true network $X^{(t)}$ and IPF estimate of the network $\hat{X}^{(t)} = D^0 \bar{X} D^1$ is, on average, 0.911 (Table E.1). If we replace the Poisson with an exponential distribution, the cosine similarity only decreases to 0.855; if we replace it with a negative binomial distribution ($\gamma = 0.5$), it decreases to 0.843. IPF is similarly robust when we test it on data from an "interaction model" (42), which allows $X_{ij}^{(t)}$ to additionally depend on interaction terms between rows and columns, such as distance (Figure E.5).

IPF convergence on mobility data. To test IPF convergence, we use mobility data from SafeGraph (SafeGraph, 2020a). Here, we seek to infer the hourly visit network from neighborhoods to points-of-interest (POIs), so the marginals represent hourly total visits from neighborhoods and to POIs, and $\bar{X}$ represents the time-aggregated network. SafeGraph data provides a real-world example where only the hourly marginals and time-aggregated network are provided, with missing data in the time-aggregated network due to under-reporting and data clipping (see Appendix E.2). We use mobility data from the Richmond metro area in Virginia, which has 9917 POIs and 1098 neighborhoods (Chang et al., 2021b). Despite aggregating over 10 months (January to October 2020), $\bar{X}$ remains sparse: only 8% of its entries are non-zero. From running IPF on two days (48 hours), we find that IPF does not converge for three nighttime hours when POI marginals are particularly sparse.

Using one of these hours—2AM on March 2, 2020—as an example, we apply our algorithm `ConvIPF` and evaluate the change in $\bar{X}$. We compare our solutions to the typical solution for IPF non-convergence, which is to replace all zeros with a very small value ϵ (Lomax & Norman, 2015; Lovelace et al., 2015). If minimizing the number of new edges added, `ConvIPF` only adds 2 edges, while the typical solution results in 10012193 new edges. If minimizing the change in λ_1, `ConvIPF` only changes it by $5.60 \cdot 10^{-9}$, while the typical solution changes λ_1 by 30.04. The magnitude of reduction achieved by `ConvIPF` is similar for the other two hours (Table E.2). Furthermore, we find that the choice of objective in `ConvIPF` makes a difference: choosing to minimize the number of edges always results in the fewest number of edges added and choosing to minimize the change in λ_1 always results in the smallest change in λ_1, by an order of magnitude compared to the other `ConvIPF` variants. Finally, `ConvIPF` terminates quickly in practice, typically requiring 5 or fewer iterations.

Ground-truth networks from bikeshare data. Using data from New York City's Citibike system, we can construct *ground-truth* hourly networks that record the number of bike trips between stations. Acquiring ground-truth networks enables us to test IPF's ability to infer these networks, compared to baselines, which assesses the downstream utility of our work in bridging IPF to this network inference problem. We test IPF's inferred networks, given hourly

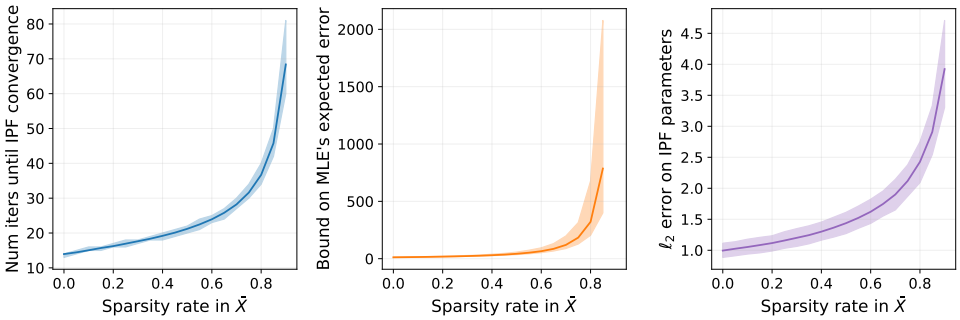


Figure 2. Comparing sparsity rate in $\bar{X}$ to number of IPF iterations (left), bound on MLE's expected estimation error, without constants (middle), and observed ℓ_2 error of IPF estimates (right). Lines represent mean and shaded region represents 95% CIs over 1000 trials.

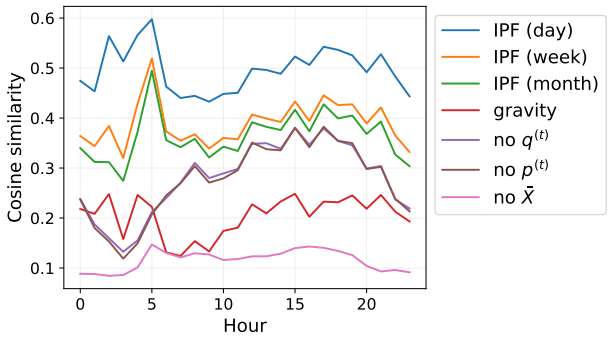


Figure 3. Cosine similarity between ground-truth hourly networks from bikeshare data and inferred networks from IPF and baselines.

marginals and time-aggregated networks at the month-, week-, and day-level. As a baseline, we try the classic gravity model, which assumes that the amount of travel between two regions is proportional to their inverse distance (Zipf, 1946; Erlander & Stewart, 1990). We fit the "doubly constrained gravity model" (Navick & Furth, 1994), which is equivalent to running IPF on a distance matrix, instead of the time-aggregated network, and the hourly marginals (Appendix E.3.1). This baseline enables us to test how much additional information the time-aggregated network provides, beyond what is captured by common geographical assumptions. We also test other baselines:

- An ablation that removes $\bar{X}$ and distributes $\sum_i p_i^{(t)}$ proportional to $p_i^{(t)} q_j^{(t)}$;
- An ablation that removes $q^{(t)}$ and distributes $p_i^{(t)}$ within each row proportional to $\bar{X}_{ij} / \sum_j \bar{X}_{ij}$;
- An ablation that removes $p^{(t)}$ and distributes $q_j^{(t)}$ within each column proportional to $\bar{X}_{ij} / \sum_i \bar{X}_{ij}$.

We present main results in Figure 3. First, we find that IPF strongly outperforms the gravity model, with a 78% improvement in cosine similarity even when using the month-aggregated network. Second, IPF also outperforms the ablation baselines: when $\bar{X}$ is the month-aggregated network, IPF outperforms the ablation without $\bar{X}$ by 214% and the ablations without $p^{(t)}$ or $q^{(t)}$ by around 31%. Third, finer temporal granularity in the time-aggregated network improves IPF's performance, but the relative improvement is much larger from week to day than from month to week. This suggests that bike trips vary more at a daily scale within the week (e.g., weekday vs. weekend) than a weekly scale within the month. Overall, these results demonstrate the effectiveness of IPF on this difficult network inference problem as well as the benefits of more granular data.

7. Conclusion

In this work, we have established a statistical framework for using IPF to infer a dynamic network from its marginals. Our primary contribution is the biproportional Poisson model. We show that IPF uniquely recovers the MLEs of this model and derive statistical results on the MLEs. Our model not only provides justification for using IPF to infer dynamic networks, but also enables new analyses of IPF, clarifies implicit network assumptions, and creates natural paths for testing alternate models and misspecification. We also introduce `ConvIPF`, a principled algorithm that guarantees IPF convergence on sparse data, which can be broadly applied wherever IPF is used. Our empirics confirm our theoretical results, demonstrate the value of our modeling and algorithmic contributions, and reveal IPF's ability to infer networks in practice. Given the long history of IPF and richness of the network inference problem, connecting them generates many avenues for future research. Some future directions include characterizing how IPF's ability to infer networks varies based on the network's temporal dynamics (e.g., periodicity), designing variants of IPF that outperform traditional IPF on this network inference problem, and providing a statistical interpretation of our convergence algorithm. Lastly, we note that the biproportional Poisson model is applicable more generally to other instances of the matrix balancing problem beyond network settings.

Acknowledgements

S.C. was supported in part by an NSF Graduate Research Fellowship, the Meta PhD Fellowship, and NSF award CCF-1918940. F.K. was supported in part by NSF award CCF-1704417, NSF award IIS1908774, and N. Anari's Sloan Research Fellowship. Z.Q. was supported in part by ONR Grant N00014-19-1-2468 and NSF CAREER Award IIS-2143176. J.U. was supported in part by NSF CAREER Award IIS-2143176. J.L. was supported in part by NSF awards OAC-1835598, CCF-1918940, DMS-2327709, and Stanford Data Applications Initiative. The authors thank Alfred Galichon, Emma Pierson, and Arjun Seshadri for helpful discussions and comments as well as Devin Caughey, Guido Imbens, Pang Wei Koh, and members of Jure Leskovec's lab for helpful feedback on early versions of this work.

Impact Statement

The primary goal of our work is to advance understanding of IPF and dynamic network inference, with contributions to machine learning and applied statistics. However, our work is deeply motivated by problems of societal importance, such as epidemic response and transportation planning. Prior work has required inferring dynamic networks from their marginals in order to model COVID-19 spread over detailed mobility networks, inform pandemic interventions, and analyze socioeconomic disparities in infection rates (Chang et al., 2021a;b; Chaudhuri et al., 2022; Li et al., 2023; Alimohammadi et al., 2023). This dynamic network inference problem also appears in transportation (Carey et al., 1981), communication (Kruithof, 1937), and human migration (Plane, 1982; Pham & Komiyama, 2022). Our work provides broadly applicable insights for practitioners in all of these domains, allowing them to fully leverage the empirical advantages of IPF with a deeper understanding of its assumptions and behavior (e.g., when it is justified, how to interpret its estimates, how to ensure convergence). Another advantage of IPF is *privacy*: it enables practitioners to estimate detailed networks while data providers only need to release aggregated data, i.e., only the time-varying marginals and the time-aggregated network. Due to such limited information, IPF will only reconstruct the true network under very restrictive assumptions (Theorem B.1), but otherwise, it will infer a *realistic* network that is correlated with the real network, enabling analyses that require time-varying networks without revealing private information.

References

Aas, E. Limit points of the iterative scaling procedure. *Ann Oper Res*, 215:15–23, 2014.

Adamczak, R. A tail inequality for suprema of unbounded empirical processes with applications to markov chains. 2008.

Akagi, Y., Nishimura, T., Kurashima, T., and Toda, H. A fast and accurate method for estimating people flow from spatiotemporal population data. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI '18)*, 2018.

Alimohammadi, Y., Borgs, C., van der Hofstad, R., and Saberi, A. Epidemic forecasting on networks: Bridging local samples with global outcomes. *Working paper*, 2023.

Anderson, G. W., Guionnet, A., and Zeitouni, O. *An introduction to random matrices*. Number 118. Cambridge university press, 2010.

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

Bennett, G. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45, 1962.

Bishop, Y. M. M. and Fienberg, S. E. Incomplete two-dimensional contingency tables. *Biometrics*, 25(1):119–128, 1969.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. *Discrete Multivariate Analysis*. Springer, New York, NY, 1974.

Bong, H. and Rinaldo, A. Generalized results for the existence and consistency of the mle in the bradley-terry-luce model. In *International Conference on Machine Learning*, pp. 2160–2177. PMLR, 2022.

Bregman, L. M. 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.

Carey, M., Hendrickson, C., and Siddharthan, K. A method for direct estimation of origin/destination trip matrices. *Transportation Science*, 15(1):32–49, 1981.

Carlier, G. On the linear convergence of the multimarginal sinkhorn algorithm. *SIAM Journal on Optimization*, 32 (2):786–794, 2022.

Chamakh, L., Gobet, E., and Szabó, Z. Orlicz random fourier features. *The Journal of Machine Learning Research*, 21(1):5739–5775, 2020.

Chang, S., Pierson, E., Koh, P. W., Gerardin, J., Redbird, B., Grusky, D., and Leskovec, J. Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589(7840):82–87, 2021a.

Chang, S., Wilson, M. L., Lewis, B., Mehrab, Z., Dudakiya, K. K., Pierson, E., Koh, P. W., Gerardin, J., Redbird, B., Grusky, D., Marathe, M., and Leskovec, J. 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)*, 2021b.

Chaudhuri, S., Kasibhatla, P., Mukherjee, A., Pan, W., Morrison, G., Mishra, S., and Murty, V. K. Analysis of overdispersion in airborne transmission of covid-19. *Physics of Fluids*, 34(051914), 2022.

Chen, W., Sun, X., Zhang, J., and Zhang, Z. Network inference and influence maximization from samples. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

CitiBike. System data, 2023. Available at https://citibikenyc.com/system-data.

Coston, A., Guha, N., Ouyang, D., Lu, L., Chouldechova, A., and Ho, D. E. Leveraging administrative data for bias audits: Assessing disparate coverage with mobility data for covid-19 policy. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT'21)*, 2021.

Csiszár, I. I-divergence geometry of probability distributions and minimization problems. *Ann. Probab.*, 3(1):146–158, 1975.

Csiszár, I. and Tusnády, G. Information geometry and alternating minimization procedures. *Statistics and Decisions, Supplement Issue*, 1:205–237, 1984.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Demange, G., Gale, D., and Sotomayor, M. Multi-item auctions. *Journal of Political Economy*, 94(4), 1986.

Deming, W. E. and Stephan, F. F. 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.

Dewey. Safegraph data for academic research, 2023. Available at https://www.deweydata.io/data-partners/safegraph.

Easley, D. and Kleinberg, J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

Erlander, S. and Stewart, N. F. *The gravity model in transportation analysis : theory and extensions*. VSP, Utrecht, Netherlands, 1990.

Fiedler, M. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

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

Franklin, J. and Lorenz, J. On the scaling of multidimensional matrices. *Linear Algebra and its applications*, 114: 717–735, 1989.

Galichon, A. and Salanié, B. Cupid's invisible hand: Social surplus and identification in matching models. *The Review of Economic Studies*, 89(5):2600–2629, 2022a.

Galichon, A. and Salanié, B. Estimating separable matching models. *arXiv preprint arXiv:2204.00362*, 2022b.

Gardner, W., Mulvey, E. P., and Shaw, E. C. Regression analyses of counts and rates: Poisson, overdispersed poisson, and negative binomial models. *Psychological Bulletin*, 118(3):392–404, 1995.

Gomez-Rodriguez, M., Leskovec, J., and Krause, A. Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data*, 5(4):1–37, 2012.

Gourieroux, C., Monfort, A., and Trognon, A. Pseudo maximum likelihood methods: Applications to poisson models. *Econometrica: Journal of the Econometric Society*, pp. 701–720, 1984.

Gross, D. and Nesme, V. Note on sampling without replacing from a finite collection of matrices. *arXiv preprint arXiv:1001.2738*, 2010.

Hall, P. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1), 1935.

Hallac, D., Park, Y., Boyd, S., and Leskovec, J. Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*, 2017.

Harremoës, P. Binomial and poisson distributions as maximum entropy distributions. *IEEE Transactions on Information Theory*, 47(5):2039–2041, 2001.

Hendrickx, J., Olshevsky, A., and Saligrama, V. Minimax rate for learning from pairwise comparisons in the btl model. In *International Conference on Machine Learning*, pp. 4193–4202. PMLR, 2020.

Holford, T. R. The analysis of rates and of survivorship using log-linear models. *Biometrics*, 36(2):299–305, 1980.

Huang, X., Lu, J., Gao, S., Wang, S., Liu, Z., and Wei, H. Staying at home is a privilege: Evidence from finegrained mobile phone location data in the united states during the covid-19 pandemic. *Annals of the American Association of Geographers*, 2021.

Huber, P. J. The behavior of maximum likelihood estimates under nonstandard conditions. *Berkeley Symp. on Math. Statist. and Prob.*, 5.1:221–233, 1967.

Idel, M. A review of matrix scaling and sinkhorn's normal form for matrices and positive maps. *arXiv*, 2016.

Ireland, C. T. and Kullback, S. Contingency tables with given marginals. *Biometrika*, 55(1):179–188, 1968.

Iwata, T. and Shimizu, H. Neural collective graphical models for estimating spatio-temporal population flow from aggregated data. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'19)*, pp. 3935–3942, 2019.

Iwata, T., Shimizu, H., Naya, F., and Ueda, N. Estimating people flow from spatiotemporal population data via collective graphical mixture models. *ACM Trans. Spatial Algorithms Syst.*, 3(1), 2017.

Kruithof, J. Telefoonverkeersrekening. *De Ingenieur*, 52: 15–25, 1937.

Kumar, R., Tomkins, A., Vassilvitskii, S., and Vee, E. Inverting a steady-state. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM'15)*, 2015.

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

Li, D., Eliassi-Rad, T., and Zhang, H. R. Optimal intervention on weighted networks via edge centrality. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM'23)*, 2023.

Liang, G., Taft, N., and Yu, B. A fast lightweight approach to origin-destination ip traffic estimation using partial measurements. *IEEE Transactions on Information Theory*, 52(6):2634–2648, 2006.

Little, R. J. Post-stratification: A modeler's perspective. *Journal of the American Statistical Association*, 88(423): 1001–1012, 1993.

Little, R. J. and Wu, M.-M. Models for contingency tables with known margins when target and sampled populations differ. *Journal of the American Statistical Association*, 86(413):87–95, 1991.

Lomax, N. and Norman, P. Estimating population attribute values in a table: "get me started in"' iterative proportional fitting. *The Professional Geographer*, 68(3):451–461, 2015.

Lovelace, R., Birkin, M., Ballas, D., and van Leeuwen, E. Evaluating the performance of iterative proportional fitting for spatial microsimulation: New tests for an established technique. *Journal of Artificial Societies and Social Simulation*, 18(2), 2015.

Luce, R. D. Individual choice behavior. 1959.

Luo, Z.-Q. and Tseng, P. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

Marino, S. D. and Gerolin, A. An optimal transport approach for the schrödinger bridge problem and convergence of sinkhorn algorithm. *Journal of Scientific Computing*, 85(2):27, 2020.

Maystre, L. and Grossglauser, M. Choicerank: Identifying preferences from node traffic in networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, 2017.

McCord, M. R., Mishalani, R. G., Goel, P., and Strohl, B. Iterative proportional fitting procedure to determine bus route passenger origin–destination flows. *Transportation Research Record*, 2145(1):59–65, 2010.

McCullagh, P. *Generalized linear models*. Routledge, 2019.

Navick, D. S. and Furth, P. G. Distance-based model for estimating a bus route origin-destination matrix. *Transportation Research Record*, pp. 16–16, 1994.

Pham, K. H. and Komiyama, J. Strategic choices of migrants and smugglers in the central mediterranean sea. *arXiv*, 2022.

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

Pukelsheim, F. Biproportional scaling of matrices and the iterative proportional fitting procedure. *Ann. Oper. Res.*, 215:269–283, 2014.

Pukelsheim, F. and Simeone, B. On the iterative proportional fitting procedure: Structure of accumulation points and l1-error analysis. *Preprint*, 2009.

Qu, Z., Galichon, A., and Ugander, J. On sinkhorn's algorithm and choice modeling. *arXiv preprint arXiv:2310.00260*, 2023.

Renner, I. W. and Warton, D. I. Equivalence of maxent and poisson point process models for species distribution modeling in ecology. *Biometrics*, 69(1):274–281, 2013.

Rigollet, P. and Hütter, J.-C. High-dimensional statistics. *arXiv preprint arXiv:2310.19244*, 2023.

Rossi, E., Monti, F., Leng, Y., Bronstein, M. M., and Dong, X. Learning to infer structures of network games. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, 2022.

Ruschendorf, L. Convergence of the iterative proportional fitting procedure. *The Annals of Statistics*, 23(4):1160–1174, 1995.

SafeGraph. What about bias in the safegraph dataset? 2019. Available at https://www.safegraph.com/blog/what-about-bias-in-the-safegraph-dataset.

SafeGraph. Patterns. 2020a. Available at https://docs.safegraph.com/docs/monthly-patterns.

SafeGraph. Social distancing metrics. 2020b. Available at https://docs.safegraph.com/docs/social-distancing-metrics.

SafeGraph. Determining points of interest visits from location data: A technical guide to visit attribution. 2021. Available at https://www.safegraph.com/guides/visit-attribution-white-paper.

Saha, S., Adiga, A., Prakash, B. A., and Vullikanti, A. K. S. Approximation algorithms for reducing the spectral radius to control epidemic spread. In *Proceedings of the 2015 SIAM International Conference on Data Mining (SDM'15)*, 2015.

Schneider, M. H. and Zenios, S. A. A comparative study of algorithms for matrix balancing. *Operations Research*, 38(3):439–455, 1990.

Seshadri, A., Ragain, S., and Ugander, J. Learning rich rankings. *Advances in Neural Information Processing Systems*, 33:9435–9446, 2020.

Shah, N., Balakrishnan, S., Bradley, J., Parekh, A., Ramchandran, K., and Wainwright, M. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. In *Artificial intelligence and statistics*, pp. 856–865. PMLR, 2015.

Sheldon, D. R. and Dietterich, T. Collective graphical models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'11)*, 2011.

Silva, J. S. and Tenreyro, S. The log of gravity. *The Review of Economics and statistics*, 88(4):641–658, 2006.

Singh, R., Haasler, I., Zhang, Q., Karlsson, J., and Chen, Y. Inference with aggregate data: An optimal transport approach. *arXiv*, 2020.

Sinkhorn, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

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

Sion, M. On general minimax theorems. 1958.

Spielman, D. Spectral graph theory. *Combinatorial scientific computing*, 18:18, 2012.

Spielman, D. A. and Teng, S.-H. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4), 2011.

Stewart, G. W. and Sun, J.-g. *Matrix perturbation theory*. 1990.

Tong, H., Prakash, B. A., Eliassi-Rad, T., Faloutsos, M., and Faloutsos, C. Gelling, and melting, large graphs by edge manipulation. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM'12)*, 2012.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Vojnovic, M. and Yun, S. Parameter estimation for generalized thurstone choice models. In *International Conference on Machine Learning*, pp. 498–506. PMLR, 2016.

Vojnovic, M., Yun, S.-Y., and Zhou, K. Convergence rates of gradient descent and mm algorithms for bradley-terry models. In *International Conference on Artificial Intelligence and Statistics*, pp. 1254–1264. PMLR, 2020.

Wang, Y., Chakrabarti, D., Wang, C., and Faloutsos, C. Epidemic spreading in real networks: an eigenvalue viewpoint. In *22nd International Symposium on Reliable Distributed Systems*, 2003.

Wong, D. W. S. The reliability of using the iterative proportional fitting procedure. *The Professional Geographer*, 44(3):340–348, 1992.

Zipf, G. K. The P_1P_2/D hypothesis: On the intercity movement of persons. *American Sociological Review*, 11 (6), 1946.

Appendix

A. Details on IPF Implementation

In this section, we provide details on our implementation of IPF, along with pseudocode in Algorithm 1. As discussed in the main text, IPF aims to solve the *matrix balancing problem* (Deming & Stephan, 1940; Schneider & Zenios, 1990):

> 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 $D^1 X^T D^0 \cdot \mathbf{1}_m = q$.

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:

$$d^0_i(k+1) = \frac{p_i}{\sum_j X_{ij} d^1_j(k)}, \quad d^1_j(k+2) = \frac{q_j}{\sum_i X_{ij} d^0_i(k+1)}.$$

We denote by $M^{\text{IPF}}(k) := D^0(k) X D^1(k)$ 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 $M^{\text{IPF}}(k)$ converges; or $M^{\text{IPF}}(k)$ oscillates between accumulation points (Pukelsheim & Simeone, 2009). Furthermore, it is known that there are at most *two* accumulation points, so if IPF does not converge, it oscillates between two solutions, one that matches the target row marginals p and one that matches the target column marginals q (Csiszár & Tusnády, 1984; Aas, 2014). Thus, in our implementation of IPF (Algorithm 1), we check for two stopping conditions, one of which must be met eventually: either IPF converges, such that $M^{\text{IPF}}(k) \approx M^{\text{IPF}}(k+1)$, or IPF exhibits period-2 oscillation, such that $M^{\text{IPF}}(k) \approx M^{\text{IPF}}(k-2)$ and $M^{\text{IPF}}(k-1) \approx M^{\text{IPF}}(k-3)$.

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 marginals p or target column marginals q. However, zero marginals are common in real-world data: for example, in mobility data (Section E.2), many points-of-interest have zero visits at night, and in bikeshare data (Section E.3), some bike stations have zero trips at night. So, our implementation of IPF modifies it slightly to allow for non-negative, instead of strictly positive, marginals. Our version sets $d^0_i = 0$, for all $p_i = 0$, and $d^1_j = 0$, for all $q_j = 0$, then updates all other entries in d^0 and d^1 as usual, as described in (1). We show below 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 our 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

$$d^0_i(k+1) = \frac{p_i}{\sum_j X_{ij} d^1_j(k)} = \frac{p_i}{\sum_{j; q_j > 0} X_{ij} \tilde{d}^1_j(k)} = \tilde{d}^0_i(k+1). \tag{9}$$

In the denominator, 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 network model (4), the number of nonzero entries in $\bar{X}$ 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 substantially reduce our number of observations, since the submatrix drops entire rows and columns. Thus, for a given hour t, our set of observations consists of $\{(i,j) | i \in [m], j \in [n], \bar{X}_{ij} > 0, p^{(t)}_i > 0, q^{(t)}_j > 0\}$, which can be much fewer than mn observations, given high levels of sparsity in real-world $\bar{X}$, $p^{(t)}$, and $q^{(t)}$. Fewer observations result in less accurate and more uncertain parameter estimates, making explicit the connection between data sparsity and quality of IPF estimates, which we also demonstrate empirically in Section E, especially Figure 2.

Algorithm 1 Our implementation of the iterative proportional fitting procedure.

Input: matrix X, row marginals p, column marginals q, tolerance ϵ
Initialize $converged = false$, $oscillate = false$, $\tau = 1$, $d^0 = \mathbf{1}_m$, $d^1 = \mathbf{1}_n$
repeat
  if τ is odd **then**
    for $i = 1$ **to** m **do**
      if $p_i = 0$ **then**
        $d_i^0 \leftarrow 0$
      else
        $d_i^0 \leftarrow \frac{p_i}{\sum_j X_{ij} d_j^1}$
      end if
    end for
  else
    for $j = 1$ **to** n **do**
      if $q_j = 0$ **then**
        $d_j^1 \leftarrow 0$
      else
        $d_j^1 \leftarrow \frac{q_j}{\sum_i X_{ij} d_i^0}$
      end if
    end for
  end if
  $M^{\text{IPF}}(\tau) = \textbf{diag}(d^0) X \textbf{diag}(d^1)$
  if $||M^{\text{IPF}}(\tau) - M^{\text{IPF}}(\tau - 1)||_1 < \epsilon$ **then**
    $converged \leftarrow True$
  else if $||M^{\text{IPF}}(\tau) - M^{\text{IPF}}(\tau - 2)||_1 < \epsilon$ **and** $||M^{\text{IPF}}(\tau - 1) - M^{\text{IPF}}(\tau - 3)||_1 < \epsilon$ **then**
    $oscillate \leftarrow True$
  end if
  $\tau \leftarrow \tau + 1$
until $converged$ is $true$ or $oscillate$ is $true$

B. Deriving Our Generative Network Model

In this section, we provide details on our generative network model. In Appendix B.1, we provide details on the known KL divergence duality result that is key to our following proof. In Appendix B.3, we derive the log-likelihood of our model, prove that IPF recovers the MLEs of our model (Theorem 3.1), and define the equivalent Poisson regression problem. In Appendix B.4, we formalize the "joint" network inference problem, where $\bar{X} = \sum_t X^{(t)}$, and characterize when the joint problem reduces to the decoupled problem solved by our model. In Appendix B.5, we provide necessary and sufficient conditions for IPF to recover the true network exactly (Theorem B.1). Finally, in Appendix B.6, we prove that among a larger class of generalized linear models, the Poisson model is the unique one where the MLE is the IPF solution — so from the perspective of IPF, our generative model is canonical (Theorem B.2). In Figure B.1, we also summarize many of the conditions we discuss in this work (e.g., when IPF converges, when MLEs are finite) and visualize how they fit together.

B.1. Duality result for KL divergence minimization

For completeness, we provide the details for deriving the dual problem (3) of the KL minimization problem (2). Let u and v be the multipliers of the constraints $\hat{Y}\mathbf{1}_n = p, \hat{Y}^T\mathbf{1}_m = q$, respectively. Applying Sion's minimax theorem (Sion, 1958), 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

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

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

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

which we recognize as $g(u, v)$.

B.2. Connections to Entropy Regularized Optimal Transport

In many applications of IPF, the initial matrix X arises from some transportation cost function C of the entropic-regularized optimal transport problem associated with (2). More precisely, with the transformation $X = \exp(-C/\varepsilon)$, the KL minimization problem

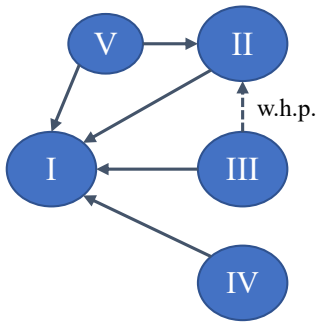
$$\begin{aligned}\min_{\hat{Y} \in \mathbb{R}_+^{n \times m}} D_{\text{KL}}(\hat{Y} \| X) \\ \hat{Y}\mathbf{1}_m = p, \quad \hat{Y}^T \mathbf{1}_n = q\end{aligned}$$

is equivalent to the following regularized optimal transport problem

$$\begin{aligned}\min_{\hat{Y} \in \mathbb{R}_+^{n \times m}} \langle \hat{Y}, C \rangle + \varepsilon D_{\text{KL}}(\hat{Y} \| p \otimes q) \\ \hat{Y}\mathbf{1}_m = p, \quad \hat{Y}^T \mathbf{1}_n = q.\end{aligned}$$

For example, C could be the distance between the origin and the destination or total travel time. In our setting, X is given as $\bar{X}$, the time-aggregated network (e.g., number of bike trips between stations over time or number of visits from neighborhoods to places), but we could still interpret it as a surrogate measure for the above mentioned physical quantities. For example, we can expect that the longer it takes to travel between a particular pair of locations, the fewer people on average will travel between those nodes. Therefore, when we do not have access to information about distance or travel cost, $\bar{X}$ can be used as a reasonable surrogate.

We also prefer to directly use $\bar{X}$ instead of writing it as $\exp(-C/\epsilon)$ since in real-world data, $\bar{X}$ contains many zero elements. If we want to interpret it in terms of C, we need to allow "infinite transportation cost" for some origin-destination pairs. In Appendix E.1.3, we also briefly discuss a possible extension of our biproportional Poisson model based on the gravity model of Navick & Furth (1994), which uses the distance between origin-destination pairs as the cost function C, and with $\bar{X}$ the aggregate number of trips as a constant multiplier instead. When we have access to additional information (besides aggregate trips) that can serve as the cost C, such a model could be preferable.



Conditions

I. IPF converges on inputs $\bar{X}$, $p^{(t)}$, $q^{(t)}$.

II. Matrix balancing problem has a finite solution on $\bar{X}$, $p^{(t)}$, $q^{(t)}$ $\leftrightarrow$ IPF recovers the MLEs of the biproportional Poisson model, given $\bar{X}$, $p^{(t)}$, $q^{(t)}$ (Thm 3.1).

III. $X^{(t)}$ is truly generated from the biproportional Poisson model (Thm 4.2, Cor D.2).

IV. $\bar{X}$ is the true time-aggregated network over snapshots including $X^{(t)}$ (Cor D.1).

V. $X^{(t)} = A\bar{X}B$, for positive diagonal matrices A, B $\leftrightarrow$ IPF recovers $X^{(t)}$ perfectly (Thm B.1).

Figure B.1. Summary of how the different conditions we discuss in this work fit together. Arrows indicate that one condition implies (i.e., is a sufficient condition for) another. Conditions II-V are all sufficient, but not necessary, conditions for IPF to converge. Prior work has also defined several necessary *and* sufficient conditions for IPF to converge (Pukelsheim, 2014); see Section D for details.

B.3. Proof of Theorem 3.1

For convenience, we restate our biproportional Poisson model (4) and Theorem 3.1 below. Our model is

$$X_{ij}^{(t)} \sim \begin{cases} \text{Poisson}(e^{u_i}\bar{X}_{ij}e^{-v_j}), \text{ if } \bar{X}_{ij} > 0, \\ 0, \text{ otherwise.} \end{cases}$$
$$p_i^{(t)} = \sum_j X_{ij}^{(t)}, \quad q_j^{(t)} = \sum_i X_{ij}^{(t)}.$$

where $\bar{X}$ is the time-aggregated network, $X^{(t)}$ is the time-varying network, u and v are scaling factors, and $p^{(t)}$ and $q^{(t)}$ are the row and column sums of $X^{(t)}$, respectively. Our theorem states:

> Assume that the matrix balancing problem on $\bar{X}$, $p^{(t)}$, and $q^{(t)}$ has a finite solution (D^0, D^1). Then d^0 and d^1 are limits of the IPF iterations if and only if $\hat{u} = \log d^0$ and $\hat{v} = -\log d^1$ are solutions to the above maximum likelihood problem given $\bar{X}$, $p^{(t)}$, and $q^{(t)}$, with log-likelihood $\ell = -g(u, v)$ in (3), i.e.,
>
> $$\ell(u, v) = \sum_i p_i^{(t)} u_i - \sum_j q_j^{(t)} v_j - \sum_{ij} \bar{X}_{ij} e^{u_i - v_j}.$$
>
> Moreover, this problem is equivalent to the maximum likelihood estimation of a Poisson regression model, and $p^{(t)}, q^{(t)}$ are the sufficient statistics.

Proof. Since we know that IPF minimizes $g(u, v)$ in (3), our goal is to identify a network model whose likelihood is equivalent to $-g(u, v)$. Several intuitions guide our construction of the model. First, Qu et al. (2023) recently established connections between IPF and choice modeling, and observed that (3) reduces to the maximum likelihood objective of a general class of choice models based on the Luce choice axiom, which suggests the same may be true for a network model. Second, it is well-known that finding maximum likelihood parameters $\hat{\theta}$ is asymptotically equivalent to minimizing the KL divergence to the data-generating distribution, i.e., $\min_\theta D_{\text{KL}}(p_{\text{data}} || p_\theta)$ (Huber, 1967). Since the parameters appear on the right side of this KL problem and X appears in the right side of the KL problem implied by IPF (2), and KL divergence is not symmetric, then our intuition is that the time-aggregated network $\bar{X}$ (which takes the place of X) should appear in the *parameters* of our network model. Lastly, the choice of Poisson distribution is natural given its close connections with KL divergence (Harremoës, 2001; Renner & Warton, 2013), similar to the associations between ℓ_1 and Laplace, as well as ℓ_2 and Gaussian distributions.

With these intuitions, we arrived at our biproportional Poisson model. Now, we derive the log-likelihood of our model to verify that it matches $-g(u, v)$. To simplify notation, we will employ the notation (Y, X, p, q) as shorthand for

$(X^{(t)}, \bar{X}, p^{(t)}, q^{(t)})$ throughout. The model's likelihood is given by

$$\mathcal{L}(Y|X,u,v) = \Pi_{i,j;X_{ij}>0} \frac{(e^{u_i} X_{ij} e^{-v_j})^{Y_{ij}} \exp(-(e^{u_i} X_{ij} e^{-v_j}))}{Y_{ij}!}. \tag{10}$$

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:

$$\max_{u,v} \sum_{i,j;X_{ij}>0} Y_{ij} \cdot (u_i + \log X_{ij} - v_j) - e^{u_i} X_{ij} e^{-v_j}. \tag{11}$$

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

$$\max_{u,v} \sum_{i,j;X_{ij}>0} Y_{ij} u_i - Y_{ij} v_j - e^{u_i} X_{ij} e^{-v_j} \tag{12}$$

$$= \max_{u,v} \sum_i u_i (\sum_{j;X_{ij}>0} Y_{ij}) - \sum_j v_j (\sum_{i;X_{ij}>0} Y_{ij}) - \sum_{i,j;X_{ij}>0} e^{u_i} X_{i,j} e^{-v_j} \tag{13}$$

$$= \max_{u,v} \sum_i u_i p_i - \sum_j v_j q_j - \sum_{ij} e^{u_i} X_{ij} e^{-v_j}, \tag{14}$$

which is equivalent to maximizing $-g(u,v)$. Since IPF minimizes $g(u,v)$ (when the matrix balancing problem has a finite solution), it must also maximize the likelihood of our model, so we have proven the first part of our statement. Furthermore, observe that (14) 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 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

$$y \cdot \theta^T \mathbf{x} - e^{\theta^T \mathbf{x}}. \tag{15}$$

To match this to the log-likelihood of our network model in (11), 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 (11), 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 (Idel, 2016). For details on the maximum flow algorithm, see Section D.1. □

Other settings where IPF recovers MLEs. A popular application of IPF is to estimate contingency tables, where it is known that IPF can be used to derive MLEs for certain log-linear models of contingency tables (Bishop & Fienberg, 1969; Bishop et al., 1974; Little & Wu, 1991; Little, 1993). For example, a related formulation using $\lambda_{ij} = e^{u_i} \bar{X}_{ij} e^{-v_j}$ for contingency tables is studied by Little & Wu (1991). However, instead of Poisson random variables, $X_{ij}^{(t)}$ are treated as probability parameters of a target population. Data from a two-step sampling procedure fitted using IPF then yields the MLE. Recently, Qu et al. (2023) revealed interesting connections of IPF to algorithms in the choice modeling literature, where it is shown that IPF can be used to compute MLEs of a general class of choice models based on Luce's axiom of choice (Luce, 1959). Our network model is closely related to such choice models, and our statistical theory results also draw inspirations from corresponding works in the choice literature (Seshadri et al., 2020; Bong & Rinaldo, 2022).

Comparison to collective graphical models. Sheldon & Dietterich (2011) introduce the problem of collective graphical models (CGMs): how to fit a model of individual behavior from aggregate data (e.g., counts). The general problem that CGM addresses shares strong similarities with the dynamic network inference problem in our work, where we are also trying to estimate finer-grained information (time-varying networks) based on coarser, aggregate data (time-varying marginals and time-aggregated network) due to privacy or data collection costs. A common feature is that the available aggregate data provides the sufficient statistics of the model, which enables valid estimation and inference. An additional algorithmic connection between our work and CGM is that IPF has also been used in the inference of CGM (Singh et al., 2020).

However, a notable difference is that CGM is typically applied in settings where only the time-varying marginals are known, without access to the time-aggregated network. For example, several papers use CGMs to estimate population flows given the number of people in each area over time (Iwata et al., 2017; Akagi et al., 2018; Iwata & Shimizu, 2019). Furthermore, CGM seeks to relate population-level observations to a model of individual behavior (e.g., a single person's mobility or a bird's migration patterns), while we are not trying to estimate individual models. Instead of disaggregating over populations, we seek to disaggregate over time, by taking the time-aggregated network and distributing it over smaller time intervals (e.g., from monthly to hourly). Finally, the specific mathematical formulations of the problems in CGM and in the network inference problem seem to be distinct. CGM is built on probabilistic graphical models of independent and identically distributed individuals. It aims to estimate the parameters of random vectors on nodes of a graph given only aggregate counts of specific states using MAP inference. In contrast, our biproportional Poisson model is used to model the random population flows between nodes of a graph using maximum likelihood estimation, which is solved via IPF.

B.4. The "joint" network inference problem

In our biproportional Poisson model (4), we used the time-aggregated network $\bar{X}$ as the baseline intensity in the Poisson parameter $e^{u_i}\bar{X}_{ij}e^{-v_j}$. We then solve many "decoupled" problems, where we fit the model separately for every time step t. However, in doing so we are potentially leaving out additional information implied by the constraint that the Poisson variables aggregated over time should be *equal* to the observed time aggregated traffic $\bar{X}$. This constraint is only applicable when we are inferring $X^{(t)}$ for the same set of time steps t over which $\bar{X}$ is aggregated, so it would not apply, for example, if $\bar{X}$ is aggregated from historical data or we are only inferring $X^{(t)}$ for some subset of time steps. In cases where it does apply, we consider instead the following "joint" model

$$Y_{ijt} = \begin{cases} \text{Poisson}(X_{ij}e^{-v_{it}+u_{jt}}) & X_{ij} > 0 \\ 0 & X_{ij} = 0 \end{cases}, \tag{16}$$

where Y_{ijt} describes the traffic between nodes i, j at time t, X_{ij} is an *unknown* parameter that describes the time-invariant propensity of traffic between i, j, and v_{it}, u_{jt} are now time-dependent intensity parameters that describe the level of activity at different nodes.

In the cross-sectional problem, we use the observed $\bar{X}_{ij}$ which aggregates over different times in place of X_{ij}. In (16), instead of having X_{ij} as an observable, we posit that it is a model parameter that needs to be estimated, and we observe the three marginals quantities $\bar{X}, P, Q$:

$$\bar{X}_{ij} = \sum_t Y_{ijt},$$
$$Q_{it} = \sum_j Y_{ijt},$$
$$P_{jt} = \sum_i Y_{ijt}.$$

Importantly, we no longer require that X_{ij} and $\bar{X}_{ij}$ have the same zero patterns, as $\bar{X}_{ij} = 0$ could result from $\text{Poisson}(X_{ij}e^{-v_{it}+u_{jt}}) = 0$ for all t. The relevant part of the log-likelihood function of the joint model (16) is given by

$$\sum_{ijt} Y_{ijt}\log(X_{ij}e^{-v_{it}+u_{jt}}) - \sum_{ijt} X_{ij}e^{-v_{it}+u_{jt}} =$$
$$\sum_{ij}\bar{X}_{ij}\log X_{ij} - \sum_{it} Q_{it}v_{it} + \sum_{jt} P_{jt}u_{jt} - \sum_{ijt} X_{ij}e^{-v_{it}+u_{jt}},$$

where again the marginal quantities $\bar{X}, P, Q$ are the sufficient statistics of the model. The first order conditions are given by

$$e^{-v_{it}} \cdot \sum_j X_{ij} e^{u_{jt}} = Q_{it}$$
$$e^{u_{jt}} \cdot \sum_i X_{ij} e^{-v_{it}} = P_{jt}$$
$$X_{ij} \cdot \sum_t e^{-v_{it}+u_{jt}} = \bar{X}_{ij}.$$

Compare this system with that of the cross-sectional problem, where for each fixed t, we solve

$$e^{-v_{it}} \cdot \sum_j \bar{X}_{ij} e^{u_{jt}} = Q_{it}$$
$$e^{u_{jt}} \cdot \sum_i \bar{X}_{ij} e^{-v_{it}} = P_{jt}.$$

In contrast, for the joint problem we instead solve

$$e^{-v_{it}} \cdot \sum_j X_{ij} e^{u_{jt}} = Q_{it}$$
$$e^{u_{jt}} \cdot \sum_i X_{ij} e^{-v_{it}} = P_{jt}$$

where X_{ij} satisfies the additional constraint

$$X_{ij} \cdot \sum_t e^{-v_{it}+u_{jt}} = \bar{X}_{ij}.$$

Therefore, the implicit assumption used to reduce the joint problem to the cross-sectional problems is that

$$\sum_t e^{-v_{it}+u_{jt}} \approx c \tag{17}$$

for all i, j where $\bar{X}_{ij} > 0$; otherwise, $X_{ij} = 0$ or $\sum_t e^{-v_{it}+u_{jt}} = 0$. In other words, the aggregated time-dependent intensities for each pair (i, j) does not depend on i, j. We may justify (17) in several ways. For example, if we assume v_{it} and u_{jt} are drawn i.i.d. from two distributions with the same mean, then $\frac{1}{T}\sum_t e^{-v_{it}+u_{jt}} \to 1$ by the continuous mapping theorem. Alternatively, if X_{ij} is an accurate description of the long-run traffic propensity between i, j, then the intensity of the aggregate traffic Poisson variables $\sum_t Y_{ijt}$ should largely be captured by X_{ij}, i.e., *independent* of $\sum_t e^{-v_{it}+u_{jt}}$.

B.5. When IPF recovers the true network exactly

If we swap out the Poisson distribution in our biproportional Poisson model (4) for an identity function, we find that this condition is necessary and sufficient for IPF to exactly recover the true network, $X^{(t)}$, given $\bar{X}$, $p^{(t)}$, $q^{(t)}$.

Theorem B.1. *IPF returns $X^{(t)}$ if and only if $X^{(t)} = A\bar{X}B$, for some positive diagonal matrices A and B.*

Proof. First, we prove that, if IPF returns $X^{(t)}$, then $X^{(t)} = A\bar{X}B$. All IPF solutions take the form $D^0\bar{X}D^1$, where D^0 and D^1 are positive diagonal matrices. If IPF returns $X^{(t)}$, then $X^{(t)}$ can be written as $A\bar{X}B$, with $A = D^0$ and $B = D^1$.

Second, we prove that, if $X^{(t)} = A\bar{X}B$, then IPF will return $X^{(t)}$. First, if there is a finite solution to the matrix balancing problem, IPF will converge to a solution (Pukelsheim, 2014). We know that $A\bar{X}B$ is a solution, so IPF will return $D^0\bar{X}D^1$, where $(D^0\bar{X}D^1) \cdot \mathbf{1}_n = p^{(t)}$ and $(D^1\bar{X}^TD^0) \cdot \mathbf{1}_m = q^{(t)}$. Furthermore, 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 $M^1 = M^2$ (Pukelsheim, 2014). Thus, IPF will return $X^{(t)}$, since IPF will converge to $D^0\bar{X}D^1$, with marginals $p^{(t)}$ and $q^{(t)}$, and $X^{(t)}$ is the unique biproportional scaling of $\bar{X}$ that matches the marginals. □

While this statement about $X^{(t)}$, that $X^{(t)} = A\bar{X}B$, is rarely true in practice, this result allows us to pin down when IPF works "perfectly" for dynamic network inference. Notably, this result implies that the scaling matrix from $\bar{X}$ to $X^{(t)}$, i.e., $X^{(t)}_{ij}/\bar{X}_{ij}$ where $\bar{X}_{ij} > 0$, must be a rank-1 matrix with entries $a_i b_j$ (the diagonals of A and B). If we interpret $X^{(t)}$ as a dynamic network and $\bar{X}$ as its time-aggregated form, then we are essentially constraining the complexity of the network's temporal variation.

B.6. Poisson links uniquely recover IPF within a class of generalized linear models

In Theorem 3.1, we showed that IPF recovers the MLEs of the biproportional Poisson model, and then we argued that spelling out this model clarifies implicit assumptions when using IPF to infer dynamic networks from their marginals. However, this argument requires some degree of *uniqueness* in the model we identified, since if IPF recovers the MLEs of many different models, then using IPF to infer dynamic networks may be justified under the assumptions of many different models, not only the biproportional Poisson. To analyze uniqueness, in this section we consider a very natural family of generalized linear models which generalizes the biproportional Poisson model. We prove a rigorous result which shows that if we want to recover IPF as the solution to maximum likelihood estimation, then we are *forced* to choose scaled Poisson distributions for the link[3]. This shows that in some sense our model is canonical, and it is unique within this general family.

Throughout this section, time superscripts are omitted.

Class of models where row and column statistics are sufficient. Suppose that $(p_\alpha)_{\alpha \in A}$ is an exponential family with canonical parameter[4] α and sufficient statistic x. Assuming without loss of generality that $0 \in A$, this means that

$$\frac{dp_\alpha}{dp_0}(x) = \exp(\alpha x - \Phi(\alpha)). \tag{18}$$

We suppose the observations are generated by the following generalized linear model: independently for each i and j we sample

$$Y_{ij} \sim p_{\alpha_{ij}}, \qquad \alpha_{ij} = u_i + \log X_{ij} - v_j. \tag{19}$$

Note that the algebraic form of (18) and the weights α_{ij} in (19) is exactly what leads in the derivation of (14) to the row and column sums being sufficient statistics for the overall model. This is also illustrated below in the computation of the log-likelihood. In other words, the key desiderata that the row and column statistics are sufficient does not *by itself* force us to consider the biproportional Poisson model — rather, it is the fundamental property of this larger class of GLMs. Many important distribution families satisfy (18): for example, the class of unit-variance Gaussians $N(\mu, 1)$ or the class of exponential distributions $\mathrm{Exp}(\lambda)$.

Thus, this is the natural class of models to consider if we want to see if IPF is also the MLE for a variant of our model, or in other words test how unique the connection between IPF and the biproportional Poisson model is. As we will show, this connection is unique — the only measures p_α that will recover IPF are scalings of the Poisson family, and this is true even if we only require the measure to satisfy a significantly generalized version of the IPF equations.

Log-likelihood under general model. Recall that we use (y, X, p, q) as shorthand for $(X^{(t)}, \bar{X}, p^{(t)}, q^{(t)})$. The log-likelihood of our model, assuming X is known, is

$$\begin{aligned} L(u, v; y) &= \sum_{ij} \log p_{\alpha_{ij}}(y_{ij}) = \sum_{ij} (u_i + \log X_{ij} - v_j) y_{ij} - \sum_{ij} \Phi(u_i + \log X_{ij} - v_j) \\ &= \sum_i u_i p_i - \sum_j v_j q_j - \sum_{ij} \Phi(u_i + \log X_{ij} - v_j) + C(y). \end{aligned}$$

where $C(y) = \sum_{ij} y_{ij} \log X_{ij}$ does not depend on the model parameters u or v. As promised, we see that the row sums $p_i = \sum_j y_{ij}$ and column sums $q_j = \sum_i y_{ij}$ are sufficient statistics for this model.

[3]Note that if the inputs to the IPF algorithm are uniformly scaled, the outputs of IPF are scaled in the same way, so the fact that scaling doesn't matter is expected. This is a useful property of IPF, because it means it works even when data is scaled/normalized.

[4]See e.g. (McCullagh, 2019) for a reference on exponential families and generalized linear models. As an example, for a Poisson distribution with rate λ the canonical parameter is $\log \lambda$.

Differentiating, we see that

$$\partial_{u_i} L(u, v; y) = p_i - \sum_j \Phi'(u_i + \log X_{ij} - v_j) = p_i - \sum_j \Phi'(u_i - w_{ij}), \tag{20}$$

where for convenience we defined

$$w_{ij} = \log X_{ij} - v_j$$

to be the term in the linear model added to u_i. The partial derivative with respect to v_j satisfies a symmetrical equation. Because of this symmetry, it will suffice for us to consider the equation for the row weights in what follows.

(Generalized) IPF is uniquely recovered by a scaled Poisson link. We say that the MLE satisfies a *generalized proportional fitting equation* if

$$\partial_{u_i} L(u, v; y) = 0 \iff h(u_i) = \frac{p_i}{\sum_j f(w_{ij})} \tag{21}$$

for some smooth functions f, h. This definition captures and generalizes the fact that in IPF, the row weight, parameterized here by u_i, is determined by the ratio of the empirical count p_i and the corresponding row sum. In the special case of the Poisson distribution, we have $f = h = \exp$.

Now we show that if the critical point equation has the generalized form (21), the distribution corresponding to the link must be a scaled Poisson family — in which case, we know that we get exactly the IPF equations discussed in the main body of the paper. So for this family of generalized linear models, we recover the IPF equations as a characterization of the MLE exactly when the link corresponds to a scaled Poisson distribution.

Theorem B.2. *In the above generalized linear model, the MLE satisfies a generalized proportional fitting equation for all parameters u, v iff the exponential family $\{p_\alpha\}_\alpha$ is the set of Poisson distributions scaled by some factor $a \in \mathbb{R}$. In other words, $\{p_\alpha : \Phi(\alpha) < \infty\} = \{a \operatorname{Poisson}(\lambda) : \lambda \geq 0\}$ for some a.*

Proof. Substituting (20), we see that (21) is equivalent to asking that

$$h(u_i) \sum_j f(w_{ij}) = \sum_j \Phi'(u_i + w_{ij}). \tag{22}$$

Taking the partial derivative with respect to v_j (equivalently, w_{ij}), this implies that

$$h(u_i) f'(w_{ij}) = \Phi''(u_i + w_j).$$

This holding for all possible u, v is the same as requiring that

$$\Phi''(u + t) = h(u) \cdot f'(t).$$

Letting $t = 0$ gives that $h(u) = \Phi''(u)/f'(0)$ and letting $u = 0$ gives that $f'(t) = \Phi''(t)/h(0) = f'(0)\Phi''(t)/\Phi''(0)$. So

$$\Phi''(u + t) = \frac{\Phi''(u)\Phi''(t)}{\Phi''(0)}.$$

Taking the partial derivative with respect to t on both sides and setting $t = 0$, we have

$$\Phi'''(u) = \Phi''(u) \frac{\Phi'''(0)}{\Phi''(0)}.$$

Equivalently, letting $a = \frac{\Phi'''(0)}{\Phi''(0)}$, we have

$$\frac{d}{du} \log \Phi''(u) = \frac{\Phi'''(u)}{\Phi''(u)} = a$$

Integrating yields

$$\log \Phi''(u) = au + b$$

so

$$\Phi''(u) = e^{au+b}.$$

Integrating twice more yields

$$\Phi(u) = e^{au+b} + cu + d.$$

From the definition, we must have $\Phi(0) = 0$ so

$$\Phi(u) = e^{au+b} + cu - e^b. \tag{23}$$

Recalling that $f'(t) = f'(0)\Phi''(t)/\Phi''(0)$ and $h(u) = \Phi''(u)/f'(0)$, returning to (22) yields that

$$\frac{\Phi''(u_i)}{\Phi''(0)} \sum_j (\Phi'(w_{ij}) + f(0)) = \sum_j \Phi'(u_i + w_{ij}),$$

and plugging in (23) yields

$$ae^{au_i} \sum_j (e^{aw_{ij}+b} + f(0)) = \sum_j (ae^{a(u_i+w_{ij})+b} + c),$$

so

$$c = ae^{au_i} f(0)$$

for all u_i. Either $a = 0$, in which case this implies $c = 0$, or if $a \neq 0$ this implies $f(0) = 0$ and so $c = 0$ regardless. Hence

$$\Phi(u) = e^{au+b} - e^b.$$

Recall that the cumulant generating function of the Poisson distribution with canonical parameter b (i.e. with rate $\lambda = e^b$) is $u \mapsto e^b(e^u - 1)$. So the above Φ is exactly the cumulant generating function of a scaled Poisson distribution $a\,\mathrm{Poisson}(e^b)$. We know that such distributions satisfy the IPF equations, so this is indeed an if-and-only-if statement. □

C. Statistical theory of the biproportional Poisson model

In this section, we provide a comprehensive presentation including proofs of our statistical results on the biproportional Poisson model. In Appendix C.1, we derive bounds on the estimation error of the model's MLEs (when bounded). In Appendix C.2, we prove that, under the correct specification of the biproportional Poisson model, the maximum likelihood estimation problem has a unique bounded normalized solution with high probability.

We begin with a precise definition of the graph Laplacian used in this work. Given the time-aggregated matrix $\bar{X}$, we define the bipartite graph G_b with adjacency matrix $A(\bar{X})$:

$$A(\bar{X}) := \begin{bmatrix} 0 & \bar{X} \\ \bar{X}^T & 0 \end{bmatrix}, \mathcal{L} := \begin{bmatrix} \mathcal{D}(\bar{X}\mathbf{1}_m) & -\bar{X} \\ -\bar{X}^T & \mathcal{D}(\bar{X}^T\mathbf{1}_n) \end{bmatrix}. \tag{24}$$

We can verify that $\mathcal{L}$ is also the Hessian of the negative log-likelihood of the biproportional Poisson model evaluated at $(u, v) = (0, 0)$.

C.1. Structure-dependent Bounds on Estimation Error of MLE

We begin with a lemma from Qu et al. (2023), which bounds the Hessian of the negative log-likelihood (3) in a bounded domain by the Fiedler eigenvalue. Recall that we use X as shorthand for $\bar{X}$.

Lemma C.1 (Appendix F.3 of (Qu et al., 2023))**.** *For all $(u, v) \in 1_{m+n}^{\perp}$ with $\|(u, v)\|_\infty \leq B$,*

$$\lambda_{-2}(\nabla^2 g(u, v)) \geq e^{-2B} \lambda_{-2}(\mathcal{L})$$

where

$$\mathcal{L} = \mathcal{D}\left(\begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix} \begin{bmatrix} 1_n \\ 1_m \end{bmatrix}\right) - \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix}$$

is the graph Laplacian of the weighted bipartite graph given by X.

Using the lemma above, we can obtain bounds on the expected squared error for the normalized MLEs quantified by the Fiedler eigenvalue, as long as the true parameters and MLEs are all bounded by some constant B.

Theorem C.2. *Suppose that the Poisson network model holds with ground truth parameters u^*, v^*, X. Suppose (u, v) is a maximizer of the likelihood (minimizer of g) and that we have the normalization condition $(u - u^*, v - v^*) \in 1_{m+n}^{\perp}$ and $\|(u, v, u^*, v^*)\|_\infty \leq B$. Then over the randomness of $p^{(t)}, q^{(t)}$ we have in expectation*

$$\mathbb{E}\|(u - u^*, v - v^*)\|^2 1_{\mathcal{B}} \leq \frac{8e^{4B}}{\lambda_{-2}(\mathcal{L})^2}\kappa \tag{25}$$

where $\mathcal{B}$ is the event that the MLE exists and is bounded by B, and

$$\kappa = \sum_{i,j} e^{u_i} X_{ij} e^{-v_j}.$$

Furthermore, with probability at least $1 - \delta$, we have that whenever the MLE exists and is bounded by B, it satisfies

$$\|(u - u^*, v - v^*)\|^2 \leq \frac{8e^{4B}}{\lambda_{-2}(\mathcal{L})^2}\left(\lambda + \sqrt{6\log(4/\delta)\lambda(1+M)} + C\log^2(12/\delta)\log^2(1+m+n)(1+M)\right)$$

where $C > 0$ is an absolute constant, and

$$M := \max\{\max_{i\in[m]} \sum_j e^{u_i^*} X_{ij} e^{-v_j^*}, \max_{j\in[n]} \sum_i e^{u_i^*} X_{ij} e^{-v_j^*}\}.$$

is the maximum of the row and column sums of the matrix $(e^{u_i} X_{ij} e^{-v_j})_{ij}$.

Remark C.3 (Interpretation of Theorem C.2). We illustrate the bound with some examples, discuss the meaning and tightness of this bound, and give a simple sufficient condition for the MLE to be bounded.

1. *Examples with strong recovery guarantees: complete network.* Suppose that X is the $n \times m$ all-ones matrix and that B is a constant (e.g. $B = 5$). Then $\kappa = \Theta(nm)$ and $\lambda_{-2}(\mathcal{L}) = \min(n, m)$ so the right hand side of (25) is of order $\Theta(\max(n/m, m/n))$. So if e.g. $n = m$, then the total error for recovering the entire vector is $O(1)$. Equivalently, the average error per coordinate of u, v is $\frac{1}{2n}\|(u - u^*, v - v^*)\|^2 = O(1/n)$.

   For a more sophisticated example on a random graph model, see Appendix E.1.2.

2. *Meaning of normalization condition.* The normalization condition $\sum_i u_i + \sum_i v_i = \sum_i u_i^* + \sum_i v_i^*$ eliminates a scaling ambiguity in the model: if we add a constant c to u^* and to v^*, it does not change the distribution of the output of the model.

3. *Dependence on κ is optimal.* Consider the special case where $X : m \times 1$ with all-ones entries, and we fix the scaling convention that $v_1^* = 0$ (because any value of v_1^* is consistent with some shift of u^*). Then the generative model reduces to observing m independent Poisson variables with rates $e^{u_i^*}$ for $i = 1$ to m, and our estimator u reduces to solving the equation $e^{u_i} = p_i^{(t)}$. Then $\kappa = \sum_{ij} e^{u_i}$ is the total variance of estimating the quantities $e^{u_i^*}$. Since $\lambda_{-2} = 1$ in this example, we see that the bound is optimal up to constants and the dependence on B in this example.

4. *Dependence on λ_{-2} is required; non-identifiability when $\lambda_{-2} = 0$.* If $\lambda_{-2} = 0$ no guarantee of this form is possible for the estimator, because the parameters of the model are not identifiable. The reason is that this corresponds to the case where the graph is disconnected, in which case the true parameters are not identifiable. To illustrate the reason why, consider the case where $X = I$. Then for every $i \in [n]$, we can replace u_i^*, v_i^* with $u_i^* + c, v^* + c$ for any $c \in \mathbb{R}$ without changing the distribution of observations from the model. Unless $n = 1$, this results in nonidentifiability in the model which is not fixed by the normalization condition. More generally, whenever the graph is disconnected, we can similarly shift the parameters corresponding to one of the components without changing the distribution of observations.

   We briefly remark that some quantitative dependence on λ_{-2} is required even if we restrict to the case $\lambda_{-2} > 0$. This is because the above argument can be modified to show that the parameters are statistically indistinguishable if the graph is not strictly disconnected but effectively so (e.g. if we add $\epsilon > 0$ to all of the entries of X for a very tiny ϵ). This can be formally proven by a similar KL calculation to the one in the example below.

5. *Example where $e^{\Theta(B)}$ term is required.* We show that the term with exponential dependence on B in (25) cannot be removed — this is by establishing a lower bound which holds not just for our estimator, but for any possible estimator. Consider a very special case of the model, where we have a single observation from a Poisson distribution with parameter $e^{u_1^*}$. Consider two such models: in the first one $u_1^* = u_1' = -B/2$ and in the second one $u_1^* = u_1'' = -B$. The KL divergence between the two Poisson distributions $Poi(e^{u_1'})$ and $Poi(e^{u_1''})$ is $e^{u_1'}(u_1' - u_1'') + e^{u_1'} - e^{u_2'} = O(Be^{-B/2}) = O(e^{-B/4})$. By Pinsker's inequality and the Neyman-Pearson Lemma (see e.g. (Rigollet & Hütter, 2023)), this means that it is impossible to distinguish between these two distributions from a single sample with probability of success better than $1/2 + e^{-\Omega(B)}$. This means any estimator for the true parameter will have to make an error of size $\Omega(B)$ with probability at least $1/2 - e^{-\Omega(B)}$ in one of these models. Because $\kappa = e^{-\Omega(B)}$ and $\lambda_{-2}(\mathcal{L}) = 1$, the right hand side of (25) would be exponentially small in B without the presence of the term e^{4B}, which would contradict our lower bound.

6. *Generalization of conclusion.* From the proof of the theorem, we can see that the final high probability guarantee doesn't hold specifically for u, v the MLE, but for any parameters u, v which achieve at least as high log-likelihood than the ground truth u^*, v^* on the training data. (So in this form, the theorem could be applied even if the MLE doesn't exist.)

7. *MLE is bounded with high probability if entries of X large enough.* Consider scaling X by $\beta > 1$. When β is large enough, one can actually show that with high probability the MLE must be close to (u^*, v^*) — in particular it exists and it is bounded. From the proof of Theorem 4.1 and Markov, with 99% probability we have

$$g(u,v) \geq g(u^*,v^*) - \sqrt{200 \sum_{ij} e^{u_i^*} X_{ij} e^{-v_j^*}} + e^{-2b}\lambda_{-2}(\mathcal{L})(\|u-u^*\|^2 + \|v-v^*\|^2)/2.$$

Suppose that $X = \beta Z$ and let $\mathcal{L}_Z$ be the Laplacian for Z. Then this is

$$g(u,v) \geq g(u^*,v^*) - \sqrt{\beta}\sqrt{200 \sum_{ij} e^{u_i^*} X_{ij} e^{-v_j^*}}\|(u,v)-(u^*,v^*)\| + e^{-2b}\beta\lambda_{-2}(\mathcal{L}_Z)(\|(u,v)-(u^*,v^*)\|^2)/2.$$

Define $B = \|(u^*,v^*)\|_\infty$ and observe $b \leq B + \|(u-u^*, v-v^*)\|$ so letting $r = \|(u-u^*, v-v^*)\|$ we have the inequality

$$g(u,v) \geq g(u^*,v^*) - \sqrt{\beta}\sqrt{200 \sum_{ij} e^{u_i^*} Z_{ij} e^{-v_j^*}}\, r + e^{-2B-2r}\beta\lambda_{-2}(\mathcal{L}_Z) r^2/2.$$

This implies $g(u,v) > g(u^*,v^*)$ if

$$\sqrt{\beta} e^{-2B-2r} r \lambda_{-2}(\mathcal{L}_Z) > 40\sqrt{\kappa}$$

where $\kappa = \sum_{ij} e^{u_i^*} X_{ij} e^{-v_j^*}$. In particular, if $r = 1$ this reduces to the condition

$$\sqrt{\beta} e^{-2B} \lambda_{-2}(\mathcal{L}_Z) > 40 e^2 \sqrt{\kappa}$$

so taking

$$\beta > 400^2 \frac{e^{4B}}{\lambda_{-2}(\mathcal{L}_Z)^2}\kappa$$

suffices. Finally, observe by convexity that this implies the minimum of g is attained within a ball of radius 1 around (u^*, v^*), since $g(u,v) > g(u^*,v^*)$ for all u, v on the sphere of radius 1 centered at (u^*, v^*).

Proof of Theorem C.2. In this proof, we omit the time superscripts so $p = p^{(t)}$ and $q = q^{(t)}$.

We have by Lemma C.1, Taylor's theorem, and the Cauchy-Schwarz inequality that

$$\begin{aligned} g(u,v) &\geq g(u^*,v^*) + \langle \nabla g(u^*,v^*), (u-u^*, v-v^*)\rangle + e^{-2B}\lambda_{-2}(\mathcal{L})(\|u-u^*\|^2 + \|v-v^*\|^2)/2 \\ &\geq g(u^*,v^*) - \|\nabla g(u^*,v^*)\|\|(u-u^*, v-v^*)\| + e^{-2B}\lambda_{-2}(\mathcal{L})(\|u-u^*\|^2 + \|v-v^*\|^2)/2. \end{aligned}$$

Therefore, using that $g(u^*,v^*) \geq g(u,v)$ and rearranging, we have that

$$\|\nabla g(u^*,v^*)\|\|(u-u^*, v-v^*)\| \geq e^{-2B}\lambda_{-2}(\mathcal{L})(\|u-u^*\|^2 + \|v-v^*\|^2)/2.$$

Dividing through by $\|(u-u^*, v-v^*)\|$ gives

$$\|\nabla g(u^*, v^*)\| \geq e^{-2B}\lambda_{-2}(\mathcal{L})\|(u-u^*, v-v^*)\|/2. \tag{26}$$

It remains to control $\|\nabla g(u^*, v^*)\|$. First observe that

$$\partial_{u_i} g = \sum_j X_{ij} e^{-v_j+u_i} - p_i = \lambda_i - p_i$$

and $p_i \sim \text{Poisson}(\lambda_i)$ if we define $\lambda_i = \sum_j e^{u_i} X_{ij} e^{-v_j}$, so $\mathbb{E}\nabla_u g(u^*, v^*) = 0$. Next, note that

$$\|\nabla_u g(u^*, v^*)\|^2 = \sum_i (\partial_{u_i} g(u^*, v^*))^2 = \sum_i (\lambda_i - p_i)^2 .$$

Since $p_i \sim \text{Poisson}(\lambda_i)$, we have that

$$\mathbb{E}\,(\lambda_i - p_i)^2 = \lambda_i, \qquad \mathbb{E}\,(\lambda_i - p_i)^4 = 3\lambda_i^2 + \lambda_i$$

so $\text{Var}\left((\lambda_i - p_i)^2\right) = 2\lambda_i^2 + \lambda_i$. Therefore,

$$\mathbb{E}\|\nabla_u g(u^*, v^*)\|^2 = \sum_i \lambda_i = \sum_{ij} e^{u_i} X_{ij} e^{-v_j}, \qquad \text{Var}\left(\|\nabla_u g(u^*, v^*)\|^2\right) = \sum_j (2\lambda_i^2 + \lambda_i).$$

Applying Lemma C.10 and Lemma C.9 yields that

$$\left\|\max_i(|\lambda_i - p_i|^2 - \lambda_i)\right\|_{\psi_{1/2}} \leq K \log^2(1+m) \max_i(1+\lambda_i)$$

where K is an absolute constant. Hence, applying Theorem C.6 yields that with probability at least $1-\delta/2$,

$$\begin{aligned}\|\nabla_u g(u^*, v^*)\|^2 - \sum_i \lambda_i &\leq \sqrt{3\log(4/\delta)\sum_i (2\lambda_i^2 + \lambda_i)} + K' \log^2(12/\delta)\log^2(1+m)\max_i(1+\lambda_i) \\ &\leq \sqrt{6\log(4/\delta)\sum_i \lambda_i \max_i(1+\lambda_i)} + K' \log^2(12/\delta)\log^2(1+m)\max_i(1+\lambda_i).\end{aligned}$$

By a completely symmetrical argument, an analogous bound holds for $\|\nabla_v g(u^*, v^*)\|^2$. Summing the two bounds and using the union bound yields that with total probability at least $1-\delta$,

$$\|\nabla g(u,v)\|^2 \leq 2\sum_{ij} e^{u_i} X_{ij} e^{-v_j} + \sqrt{24\log(4/\delta)\sum_{ij} e^{u_i} X_{ij} e^{-v_j}(1+M)} + 2K' \log^2(12/\delta)\log^2(1+m+n)(1+M)$$

where we recall that

$$M = \max\left\{\sum_j e^{u_i} X_{ij} e^{-v_j} \mid i \in [m]\right\} \cup \left\{\sum_i e^{u_i} X_{ij} e^{-v_j} \mid j \in [n]\right\}$$

is the maximum of the row and column sums of the matrix $(e^{u_i} X_{ij} e^{-v_j})_{ij}$. Finally, recalling (26) we see that

$$\begin{aligned}&\|(u-u^*, v-v^*)\|^2 \\ &\leq 4\frac{e^{4B}}{\lambda_{-2}(\mathcal{L})^2}\|\nabla g(u^*, v^*)\|^2 \\ &\leq \frac{8e^{4B}}{\lambda_{-2}(\mathcal{L})^2}\left(\sum_{ij} e^{u_i} X_{ij} e^{-v_j} + \sqrt{6\log(4/\delta)\sum_{ij} e^{u_i} X_{ij} e^{-v_j}(1+M)} + K' \log^2(12/\delta)\log^2(1+m+n)(1+M)\right).\end{aligned}$$

A simpler version of this argument proves the in-expectation bound (simply combine (26) with the calculation for $\mathbb{E}\|\nabla g(u^*, v^*)\|^2$). □

C.1.1. Concentration inequalities with Orlicz norms

We make use of powerful concentration of measure estimates from the literature in terms of Orlicz norms. These are useful because we need to give concentration estimates for sums of squares of Poisson random variables, and a standard "Chernoff bound" argument cannot be applied to these because their tails are too heavy for moment generating functions to exist.

Definition C.4 (Orlicz norm)**.** Given a function $\psi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, we define the corresponding Orlicz norm of a random variable X by

$$\|X\|_\psi = \inf\{\lambda > 0 : \mathbb{E}\psi(|X|/\lambda) \leq 1\}.$$

The Orlicz norm may not be a norm in the usual sense (in particular, satisfy the triangle inequality) if ψ is not convex, but this terminology is commonly used anyway.

Definition C.5 (α-exponential Orlicz norm)**.** For $\alpha > 0$, we define

$$\psi_\alpha(x) = \exp(x^\alpha) - 1 \tag{27}$$

and refer to $\|\cdot\|_{\psi_\alpha}$ as the α-exponential Orlicz norm.

The following result arises by specializing the upper tail bound of Theorem 4 of (Adamczak, 2008) to the case of the identity function class with $\delta = 1$ and $\eta = 1/2$.

Theorem C.6 (Special case of Theorem 4 from (Adamczak, 2008))**.** *Let $\alpha > 0$. There exists a constant $C_\alpha > 0$ such that the following is true. Suppose that $X_1, \ldots, X_n$ are independent, mean-zero random variables with finite α-exponential Orlicz norm. Then*

$$\Pr\left(\left|\sum_{i=1}^n X_i\right| \geq t\right) \leq \exp\left(-\frac{t^2}{3\sigma^2}\right) + 3\exp\left(-\left(\frac{t}{C_\alpha \|\max_{1\leq i\leq n} |X_i|\|_{\psi_\alpha}}\right)^\alpha\right)$$

where

$$\sigma^2 = \sum_{i=1}^n \mathbb{E}X_i^2$$

Remark C.7. The conclusion of the above theorem directly implies the following. With probability at least $1 - \delta$,

$$\left|\sum_{i=1}^n X_i\right| \leq \sigma\sqrt{3\log(2/\delta)} + C_\alpha \|\max_{1\leq i\leq n} |X_i|\|_{\psi_\alpha} \sqrt[\alpha]{\log(6/\delta)}.$$

While ψ_α may not be a norm it has similar properties:

Lemma C.8 (Property 4.ii and 4.vi of (Chamakh et al., 2020))**.** *For $c \in \mathbb{R}$,*

$$\|c\|_{\psi_\alpha} = |c|/\sqrt[\alpha]{\log 2}.$$

For any random variables X, Y and $\alpha > 0$

$$\|X + Y\|_{\psi_\alpha} \leq 2(\|X\|_{\psi_\alpha} + \|Y\|_{\psi_\alpha}).$$

Lemma C.9 (Property 4.viii of (Chamakh et al., 2020))**.** *For any $\alpha > 0$ and random variables $X_1, \ldots, X_n$,*

$$\left\|\max_{1\leq i\leq n} |X_i|\right\|_{\psi_\alpha} \leq \left[\frac{\log(1+n)}{\log(3/2)}\right]^{1/\alpha} \max_{1\leq i\leq n} \|X_i\|_{\psi_\alpha}.$$

We also use the following standard tail bound for Poisson random variables (it can be derived from Bernstein's inequality via the Poisson CLT, see (Vershynin, 2018)).

Lemma C.10 (Standard, see e.g. (Vershynin, 2018))**.** *For any $\lambda \geq 0$, if $X \sim$ Poisson(λ) then*

$$\Pr(|X - \lambda| \geq t) \leq 2\exp\left(\frac{-t^2}{2(\lambda + t/3)}\right).$$

The tail bound implies an upper bound on the subexponential norm of Poisson variables, which can be reinterpreted as a bound on $1/2$-exponential Orlicz norm of their square.

Lemma C.11. *There exists an absolute constant $C > 0$ such that the following holds. Suppose that $X \sim$ Poisson(λ) for some $\lambda \geq 0$. Then*

$$\|(X-\lambda)^2\|_{\psi_{1/2}} = \|X-\lambda\|_{\psi_1} \leq C(1+\lambda).$$

Proof. Recall from Lemma C.10, we have that

$$\Pr(|X-\lambda| \geq t) \leq 2\exp\left(\frac{-t}{2(\lambda/t+1/3)}\right).$$

For any $t \geq 1$, this implies that $\Pr(|X-\lambda| \geq t) \leq 2\exp\left(\frac{-t}{2(\lambda+1/3)}\right) < 2\exp\left(\frac{-t}{6(\lambda+1/3)}\right)$ and so in fact $\Pr(|X-\lambda| \geq t) \leq 2\exp\left(\frac{-t}{6(\lambda+1/3)}\right)$ for all $t \geq 0$, because the right hand side is larger than 1 for $t < 1$. Therefore the conclusion follows from an equivalent characterization of subexponential random variables, more preccisely Proposition 2.7.1 of (Vershynin, 2018). □

C.2. Well-posedness of maximum likelihood estimation of biproportional Poisson

Given the biproportional Poisson model (4) restated below with (Y, X, p, q) as shorthand for $(X^{(t)}, \bar{X}, p^{(t)}, q^{(t)})$:

$$\begin{aligned} Y_{ij} &\sim \text{Poisson}(e^{u_i} X_{ij} e^{-v_j}) \text{ for } X_{ij} > 0, \\ p_i &= \sum_{j, X_{ij}>0} Y_{ij} \\ q_j &= \sum_{i, X_{ij}>0} Y_{ij}, \end{aligned}$$

we briefly explained in Section 4 why the maximum likelihood estimation problem

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

given data X, p, q may not have a finite solution. Here we provide a bit more detail on this matter. The key to the well-posedness problem is the gap between the strong and the weak existence conditions discussed in Qu et al. (2023). Suppose all the Poisson random variables Y_{ij} happen to be non-zero. Then the strong existence condition is satisfied, i.e., there exists a matrix (in fact Y) with the same zero pattern as the base matrix X with marginals p, q, so the ML estimation problem has a unique (normalized) finite solution. However, as soon as any of the Poisson variables Y_{ij} is zero, only the weak condition is guaranteed, since the matrix Y inherits all zeros of X, but now contains additional zeros. We know that in this case, the balancing problem with (X, p, q) does not admit a finite solution. Sinkhorn still converges, but the estimated parameters u, v diverge to $\pm\infty$. In practice, we only observe p, q, so there is no easy way to check whether any of the *latent* individual Poisson variables Y_{ij} is zero, but in principle, it is possible that the generated data (p, q) is not consistent with the base matrix X. The corresponding case in the choice setting is exactly when some set of items always wins. We need to actually show that this event happens with small probability. We are now ready to prove Theorem 4.2.

Proof of Theorem 4.2. Recall that the negative log-likelihood of the biproportional Poisson model is equivalent to

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

The Hessian is given by

$$\mathcal{L}(u,v) := \begin{bmatrix} \mathcal{D}(e^u X e^{-v} \mathbf{1}_m) & -e^u X e^{-v} \\ -(e^u X e^{-v})^T & \mathcal{D}((e^u X e^{-v})^T \mathbf{1}_n) \end{bmatrix},$$

and the gradient is given by

$$(\sum_j e^{u_i} X_{ij} e^{-v_j} - Y_{ij}, -\sum_i e^{u_i} X_{ij} e^{-v_j} + Y_{ij}).$$

Note that each Y_{ij} is an independent Poisson random variable with mean and variance parameter $e^{u_i} X_{ij} e^{-v_j}$. Now for each non-empty $I \subseteq [m]$ and $J \subseteq [n]$ with $X_{IJ^C} = 0$, i.e., $X_{ij} \equiv 0$ for any (i, j) satisfying $i \in I^C, j \in J$, let E_{IJ} be the event that $\sum_{j \in J} q_j = \sum_{i \in I} p_i$. The event E_{IJ} corresponds to the case where the realized network of Poisson variables being *disconnected*, hence the MLE being unbounded. Our goal is to bound the union of all events E_{IJ} for $X_{IJ^C} = 0$. We have

$$\begin{aligned}
\mathbb{P}[E_{IJ}] &= \mathbb{P}\left[\sum_{i \in I, j \in J} Y_{ij} = \sum_{i \in I, j \in [n]} Y_{ij}\right] \\
&= \mathbb{P}\left[\sum_{i \in I, j \in J^C} Y_{ij} = 0\right] \\
&= \mathbb{P}\left[\sum_{i \in I, j \in J^C} Y_{ij} - e^{u_i^*} X_{ij} e^{-v_j^*} = -\sum_{i \in I, j \in J^C} e^{u_i^*} X_{ij} e^{-v_j^*}\right] \\
&\leq \mathbb{P}\left[\sum_{i \in I, j \in J^C} Y_{ij} - e^{u_i^*} X_{ij} e^{-v_j^*} \leq -\sum_{i \in I, j \in J^C} e^{u_i^*} X_{ij} e^{-v_j^*}\right] \\
&\leq \exp(-\frac{1}{2} \sum_{i \in I, j \in J^C} e^{u_i^*} X_{ij} e^{-v_j^*}),
\end{aligned}$$

where in the last step we used Bennett's inequality (Bennett, 1962).

Now we connect the bound above to the Hessian evaluated at the true parameters. Note that for any $i \in [m]$ and $j \in [n]$,

$$e^{u_i^*} X_{ij} e^{-v_j^*} = -\mathcal{L}_{ij}(u^*, v^*),$$

and letting $e_I \in \mathbb{R}^m$ be the indicator vector of I and $e_J \in \mathbb{R}^n$ be the indicator vector of J, we have

$$\sum_{i \in I, j \in J^C} e^{u_i^*} X_{ij} e^{-v_j^*} = -\begin{bmatrix} e_I \\ e_J \end{bmatrix}^T \mathcal{L}(u^*, v^*) \begin{bmatrix} \mathbf{1}_m - e_I \\ \mathbf{1}_n - e_J \end{bmatrix}$$

where we have used the fact that $X_{ij} \equiv 0$ for any (i, j) satisfying $i \in I^C, j \in J$. Now using once again that $\mathbf{1}_{m+n}$ spans the null space of $\mathcal{L}(u, v)$, we have

$$\begin{aligned}
\sum_{i \in I, j \in J^C} e^{u_i} X_{ij} e^{-v_j} &= \begin{bmatrix} e_I \\ e_J \end{bmatrix}^T \mathcal{L}(u^*, v^*) \begin{bmatrix} e_I \\ e_J \end{bmatrix} \\
&\geq \lambda_{-2}(\mathcal{L}(u^*, v^*)) \frac{(|I| + |J|)(m + n - |I| - |J|)}{m + n}.
\end{aligned}$$

As a result, we have the following bound on the probability of event E_{IJ}:

$$\begin{aligned}
\mathbb{P}[E_{IJ}] &\leq \exp(-\frac{1}{2} \begin{bmatrix} e_I \\ e_J \end{bmatrix}^T \mathcal{L}(u^*, v^*) \begin{bmatrix} e_I \\ e_J \end{bmatrix}) \\
&\leq \exp(-\frac{1}{2} \lambda_{-2}(\mathcal{L}(u^*, v^*)) \frac{(|I| + |J|)(m + n - |I| - |J|)}{m + n}).
\end{aligned}$$

Now applying a union bound on all I, J with $X_{IJ^C} = 0$, we have

$$\mathbb{P}(\text{MLE does not exist}) \leq \sum_{I \subsetneq [m], J \subsetneq [n], X_{IJ^C}=0} \exp(-\frac{1}{2}\lambda_{-2}(\mathcal{L}(u^*, v^*)) \frac{(|I|+|J|)(m+n-|I|-|J|)}{m+n})$$
$$\leq 2\left[\left(1+\exp(-\frac{1}{4}\lambda_{-2}(\mathcal{L}(u^*, v^*)))\right)^{m+n} - 1\right].$$

Finally, plugging in $\lambda_{-2}(\mathcal{L}(u^*, v^*)) \geq 2\log(m+n)$ gives

$$\mathbb{P}(\text{MLE does not exist}) \leq 2\left[\left(1+\exp(-\frac{1}{2}\log(m+n))\right)^{m+n} - 1\right]$$
$$= 2\left[\left(1+(m+n)^{-1/2}\right)^{m+n} - 1\right]$$
$$\leq \frac{2}{\sqrt{m+n}}.$$

□

Intuition on sufficient condition (8). We provide some intuition why a "large" $\lambda_{-2}(\mathcal{L}^*)$ ensures finite solutions exist. Consider again scaling $\bar{X}$ by a constant $c > 1$, which also scales up $\lambda_{-2}(\mathcal{L}^*)$ and the Poisson parameters $e^{u_i^*}\bar{X}_{ij}e^{-v_j^*}$ by c. As c increases, the probability of the Poisson entries $X_{ij}^{(t)}$ drawing zero decreases exponentially. As a result, $X^{(t)}$ is much more likely to have the same zero patterns as $\bar{X}$, which implies bounded MLEs (Qu et al., 2023).

D. Details on our convergence algorithm, `ConvIPF`

In this section, we provide details on our algorithm, `ConvIPF`, for achieving IPF convergence. First, we review the conditions for IPF convergence and discuss two implications: (1) IPF will always converge given a true time-aggregated network (Corollary D.1), (2) IPF will always converge on data generated from our network model (Corollary D.2). Then, we discuss our algorithm, `ConvIPF`, which repeats three subroutines: `MAX-FLOW`, `BLOCKING-SET`, and `MODIFY-X`. In Appendix D.1, we describe `MAX-FLOW`, which is also described in Idel (2016). In Appendix D.2, we provide our algorithm for `BLOCKING-SET` and prove that it efficiently finds a blocking set of rows. In Appendix D.3, we formalize the minimization objective in `MODIFY-X` and present two principled approaches to solving it.

Conditions for IPF convergence. Three equivalent conditions (Pukelsheim, 2014) that define exactly when IPF converges, given inputs X, p, and q, are:

1. There exists a matrix M with row sums p and column sums q such that $M_{ij} = 0$ wherever $X_{ij} = 0$.
2. For all row subsets $S \subseteq [m]$, $\sum_{i \in S} p_i \leq \sum_{j \in N_X(S)} q_j$, where $N_X(S)$ represents the set of columns connected to S in X.
3. There exist positive diagonal matrices D^0, D^1 such that $D^0 X D^1$ has row sums p and column sums q.

Condition (1) yields the `MAX-FLOW` algorithm for testing whether IPF will converge, which we describe in Appendix D.1. Condition (2) motivates our `BLOCKING-SET` algorithm (Appendix D.2), which efficiently identifies a "blocking set" of rows for which the condition is violated. Conditions (1) and (3) yield the following useful corollaries, which establish two settings where IPF is guaranteed to converge.

Corollary D.1. *Let $\mathcal{T} = \{t_1, t_2, \cdots, t_T\}$ represent a set of timesteps and $\bar{X} = \sum_{t' \in \mathcal{T}} X^{(t')}$ represent the time-aggregated network. Given the time-varying network $X^{(t)}$, with row marginals $p^{(t)}$ and column marginals $q^{(t)}$, IPF will always converge on inputs $\bar{X}$, $p^{(t)}$, and $q^{(t)}$ if $t \in \mathcal{T}$.*

Proof. We know that IPF must converge on $X^{(t)}$, $p^{(t)}$, and $q^{(t)}$, since $X^{(t)}$ has marginals $p^{(t)}$ and $q^{(t)}$, so simply setting $d^0 = \mathbf{1}_m$ and $d^1 = \mathbf{1}_n$ solves the matrix balancing problem, satisfying Condition (3). Then, Condition (2) must also be

satisfied for $X^{(t)}$, $p^{(t)}$, and $q^{(t)}$. Since $\bar{X}$ sums over a set of matrices including $X^{(t)}$, for any row subset S in $[m]$, its set of connected columns under $\bar{X}$ must be a superset of its connected columns under $X^{(t)}$. Thus, if Condition (2) is satisfied for $X^{(t)}$, $p^{(t)}$, and $q^{(t)}$, then it must be satisfied for $\bar{X}$, $p^{(t)}$, and $q^{(t)}$; therefore, IPF converges on $\bar{X}$, $p^{(t)}$, and $q^{(t)}$. □

Corollary D.2. *Let $X^{(t)}, p^{(t)}, q^{(t)}$ represent data generated from our network model (4), given any $\bar{X}$ and scaling parameters u and v. IPF will always converge on $\bar{X}$, $p^{(t)}$, and $q^{(t)}$.*

Proof. Here, we do not require $\bar{X}$ to be a sum over some set of matrices that includes $X^{(t)}$. However, a similar logic applies: first, by the same argument as above, we know that IPF must converge on $X^{(t)}$, $p^{(t)}$, and $q^{(t)}$. Furthermore, under our model 4, $X^{(t)}$ adopts all zeros in $\bar{X}$ (with potential additional zeros). So, again, for any row subset S in $[m]$, its set of connected columns under $\bar{X}$ must be a superset of its connected columns under $X^{(t)}$. Thus, if Condition (2) is satisfied for $X^{(t)}$, $p^{(t)}$, and $q^{(t)}$, then it must be satisfied for $\bar{X}$, $p^{(t)}$, and $q^{(t)}$; therefore, IPF converges on $\bar{X}$, $p^{(t)}$, and $q^{(t)}$. □

These corollaries establish useful facts relating IPF, our network inference problem, and our model. Note that for Corollary D.1, we do not require the data to be generated from our model, only that the true time-aggregated network is given. In contrast, for Corollary D.2, we do not require that the true time-aggregated network is given, only that the data is generated from the model. These corollaries also serve as "certificates" of data quality and model correctness. If IPF does *not* converge, then the statements above cannot be true: we do not observe the true time-aggregated network $\bar{X}$ (at least one that includes t) and the data $p^{(t)}$ and $q^{(t)}$ is not generated from the model given $\bar{X}$. This motivates our perspective of IPF non-convergence as an issue of missing data in $\bar{X}$, and the need for additional, but minimal, new edges. In Appendix E.2, we also discuss real-world reasons for missingness in $\bar{X}$, with mobility data as an example.

Our algorithm, `ConvIPF`, achieves IPF convergence by iteratively identifying a blocking set of rows, for which the second convergence condition is violated, and unblocking those rows by minimally adding edges that connect those rows to new columns. Since our algorithm can be applied generally to any application of IPF, not only in the network inference setting, we use generic notation of X, p, and q in the following sections, to represent general IPF inputs. Our algorithm repeats three subroutines, `MAX-FLOW`, `BLOCKING-SET`, and `MODIFY-X`, until IPF converges:

1. Run `MAX-FLOW` to test for convergence. If IPF converges, then the algorithm is finished. If IPF does not converge, move on to Step 2.
2. Since IPF does not converge, run `BLOCKING-SET` to identify a blocking set of rows, S.
3. Run `MODIFY-X` to unblock S by minimally adding edges to X.

Our algorithm is similar in structure to the multi-item auction procedure described by Demange et al. (1986) for finding market-clearing prices. Their procedure iteratively (1) checks if there is a perfect matching of item-buyer based on current prices, (2) if not, finds the set of constricted buyers S and their neighbors $N(S)$, (3) adjusts prices in $N(S)$ to attempt to fix the matching. These steps map directly onto our steps 1-3 and step 2 is especially related because of the connection between blocking row sets in IPF and constricted sets in bipartite matching (Hall, 1935), which we discuss in Appendix D.2. Below, we describe each of our subroutines in detail.

D.1. `MAX-FLOW`

For completeness, we describe the algorithm from Idel (2016) for testing whether IPF will converge on inputs X, p, and q. Condition (1) described that IPF converges if and only if there exists a matrix A with row sums p and column sums q such that $A_{ij} = 0$ wherever $X_{ij} = 0$, i.e., A inherits the zeros of X. Note that this matrix is *more general* than the set of possible solutions to the matrix balancing problem, since A does not have to be a biproportional scaling of X. Now, the following algorithm will check for the existence of A. This algorithm is closely related to the max-flow-based algorithm from Kumar et al. (2015) to test their concept of graph consistency, further establishing connections between IPF and discrete choice.

`MAX-FLOW` algorithm. Create a new directed graph G_f 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 on G_f. If the maximum flow is equal to $\sum_i p_i = \sum_j q_j$, then the desired matrix A exists, meaning IPF converges for X, p, and q; otherwise, it does not converge.

Proof. The reasoning is as follows: the matrix A can be constructed from the flow values, where A_{ij} is set to $flow(n_i, n_j)$ (i.e., the flow along edge $n_i \to n_j$) wherever $X_{ij} > 0$ and 0 otherwise. This satisfies the constraint that A inherits the zeros of X. Since the total flow is equal to $\sum_i p_i$ and the source node s is only connected to the row nodes, with capacity p_i along each edge, then each row node n_i must have exactly p_i flowing through it. Due to the conservation of flow, it must be true that $p_i = \sum_j flow(n_i, n_j)$ for all rows i, so the row marginals are satisfied. A similar argument can be made for the columns: since the sink node t is only connected to the column nodes, with capacity q_j along each edge, then each column node n_j must have exactly q_j flowing through it, so $\sum_i flow(n_i, n_j) = q_j$ for all columns j. Thus, A is a matrix that inherits the zeros of X and satisfies the row and column marginals. □

D.2. BLOCKING-SET

Given inputs X, p, and q, where we know IPF does not converge, this subroutine identifies a blocking set of rows S for which Condition (2) is violated, i.e., $\sum_{i \in S} p_i > \sum_{j \in N_X(S)} q_j$. The naive approach to iterate through all subsets until a violation is found, but this approach is extremely inefficient, as there are 2^m possible subsets. Instead, our subroutine imports ideas from matching theory and constricted sets to design a much more efficient algorithm.

Connection to constricted sets. The idea of blocking subsets in IPF is closely related to the concept of *constricted sets* in bipartite matching problems. Given a bipartite graph $G = L \cup R$, a constricted set is a set of nodes $S \subseteq L$ such that $|N(S)| < |S|$, where $N(S)$ represents the set of neighbors in R who are connected to S. Then, a perfect matching on G (where all nodes are matched) exists if and only if no constricted set in G exists (Hall, 1935).

Given a graph without a perfect matching, one can efficiently find a constricted set via an alternating breadth-first search (BFS) algorithm. First, start with the maximal matching from the graph. Then, run BFS from an unmatched node on the *right* side. Since the graph is bipartite, BFS will alternate between nodes on the right side and nodes on the left side. The key is that we only keep *alternating* edges during BFS: when we move from the right to the left side, we keep edges that are unused in the matching; when we move from the left to the right side, we keep edges that are used in the matching. When alternating BFS terminates, the set of nodes visited on the *right* side forms a constricted set. This is due to the following reasons, as explained in Easley & Kleinberg (2010):

1. All of the nodes visited on the *left* side must be matched. Otherwise, we would have an "augmenting path", i.e., one that starts with an unmatched node on the right and ends with an unmatched node on the left, alternating between unused and used edges. This is called an augmenting path because by flipping the edges from unused to used and used to unused, we can increase the size of the matching. We can assume that no augmenting paths appear in alternating BFS, since we already found the maximal matching; thus, all visited nodes on the left must be matched.

2. Each odd layer (left side) has the same number of nodes as the subsequent layer (right side), since we follow used edges from left to right and each visited node on the left side is matched, so in the subsequent layer, we take each left node's partner. So, there are strictly more nodes in even (right) layers than odd (left) layers, since we start with a node on the right.

3. Every node in an even (right) layer has all of its neighbors in the graph visited during BFS, since it was added by its match in the previous layer then we add all of its non-match neighbors in the following layer.

Thus, the nodes visited on the right side form a constricted set, since all of their neighbors are included in the odd layers during BFS, but there are strictly more nodes in the even than odd layers.

BLOCKING-SET algorithm. We propose the following algorithm, inspired by the alternating BFS algorithm for constricted sets, for finding a blocking row set in IPF. First, from running MAX-FLOW on the flow graph G_f to test for convergence, we have flow values for each row/column. We say that a row i reached its capacity if the flow passing through it is p_i and a column j reached its capacity if the flow passing through it is q_j. Since the maximum flow is not equal to $\sum_i p_i$, there must be at least one row i that does not reach its capacity. Construct an undirected bipartite graph G where the nodes on the left and right are the rows and columns of X, respectively, and they are connected wherever $X_{ij} > 0$ (this graph is similar to but distinct from the flow graph G_f, which is directed and only has edges in the direction of rows to columns). Run the following variant of BFS on G starting from node n_i. As in the alternating BFS algorithm, each layer of BFS here will alternate between nodes on the left (rows) and nodes on the right (columns). When progressing from a

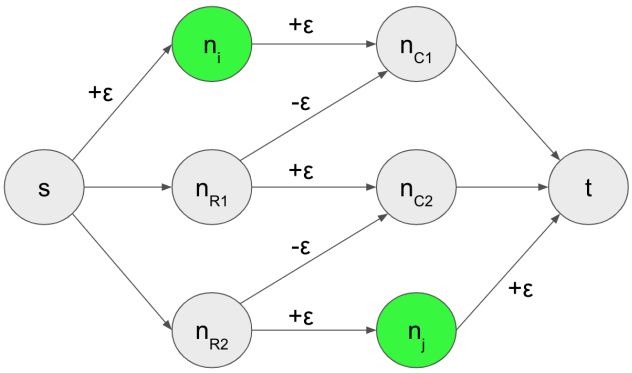


Figure D.2. Example of how we would modify G_f to increase the overall flow by ϵ, given nodes n_i and n_j (in green) that have *not* reached capacity.

column node n_C to its neighboring row node n_R, only include row nodes where $n_R \to n_C$ has non-zero flow in G_f. When progressing from row nodes to column nodes, include all (unvisited) neighbors. When this version of BFS terminates, the set of row nodes visited forms a blocking subset of rows.

Proof. First, we can use an argument similar to that of augmenting paths from bipartite matching. Here, imagine that we encountered a column j during BFS that has *not* reached capacity. Then, there is a path from node n_i to node n_j in G; for example, $n_i - n_{C_1} - n_{R_1} - n_{C_2} - n_{R_2} - n_j$, where C_1, C_2 and R_1, R_2 are other columns and rows, respectively. Then, we could increase the overall flow in G_f by adding ϵ to $s \to n_i$ and $n_i \to n_{C_1}$, removing ϵ from $n_{R_1} \to n_{C_1}$ and adding ϵ to $n_{R_1} \to n_{C_2}$, removing ϵ from $n_{R_2} \to n_{C_2}$ and adding ϵ to $n_{R_2} \to n_j$, and adding ϵ to $n_j \to t$ (Figure D.2). In other words, the flows at n_{R_1}, n_{R_2}, n_{C_1}, and n_{C_2} remain the same, but we increase the overall flow by rerouting additional flow through n_i and n_j. Note that this is only possible since n_i and n_j are under capacity, and we restricted our BFS to paths where, for consecutive nodes n_C and n_R, $n_R \to n_C$ has non-zero flow in G_f (so we can remove ϵ from those edges). For any such path, we can do this rerouting to increase the overall flow. However, since we have found the maximum flow, such a path cannot exist. Therefore, all of the columns we visited during BFS must have reached capacity.

The remaining arguments are as follows:

1. Due to the conservation of flow in G_f, the flow going through the visited rows must be equal to the flow going through the visited columns, since our BFS includes all rows that are contributing non-zero flow to these columns.

2. The total capacity of the visited rows must be strictly greater than the total capacity of the visited columns, since we established above that all visited columns must have reached capacity, but we start with a row node that did *not* reach capacity.

3. Furthermore, all of the visited rows' neighbors are included among the visited columns, since our BFS only restricted edges from columns to rows, but not from rows to columns.

Since capacity is equivalent to the original IPF marginals, this means we have identified a set of rows whose total marginal is larger than the total marginal of their connected columns. Thus, the set of rows visited during this version of BFS forms a blocking subset. □

D.3. `MODIFY-X`

Given a blocking set S, this subroutine minimally adds edges to X to unblock S. Let X^K represent the modified X after adding new edges K and let $f(X, X^K)$ represent the change in X that we are trying to minimize. Then, our goal is to find the set of edges K^* that minimizes $f(X, X^K)$,

$$K^* = \min_K f(X, X^K), \tag{28}$$

subject to unblocking S,

$$\sum_{i \in S} p_i \leq \sum_{j \in N_{X^K}(S)} q_j. \tag{29}$$

Given a current set of edges, each additional edge (i, j) will only contribute towards satisfying (29) if row i is in S and no row in S is already connected to column j. Thus, it is clear that we should only consider edge sets K such that each column in K's edges are unique and not in the original set of S's neighbors, $N_X(S)$. Furthermore, under such K, the constraint (29) reduces to

$$\sum_{(i,j) \in K} q_j \geq \delta, \tag{30}$$

where $\delta := \sum_{i \in S} p_i - \sum_{j \in N_X(S)} q_j$ is the gap in marginals we are trying to make up to unblock S.

Below, we discuss two natural definitions of $f(X, X^K)$, along with appropriate algorithms for these objectives. Once we have a set of selected edges $\hat{K}$, we add them to X with a small uniform weight. Since we are trying to minimize $f(X, X^K)$, which may be affected by the choice of edge weight, while we are trying to increase $\sum_{j \in N_{X^K}(S)} q_j$, which is not affected by edge weight (only depends on the structure of the graph), we should keep the weight as small as possible.

D.3.1. Minimizing number of edges added

One simple objective would be to minimize the number of new edges added, so $f(X, X^K) = |K|$. The solution in this case is straightforward. Let $\bar{N}_X(S)$ represent the set of columns *not* connected to S in X. Take the top-k columns in $\bar{N}_X(S)$, ordered by q_j in descending order, that satisfy $\sum_{j=1}^{k} q_j \geq \delta$. Then, any set of edges between a row in S and these k columns will unblock S, while minimizing the number of new edges added. For example, we could arbitrarily choose a row $i^* \in S$ as the row with the largest p_i, then set $K^* = \{(i^*, j)\}_{j=1}^{k}$.

D.3.2. Minimizing change in λ_1

Minimizing the number of edges in K may feel insufficient, since the objective is coarse and leaves many degrees of freedom, such as the choice of row in S. A more nuanced objective minimizes the *spectral change* in X. Motivated by the application of epidemic spread, recall that the epidemic threshold of a network is closely related to the largest eigenvalue λ_1 of its adjacency matrix (Wang et al., 2003). In fact, attempts to reduce spreading on networks often aim to minimize λ_1 through budgeted edge removals (Tong et al., 2012; Saha et al., 2015; Li et al., 2023). So, our goal is to modify X to unblock S, but otherwise change λ_1 *as little as possible*. This goal is related to spectral sparsification, which aims to sparsify a graph (i.e., greatly reduce its edges) while approximately preserving the spectrum of the graph (Spielman & Teng, 2011). In contrast, we aim to add edges to X while minimizing change in its largest eigenvalue. So, we have

$$f(X, X^K) = |\lambda_1(X) - \lambda_1(X^K)| = \lambda_1(X^K) - \lambda_1(X). \tag{31}$$

We are able to make this simplification since, by the Perron-Frobenius Theorem, a connected graph's largest eigenvalue strictly increases when an edge is added.

To solve this problem, we use a common approximation:

$$\lambda_1(X^K) - \lambda_1(X) \approx \sum_{(i,j) \in K} \vec{u}_1(i)\vec{v}_1(j), \tag{32}$$

where $\vec{u}_1$ and $\vec{v}_1$ are the left and right eigenvectors of $\lambda_1(X)$, respectively. In the closely related problem of removing k edges from a graph G to minimize $\lambda_1(G)$, Tong et al. (2012) prove that there is only an $O(k)$ gap between approximate impact and actual impact on λ_1. Let $\lambda_1(G)$ represent the original largest eigenvalue and $\lambda_1(G - R)$ represent the eigenvalue after removing the edges in R. Then, Tong et al. (2012) show that

$$\lambda_1(G) - \lambda_1(G - R) = c \sum_{(i,j) \in R} \vec{u}_1(i)\vec{v}_1(j) + O(k). \tag{33}$$

Since they approximate each edge's impact as *independent*, then their solution simply removes the top-k edges with the largest $\vec{u}_1(i)\vec{v}_1(j)$. In our case, we want to find the opposite, i.e., the edges that minimize $\vec{u}_1(i)\vec{v}_1(j)$, while unblocking S. Our algorithm first finds the row i^* in S with the smallest $\vec{u}_i(i)$, which we should use for all edges in $\hat{K}$ since unblocking S is agnostic to which row in S is chosen but λ_1 approximately increases with larger $\vec{u}_i(i)$. Now, based on our prior analysis, we want to find a set of columns $J \subseteq \bar{N}_X(S)$, where $\sum_{j\in J} q_j \geq \delta$ (30), while minimizing $\sum_{j\in J} \vec{v}_1(j)$.

We can formulate this objective as an integer linear program (ILP), by using $x \in \mathbb{R}^{|\bar{N}_X(S)|}$ as an indicator representing which columns in $\bar{N}_X(S)$ should be included:

$$x^* = \min_x \sum_{j\in\bar{N}_X(S)} \vec{v}_1(j)\cdot x_j, \quad \text{such that} \sum_{j\in\bar{N}_X(S)} q_j \cdot x_j \geq \delta \text{ and } x_j \in \{0,1\}. \tag{34}$$

Then, we keep the set of columns J where $x_j^* = 1$ and our algorithm returns $\hat{K} = \{(i^*, j) | j \in J\}$, which is guaranteed to unblock S while approximately minimizing the change in the largest eigenvalue of X.

Our ILP in fact reduces to the classic Knapsack Problem, where, given a set of η items, each with a weight w_i and value v_i, the goal is to maximize the total value of the items chosen subject to the total weight being under a given capacity W. By flipping the signs of $v_1(j)$ and q_j in (34), our ILP becomes identical to the Knapsack Problem. The Knapsack Problem, like ILPs in general, is NP-hard, but it can be solved in pseudo-polynomial time using dynamic programming, which yields a time complexity of $O(\eta W)$ (pseudo because W is not the size of input). In our case, η corresponds to $|\bar{N}_X(S)|$, which is worst-case n if S is not connected to any columns, and W corresponds to δ, which is worst-case $\sum_i p_i$ if $r_p(S) = \sum_i p_i$ and $c_{X,q}(S) = 0$. So, we can solve our problem in pseudo-polynomial time too, or, if strictly polynomial time is desired, we can use the Fully Polynomial-Time Approximation Scheme (FPTAS), which finds a solution in $O(n^3/\epsilon)$ time that is at least $(1-\epsilon)$ times the optimal value.

Further justification of eigenscore approximation. Here, we further justify the eigenscore approximation in (32) with a derivative-based approach. We consider minimizing the increase in λ_1 in the limit when the added edge weight $\epsilon \to 0$. In other words, we consider minimizing the infinitesimal change in the largest eigenvalue when adding edges. The first observation is that for each candidate column $j \in \bar{N}_X(S)$ that we want to add an edge, it is optimal to select only one row $i \in S$ to add the edge. To see this, consider the derivative

$$\frac{\partial \lambda_1}{\partial X_{ij}} = \lim_{\epsilon\to 0} \frac{\lambda_1(X + \epsilon e_i e_j^T) - \lambda_1(X)}{\epsilon}$$

and for any $j \in \bar{N}_X(S)$ define

$$\theta_j := \min_{i\in S} \frac{\partial \lambda_1}{\partial X_{ij}}.$$

In other words, θ_j is the *minimal* infinitesimal increase (cost) in λ_1 when adding an edge from S to j, and we only need to consider the row $i \in S$ that achieves θ_j. We can therefore consider the following problem:

$$\min_x \sum_{j\in\bar{N}_X(S)} \theta_j \cdot x_j \quad \text{such that} \sum_{j\in\bar{N}_X(S)} q_j \cdot x_j \geq \delta \text{ and } x_j \in \{0,1\}.$$

This is again a canonical knapsack problem. Moreover, Corollary 2.4 of Stewart & Sun (1990) gives

$$\theta_j = \min_{i\in S} \frac{\vec{u}_1^T e_i e_j^T \vec{v}_1}{\vec{u}_1^T \vec{v}_1}$$

where $\vec{u}_1, \vec{v}_1$ are the left and right eigenvectors of λ_1, respectively. The problem can be simplified because we just choose i with minimal $\vec{u}_1^T e_i$ for any $j \in \bar{N}_X(S)$, and the problem reduces to the same knapsack problem as above (34):

$$\min_x \sum_{j \in \bar{N}_X(S)} \vec{v}_1(j) \cdot x_j$$
$$\text{such that} \sum_{j \in \bar{N}_X(S)} q_j \cdot x_j \geq \delta \text{ and } x_j \in \{0, 1\}.$$

We therefore have a rigorous justification of the above ILP problem in terms of minimizing the instantaneous increase in largest eigenvalue while unblocking the subset S.

D.4. Toy example

To build intuition, here we provide a toy example of data where IPF does not converge:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{35}$$
$$p = [1, 1, 1, 1], q = [1, 1, 2].$$

IPF will not be able to converge on these inputs, since rows 0, 1, and 2 form a blocking subset: their total row marginal is 3 but their total column marginal (for columns 0 and 1) is 2. Intuitively, the problem is that row 0 is only connected to column 0, so all of its marginal must be used on column 0, leaving no marginal left in column 0. So, even though row 1 is also connected to column 0, it must use all of its marginal on its remaining connection, column 1, which leaves no room for row 2's marginal, where row 2 is only connected to column 1.

First, we verify that `BLOCKING-SET` returns $S = \{0, 1, 2\}$. The marginal gap in this case is $\delta = 1$. For `MODIFY-X`, there is only one column that S is not connected to, which is column 2. If we use the objective of minimizing number of new edges, we would add one edge based any row in S and column 2, which unblocks S and allows IPF to converge. In comparison, if we added ϵ to all zeros, as prior works have done to enable IPF to converge, we would add five edges. Alternatively, if we use the objective of minimizing change in λ_1, our algorithm identifies row 0 as the row in S with the smallest $\vec{u}_1(i)$.[5] In this case, there is only one column, column 2, that is not connected to S, so the algorithm adds an edge between row 0 and column 2 with weight $\epsilon = 0.01$. The change in λ_1 from this modification is 0.0007. In comparison, if we connected row 1 to column 2 instead, the change is λ_1 is 0.0019 ($2.8\times$), and if we connected row 2 to column 2, the change is λ_1 is 0.0013 ($1.8\times$). If we added ϵ to all zero entries, the change in λ_1 is 0.010 ($14.3\times$), which is much larger. So, this toy example demonstrates how, even on a tiny matrix X with relatively few zero values, the difference between our algorithm and the naïve solution is sizable, and our approximation of change in λ_1 still effectively minimizes that change in practice. On real-world data with much larger X, the differences are far more dramatic: in Section E.2.2, we show on real-world mobility data that our algorithm reduces change in λ_1 by orders of billions.

E. Experiments with data

In this section, we provide details on our experiments with data and describe additional experimental results. In Appendix E.1, we describe experiments with synthetic data, which enables us to test IPF's ability to recover *true* network parameters u and v. We test IPF under correct model specification and model misspecification, trying alternate models with non-Poisson distributions or interaction terms. In Appendix E.2, we describe experiments with mobility data from SafeGraph, which provide a realistic example of our network inference setting, where only the hourly marginals and time-aggregated network are provided, with missingness in the time-aggregated network. We show that IPF fails to converge several times on this data, demonstrating the importance of principled solutions for non-convergence, and evaluate our new convergence algorithm `ConvIPF` on this data, revealing dramatic improvements over prior solutions. In Appendix E.3, we describe experiments with bikeshare data from New York City's Citibike system, where we are able to construct ground-truth hourly networks. These ground-truth networks allow us to evaluate IPF ability to estimate hourly networks in our network inference setting

[5] Following Tong et al. (2012), we use $-\vec{u}_1(i)$ when $\min_i \vec{u}_1(i)$ is negative, and similarly use $-\vec{v}_1(j)$ when $\min_j \vec{v}_1(j)$ is negative, to ensure that all eigenscores are non-negative. According to the Perron-Frobenius theorem, such eigenvectors always exist.

(with hourly marginals and time-aggregated network) and compare it to baseline methods. We also use these networks as an opportunity to test our model assumptions on real-world data.

E.1. Synthetic data

Generating synthetic data. We generate synthetic data based on our biproportional Poisson model (4). 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 $\bar{X} \in \mathbb{R}^{m \times n}$ from Uniform(0, 1).
3. For a given sparsity level $r \in [0, 1)$, we randomly select $r \cdot mn$ entries from $\bar{X}$ (without replacement) and set them to 0.
4. We sample each $X_{ij}^{(t)}$ from Poisson($e^{u_i} \bar{X}_{ij} e^{-v_j}$).
5. We set $p^{(t)}$ and $q^{(t)}$ to the row sums and column sums of $X^{(t)}$, respectively.

E.1.1. Comparing IPF and Poisson regression

First, we run IPF on $\bar{X}$, $p^{(t)}$ and $q^{(t)}$, producing parameters $d^0 \in \mathbb{R}^m$ and $d^1 \in \mathbb{R}^n$. Then, we fit a Poisson regression model on all observations where $\bar{X}_{ij} > 0$, following the construction in Section B.3. 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 3.1, 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, we plot the normalized IPF parameters versus the Poisson regression parameters. We find that they lie perfectly on the $y = x$ line, validating Theorem 3.1. We also plot the 95% confidence intervals, as provided by Python's statsmodels package for fitting generalized linear models.[6] 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.

E.1.2. Evaluating IPF over varying sparsity in $\bar{X}$

We also evaluate IPF over varying levels of sparsity in the time-aggregated matrix $\bar{X}$, since sparsity is common in real-world data, as we find in our later experiments on real-world mobility data (Section E.2) and bikeshare data (Section E.3). For each sparsity rate in $r \in \{0, 0.05, \cdots, 0.9\}$, we run 1000 random trials, where in each trial, we repeat steps 2-5 of our generative process with that sparsity rate in $\bar{X}$ (i.e., we fix e^u and e^{-v} but resample all other variables) and run IPF on the newly generated $\bar{X}$, p, and q. For each trial, we evaluate three metrics:

- The number of iterations that IPF takes to converge, following our implementation in Algorithm 1 (recall that under the model, IPF will always converge; see Corollary D.2).
- The bound on the MLE's expected estimation error, $\mathbb{E}[||(\hat{u} - u, \hat{v} - v)||_2^2]$, from Theorem 4.1, without constants:

$$\frac{\sum_{ij} e^{u_i} \bar{X}_{ij} e^{-v_j}}{\lambda_{-2}^2(\mathcal{L})}.$$

- The observed ℓ_2 errors between the IPF estimates and the true model parameters:

$$||(d^0 - e^u, d^1 - e^{-v}||_2 = \sqrt{\sum_i (d_i^0 - e^{u_i})^2 + \sum_j (d_j^1 - e^{-v_j})^2}, \tag{36}$$

  where we use the mean-normalized version of all parameters.

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

In Figure 2, we visualize our results, showing the mean and 95% CIs (from 2.5th to 97.5th percentiles over random trials) per metric over sparsity rates. We find that IPF is negatively affected by sparsity in multiple ways: the number of iterations until convergence increases, and both the expected bound and observed estimation error on the network parameters worsens. However, the rate at which these metrics change differ, with the bound growing most quickly at high levels of sparsity. To better understand why, below we analyze the connection between the bound on the MLE's expected error, which we derived in Theorem 4.1, and the observed ℓ_2 error that we evaluate in these experiments.

Recall from Theorem 4.1 that in this example, we have an in-expectation bound on the error of the MLE of the form

$$\mathbb{E}\|(\hat{u}-u,\hat{v}-v)\|_2^2 = O\left(\frac{\sum_{ij} X_{ij}}{\lambda_{-2}^2(\mathcal{L})}\right)$$

where $\hat{u} = \log d^0$ and $\hat{v} = -\log d^1$. By the mean-value theorem and the fact that the exponential function is Lipschitz on the interval $[-4, 4]$, this implies an analogous in-expectation bound on the exponentiated version which corresponds to the plots:

$$\mathbb{E}[\|(d^0 - e^u, d^1 - e^{-v})\|_2^2 \mid X] = O\left(\frac{\sum_{ij} X_{ij}}{\lambda_{-2}^2(\mathcal{L})}\right).$$

Since

$$\mathbb{E}X_{ij} = 2(1-r)$$

we have that in expectation (over the randomness of X)

$$\mathbb{E}\mathcal{L} = 2(1-r)\begin{bmatrix} nI & -11^T \\ -11^T & nI \end{bmatrix}$$

and so $\lambda_{-2}(\mathbb{E}\mathcal{L}) = 2(1-r)n$. Using Weyl's inequality and standard tools from random matrix theory (sketched below, see e.g. (Gross & Nesme, 2010) or (Anderson et al., 2010)) we know that $\lambda_{-2}(\mathcal{L}) = \lambda_{-2}(\mathbb{E}\mathcal{L}) + o(n)$ with high probability for any fixed sparsity rate $r \in [0, 1)$. So with high probability over the randomness of X,

$$\mathbb{E}[\|(d^0 - e^u, d^1 - e^{-v})\|_2^2 \mid X] = O\left(\frac{(1-r)n^2}{(1-r)^2 n^2}\right) = O\left(\frac{1}{1-r}\right). \tag{37}$$

By Markov's inequality, this shows that with 99% probability overall,

$$\|(d^0 - e^u, d^1 - e^{-v})\|_2 = O\left(\frac{1}{\sqrt{1-r}}\right)$$

and the right hand side qualitatively matches the shape observed in Figure 2 (right).

Sketch of concentration argument. We only include a sketch of this argument since it is standard in the random graph/matrix literature and not part of any of the main results of this work. By standard concentration estimates (i.e. Bernstein's inequality), the number of entries of X that are deleted from each row will be $rn \pm O(\sqrt{rn\log(n)})$. Equivalently, the number of entries that remain in each row will be $(1-r)n \pm O(\sqrt{rn\log(n)})$. Next we can compute that

$$\text{Var}X_{ij} = (1-r)\frac{16}{3} + 4r(1-r).$$

Hence

$$\begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix} = (1-r)11^T + W$$

where W is a symmetric mean-zero matrix, where the variance of each coordinate above the diagonal is $O(1-r)$. So by matrix concentration (see e.g. (Gross & Nesme, 2010)) we have that $\|W\|_{OP} = O(\sqrt{1-r}\sqrt{n\log(n)})$ with high probability. Since all of the error terms in this analysis were $O(\sqrt{n}\log(n))$, they are certainly $o(n)$ as claimed before.

E.1.3. TESTING IPF UNDER MODEL MISSPECIFICATION

One advantage of defining an explicit model (the biproportional Poisson) that IPF corresponds to is that we can now explore related, modified models. First, we show that our model can be written as a multinomial model, as it is well-known that the distribution of a set of independent Poisson variables can be decomposed as a multinomial distribution over these variables, conditioned on their sum, multiplied by the distribution of their sum (which is a Poisson variable). Let $\lambda := \sum_{ij;\bar{X}_{ij}>0} e^{u_i}\bar{X}_{ij}e^{-v_j}$. Then, our biproportional Poisson model (4) is equivalent to the following model:

$$\begin{aligned} N^{(t)} &\sim \text{Poisson}(\lambda) \\ \{X_{ij}^{(t)}\}_{\bar{X}_{ij}>0} &\sim \text{Mult}(N^{(t)}, \pi) \\ \pi_{ij} &= \frac{e^{u_i}\bar{X}_{ij}e^{-v_j}}{\sum_{i'j';\bar{X}_{i'j'}>0} e^{u_{i'}}\bar{X}_{i'j'}e^{-v_{j'}}}. \end{aligned}$$

Other models are not equivalent to the Poisson, but defining them allows us to test IPF's performance under model misspecification. Below, we define three new models, discuss their MLEs, and conduct experiments where we run IPF on data generated from these models. In all of these models, we redefine how the time-varying network $X_{ij}^{(t)}$ is generated, but $p^{(t)}$ and $q^{(t)}$ remain the row and column sums of $X^{(t)}$, respectively, and $X_{ij}^{(t)} = 0$ wherever $\bar{X}_{ij} = 0$.

Non-Poisson distributions. In our biproportional Poisson model (4), we assume each $X_{ij}^{(t)}$ is drawn from Poisson(λ_{ij}), where $\lambda_{ij} = e^u\bar{X}_{ij}e^{-v}$. Here, we explore other models that keep the same expected value λ_{ij}, but replace the Poisson distribution with other distributions for count data. These experiments build on what we proved in Theorem B.2: within a family of generalized linear models, IPF uniquely recovers the MLEs of the Poisson model. Now we test empirically how IPF performs under model misspecification, by trying non-Poisson distributions.

Exponential. First, we try an exponential distribution, since Holford (1980) showed that maximum likelihood estimators are equivalent for Poisson and exponential models when the underlying *rate* at which events occur has a log-linear relationship with covariates, and the exponential model is defined by the rate while the Poisson model by the rate multiplied by the population size. In contrast, we show that replacing the Poisson with an exponential distribution in our model does not yield the same MLE, since the log-linear relationship is with the Poisson parameter instead of a rate. Since we want the expected value of $X_{ij}^{(t)}$ to be $e^{u_i}\bar{X}_{ij}e^{-v_j}$, then the exponential rate must be $\frac{1}{e^{u_i}\bar{X}_{ij}e^{-v_j}}$. So, our model is

$$X_{ij}^{(t)} \sim \text{Exp}(\frac{1}{e^{u_i}\bar{X}_{ij}e^{-v_j}}), \text{ for } \bar{X}_{ij} > 0. \tag{38}$$

The likelihood of this model is

$$\mathcal{L}(X^{(t)}|\bar{X}, u, v) = \prod_{i,j,\bar{X}_{ij}>0} \frac{1}{e^{u_i}\bar{X}_{ij}e^{-v_j}} \exp(-\frac{X_{ij}^{(t)}}{e^{u_i}\bar{X}_{ij}e^{-v_j}}).$$

Maximizing the log-likelihood yields

$$\max_{u,v} \sum_{i,j,\bar{X}_{ij}>0} -u_i - \log\bar{X}_{ij} + v_j - \frac{X_{ij}^{(t)}}{e^{u_i}\bar{X}_{ij}e^{-v_j}}. \tag{39}$$

We can see that this log-likelihood (39) clearly does not match our Poisson model's: the signs on u_i and v_j are flipped and, unlike in our derivation of the Poisson model's log-likelihood (14), we cannot marginalize out $X_{ij}^{(t)}$, meaning that the marginals $p^{(t)}$ and $q^{(t)}$ do not suffice as sufficient statistics. Thus, it is not possible for IPF to derive the maximum likelihood estimates of u, v when the distribution is exponential, instead of Poisson.

Negative binomial. We also try a negative binomial distribution, which is a common alternative to the Poisson model for count data (Gardner et al., 1995), since it allows for overdispersed data while the Poisson assumes equidispersion (equal

mean and variance). Our random variable here is $X_{ij}^{(t)}$, and the model is defined in terms of parameters s (the number of successes) and γ (the success probability). To maintain an expected value of $e^{u_i}\bar{X}_{ij}e^{-v_j}$, we have

$$s = \frac{\gamma \cdot e^{u_i}\bar{X}_{ij}e^{-v_j}}{1-\gamma},$$

which yields the following model:

$$X_{ij}^{(t)} \sim \text{NB}(\frac{\gamma \cdot e^{u_i}\bar{X}_{ij}e^{-v_j}}{1-\gamma}, \gamma), \text{ for } \bar{X}_{ij} > 0. \tag{40}$$

Each random variable has variance

$$\frac{s(1-\gamma)}{\gamma^2} = \frac{e^{u_i}\bar{X}_{ij}e^{-v_j}}{\gamma}.$$

The negative binomial model becomes identical to our Poisson model in the limit $\gamma \to 1$, but otherwise clearly will not result in equivalent MLEs due to the change in variance and additional parameter γ.

Although our statistical theory is developed for MLE under correct specification, we remark that it applies more generally to non-Poisson distributions, as long as the conditional mean of X_{ij} is given by $e^{u_i}\bar{X}_{ij}e^{-v_j}$. This is because pseudo-Poisson maximum likelihood is known to yield a consistent estimator as long as the logarithm of the conditional mean of the outcome is linear in the covariates. In our setting, this requirement is satisfied as long as the mean of X_{ij} is of the exponential form specified in (4), without requiring the distribution to be Poisson. Since we show in Theorem 3.1 that the problem is equivalent to a Poisson regression, we can conclude that IPF yields a consistent estimate for any model with the specified log-linear means.

Experiments. We compare IPF's performance on data generated from our biproportional Poisson model (4) versus data generated from an exponential distribution (38) and from a negative binomial distribution (40). For the negative binomial, we try $\gamma \in \{0.2, 0.5, 0.8\}$; since the negative binomial is equivalent to the Poisson model when $\gamma \to 1$, smaller γ corresponds to greater deviance from our original model. To test IPF on data generated from these models, we run 1000 trials, where in each trial, we sample $\bar{X}$ from Uniform$(0, 1)$ (following Step 2 above); sample $X^{(t)}$ based on the model we are using; set $p^{(t)}$ and $q^{(t)}$ to the row sums and column sums of $X^{(t)}$, respectively; then run IPF on $\bar{X}$, $p^{(t)}$, and $q^{(t)}$. As in our previous experiment, we evaluate IPF using the ℓ_2 distance between the true model parameters and IPF estimates (36). As an additional metric, we also evaluate the cosine similarity between the true network $X^{(t)}$ and IPF-estimated network $\hat{X}^{(t)} = D^0\bar{X}D^1$, where we use cosine similarity as a scale-invariant measure of the similarity:[7]

$$\text{sim}(X^{(t)}, \hat{X}^{(t)}) = \frac{\sum_{ij} X_{ij}^{(t)}\hat{X}_{ij}^{(t)}}{||X^{(t)}||_2 \cdot ||\hat{X}^{(t)}||_2}. \tag{41}$$

As shown in Table E.1, we find that IPF performs best when the model is correctly specified (i.e., the data is generated from our biproportional Poisson model), compared to data generated from exponential or negative binomial models. As expected, IPF's performance on data from the negative binomial model also worsens as γ becomes smaller. However, IPF is still reasonably effective under model misspecification: the cosine similarities between $X^{(t)}$ and $\hat{X}^{(t)}$ remain above 0.8 for the exponential model and negative binomial models, when $\gamma \geq 0.5$.

As we did in Section E.1.2, we also test IPF over varying levels of sparsity r in $\bar{X}$, but this time with $X^{(t)}$ generated from exponential distribution (38). This experiment allows us to compare the shape of ℓ_2 error over varying levels of sparsity, when the data is correctly specified (generated from Poisson) vs. incorrectly specified (generated from exponential).[8] We visualize the results in Figure E.3: we find that the shape of the ℓ_2 curves are highly similar when the data is drawn from either distribution, and matches the expected shape of $1/\sqrt{1-r}$ (see derivation in Section E.1.2), showing the usefulness of our derived error bounds even under model misspecification.

[7]We left out this metric when we were evaluating sparsity, since it could be arbitrarily improved with sparsity due to known 0s.

[8]We are not able to test the negative binomial model with sparsity greater than 0, since the number of successes s must be strictly positive, so the model (40) ill-defined for $\bar{X}_{ij} = 0$.

Model	ℓ_2 error on u and v	Cosine sim. between $X^{(t)}$ and $\hat{X}^{(t)}$
Biproportional Poisson (4)	0.995 (0.885-1.114)	0.911 (0.907-0.914)
Negative binomial (40), $\gamma = 0.8$	1.117 (0.999-1.260)	0.892 (0.887-0.897)
Negative binomial (40), $\gamma = 0.5$	1.410 (1.266-1.572)	0.843 (0.836-0.850)
Exponential (38)	1.734 (1.519-1.976)	0.855 (0.843-0.866)
Negative binomial (40), $\gamma = 0.2$	2.226 (1.957-2.479)	0.707 (0.695-0.720)

Table E.1. Evaluating IPF's performance on synthetic data generated from our biproportional Poisson model vs. models with non-Poisson distributions. The models are approximately ordered in terms of best-to-worst performance.

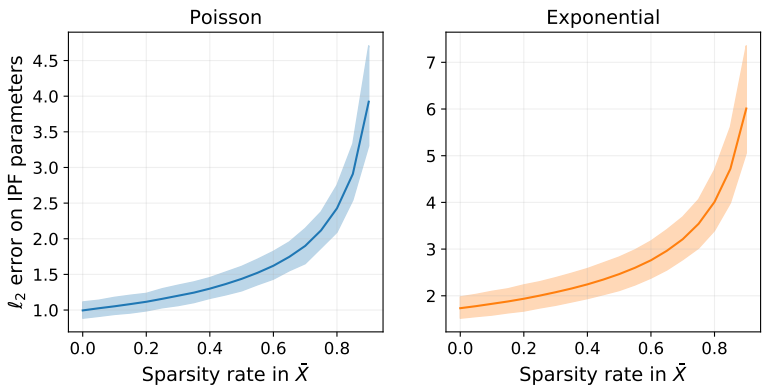


Figure E.3. Comparing sparsity rate in $\bar{X}$ to observed ℓ_2 error of IPF estimates, for data drawn from Poisson (left) vs. exponential (right) distributions. The subfigure on the left, where the model is correctly specified, exactly matches Figure 2, right. Lines represent mean and shaded region represents 95% CIs over 1000 trials.

Models with interaction terms. Our original model assumes that the expected hourly values for $X_{ij}^{(t)}$ are a scaling of $\bar{X}_{ij}$ by a row factor e^{u_i} and a column factor e^{-v_j}. However, there could also be interactions between rows and columns that influence the expected hourly values. To account for interaction, we extend our model as

$$X_{ij}^{(t)} \sim \text{Poisson}(e^{u_i}\bar{X}_{ij}e^{-v_j} \cdot d_{ij}^{\alpha}e^{-d_{ij}\beta}), \text{ for } \bar{X}_{ij} > 0, \tag{42}$$

where α and β are a new model parameters and $d_{ij} > 0$ represents an observed interaction feature, such as the geographical distance if the rows and columns correspond to physical places. This new model is inspired by the doubly constrained gravity model from Navick & Furth (1994), who estimate the number of trips between bus stops i and j as $d_{ij}^{\alpha}e^{-d_{ij}\beta}A_iB_j$, with learned row and column factors A_i and B_j which correspond to e^{u_i} and e^{-v_j}. We discuss their gravity model in detail in Section E.3.1, where we use it as a baseline against which to compare IPF. We can view this interaction model (42) as interpolating between using the time-aggregated network as the initial matrix, as a data-driven perspective, and using a gravity model to form the initial matrix, as a model-based perspective; meanwhile, continuing to learn row- and column-specific factors to match the observed time-varying marginals. Maximizing the log-likelihood of this model yields

$$\begin{aligned}
&\max_{u,v,\alpha,\beta} \sum_{i,j;\bar{X}_{ij}>0} X_{ij}^{(t)}(u_i + \log \bar{X}_{ij} - v_j + \alpha \log d_{ij} - d_{ij}\beta) - e^{u_i}\bar{X}_{ij}e^{-v_j} \cdot d_{ij}^{\alpha}e^{-d_{ij}\beta} \\
&\propto \max_{u,v,\alpha,\beta} \sum_i u_i p_i^{(t)} - \sum_j v_j^{(t)} q_j + \alpha \sum_{ij} X_{ij}^{(t)} \log d_{ij} - \beta \sum_{ij} X_{ij}^{(t)} d_{ij} - \sum_{ij} e^{u_i}\bar{X}_{ij}e^{-v_j} \cdot d_{ij}^{\alpha}e^{-d_{ij}\beta}.
\end{aligned}$$

This new log-likelihood cannot be simplified to the original log-likelihood (14) unless $\alpha = \beta = 0$. Furthermore, unlike in the biproportional Poisson model, the marginals of $X^{(t)}$ ($p^{(t)}$ and $q^{(t)}$) do not form sufficient statistics here, meaning that it cannot be used in our dynamic network inference setting where only $p^{(t)}$, $q^{(t)}$, and $\bar{X}$ are observed (but not $X^{(t)}$). However, it is still instructive to define this model, so that we can test IPF on data generated from this model.

Experiments. To generate data from the interaction model (42), first we sample 2-dimensional positions per row $i \in [m]$ and column $j \in [n]$ from Uniform$(0, 1)$, then we set the distance d_{ij} between row i and column j to be the Euclidean distance

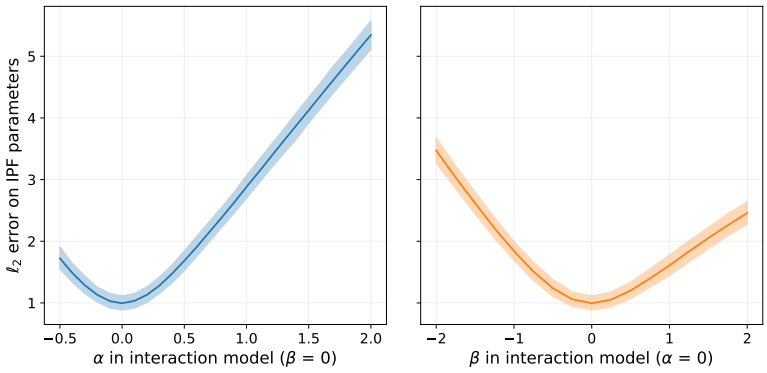


Figure E.4. IPF's ℓ_2 error on network parameters u and v (36).

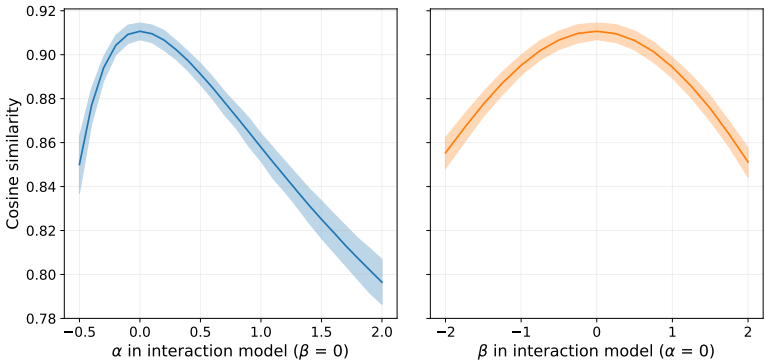


Figure E.5. Cosine similarity between true network $X^{(t)}$ and IPF-estimated network $\hat{X}^{(t)}$ (41).

Figure E.6. Evaluating IPF performance on data generated from interaction model (42), over different values of α and β.

between their positions. Then, for each value of α and β, we run 1000 trials, where (as before) in each trial we sample $\bar{X}$ from Uniform$(0, 1)$; sample $X^{(t)}$ from the interaction model with the current values of α, β, and d_{ij}; set $p^{(t)}$ and $q^{(t)}$ to the row sums and column sums of $X^{(t)}$, respectively; then run IPF on $\bar{X}$, $p^{(t)}$, and $q^{(t)}$. As before, we evaluate the ℓ_2 distance between the true model parameters and IPF estimates (36) as well as the cosine similarity between the true network $X^{(t)}$ and IPF-estimated network $\hat{X}^{(t)}$ (41). To test the impact of α alone, we fix β to 0 and try α from -0.5 to 2, which is symmetric in its impact on d_{ij} since we have d_{ij}^{α} in our model. As expected, we find that performance worsens as α deviates further from 0, but we find that over this range of α that cosine similarity between the true and estimated network remains high, at 0.8 or higher, while the ℓ_2 distance on parameters u and v is more dramatically worsened (Figure E.6, left). To test the impact of β alone, we fix α to 0 and try β from -2 to 2, which is also symmetric in its impact on d_{ij} since we have $e^{-d_{ij}\beta}$ in our model. We also find that performance worsens as β deviates from 0, but the cosine similarity remains high (above 0.85) and the ℓ_2 is less affected by β than α (Figure E.6, right). So, at least under these values of α, β, and d_{ij}, we find that IPF is relatively robust to model modifications and can still effectively infer the true network.

E.2. Experiments with real-world mobility data

We also conduct experiments with real-world mobility data, using the same aggregated location data from SafeGraph as Chang et al. (2021a;b). Like Chang et al. (2021a;b), our goal is to use the data from SafeGraph to infer hourly mobility networks that encode visits from home census block groups (CBGs), which are neighborhoods of around 600-3000 people, to individual points-of-interest (POIs), which are public locations such as restaurants, grocery stores, or gas stations. SafeGraph data provides a realistic example of where IPF would be necessary to infer a dynamic network from its 3D marginals, since SafeGraph does not provide the hourly mobility network; instead, they provide hourly total visits to POIs and from CBGs as well as an estimated, time-aggregated mobility network. Furthermore, SafeGraph's mobility data is noisy for several reasons, which creates the possibility of incorrect zeros in the time-aggregated matrix, and thus a need for our convergence

algorithm in Section 5. We discuss these reasons below.

Noise and missingness in SafeGraph data. First, unlike in the case of bikeshare data (Section E.3), where bikes are perfectly observed checking into and out of stations, SafeGraph needs to *infer* assignments to rows (i.e., POIs) and columns (i.e., CBGs). That is, SafeGraph's raw data takes the form of individual devices' GPS signals, and SafeGraph needs to perform both *visit attribution* to determine that a device is visiting a particular POI (SafeGraph, 2021) and *home attribution* to determine that a device belongs to a person who lives in a given CBG (SafeGraph, 2020b; Huang et al., 2021). Mistakes can be made at different stages of this process, such as incorrect visit attribution due to GPS drift or incorrect home attribution due to nighttime activity in a CBG that is not the person's home (e.g., if they have night shifts for work). Second, since SafeGraph relies on data from smartphones, they are missing individuals who either do not carry cell phones or opt out of data collection (SafeGraph, 2019). For example, Coston et al. (2021) showed that SafeGraph disproportionately misses older and non-white populations in their data. Third, to preserve privacy, SafeGraph documents that they apply "differential privacy techniques" to their reported time-aggregated networks (SafeGraph, 2020a). Specifically, for a given POI, when reporting time-aggregated (weekly or monthly) counts of visits from a home CBG, they add Laplacian noise to the count, drop CBGs with fewer than two devices visiting that POI, and report all visit counts from 2-4 as 4. All of these reasons lead to imperfectly observed networks and marginals, motivating the need for our convergence algorithm.

E.2.1. Constructing IPF inputs from mobility data

Using this data, our goal is to infer the mobility network at hour t from n CBGs to m POIs. We study the Richmond, Virginia metropolitan statistical area (MSA), as one of the regions studied by Chang et al. (2021b). Using their inclusion criteria, we include 9917 POIs in the MSA and 1098 CBGs that frequently visit these POIs. To capture temporal variation, we compare two days—March 2 and April 6, 2020, two Mondays just before and after the onset of the COVID-19 pandemic in the US—and infer hourly networks over their 48 hours. To infer these networks using IPF, we construct a time-aggregated network, $\bar{X}$; hourly total visitors to POIs, $p^{(t)}$; and hourly total visitors from CBGs, $q^{(t)}$. We construct these quantities from SafeGraph data in the same way as the authors did in Chang et al. (2021a;b). We summarize this procedure below, highlighting a few important facts, and refer the reader to the original text for details.

Constructing time-aggregated network $\bar{X}$. 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 of t to denote time periods longer than an hour). Since these visit matrices are sparse, we aggregate over R time periods, from January to October 2020:

$$\bar{W} = \frac{1}{R}\sum_{r=1}^{R}\hat{W}(r), \quad X_{ij} = \frac{\bar{W}_{ij}}{\sum_k W_{kj}}. \tag{43}$$

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. (2021a)). 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 E.9 (left), 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.

Constructing visitors from CBGs, $q^{(t)}$. Using SafeGraph Social Distancing Metrics (SafeGraph, 2020b), we 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

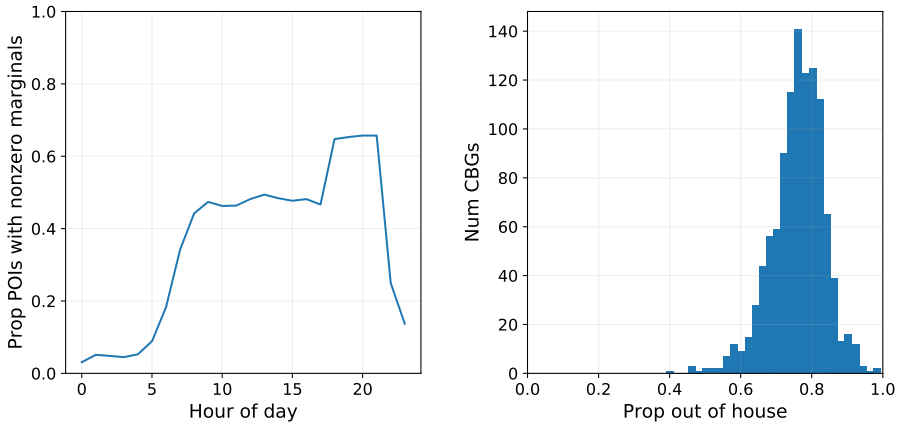


Figure E.7. Marginals on Monday, March 2, 2020.

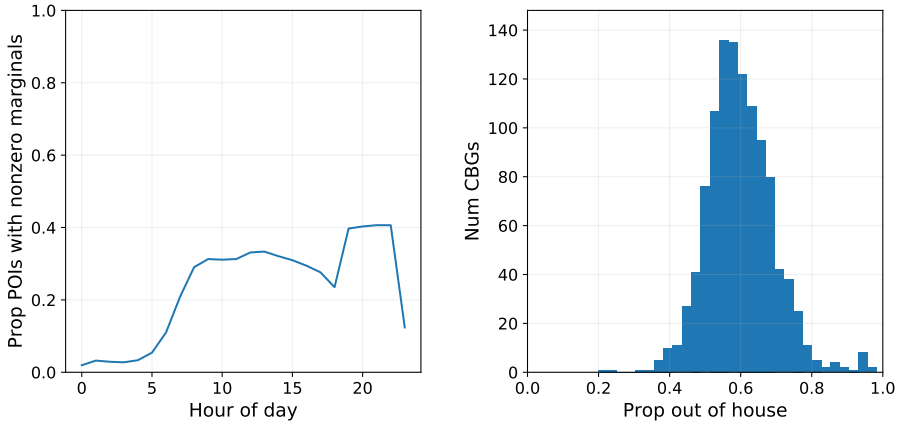


Figure E.8. Marginals on Monday, April 6, 2020.

Figure E.9. POI and CBG marginals from mobility data for Richmond, Virginia MSA.

multiplying these fractions by the CBG population N_j (from US Census), then we scale these estimates so that $q^{(t)}$ sums to $\sum_i p_i^{(t)}$. We do this since IPF requires the sums of the row and column marginals to match and because the number of people who are not at home may not exactly match the number of people who are visiting POIs. So, we have

$$q_j^{(t)} = \hat{h}_j^{(t)} N_j \cdot \frac{\sum_i p_i^{(t)}}{\sum_k \hat{h}_k^{(t)} N_k}. \tag{44}$$

In Figure E.9 (right), 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.

E.2.2. Convergence experiments on mobility data

Non-convergence. We run (modified) IPF (Algorithm 1) 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 and 4AM on April 6. In Figure E.10, 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 (Pukelsheim, 2014). This ℓ_1 error is computed as:

$$\ell_1(k) = |M^{\text{IPF}}(k) \cdot \mathbf{1} - p| + |M^{\text{IPF}}(k)^T \cdot \mathbf{1} - q|. \tag{45}$$

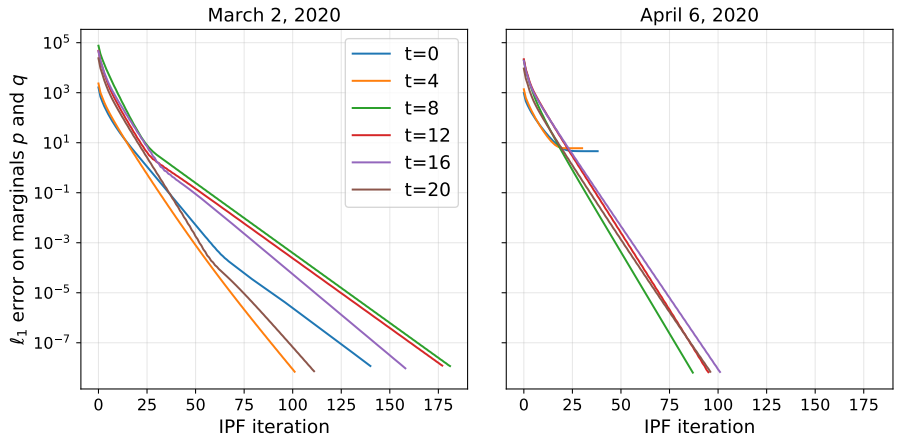


Figure E.10. ℓ_1 error on marginals p and q over IPF iterations on mobility data. We show convergence results from hours $t \in \{0, 4, 8, 12, 16, 20\}$ on March 2, 2020 (left) and April 6, 2020 (right).

	Replace all zeros	`ConvIPF`, min # edges, largest p_i	`ConvIPF`, min # edges, smallest p_i	`ConvIPF`, min change in λ_1
March 2, 2023, 2AM				
# new edges	10012193	**2**	16	5
Change in λ_1	30.04	$1.85 \cdot 10^{-8}$	$2.54 \cdot 10^{-7}$	$\mathbf{5.60 \cdot 10^{-9}}$
# `ConvIPF` iters.	–	**2**	16	5
April 6, 2023, 12AM				
# new edges	10012193	**2**	3	3
Change in λ_1	30.04	$2.14 \cdot 10^{-7}$	$1.02 \cdot 10^{-8}$	$\mathbf{9.98 \cdot 10^{-9}}$
# `ConvIPF` iters.	–	**1**	2	2
April 6, 2023, 4AM				
# new edges	10012193	**1**	9	5
Change in λ_1	30.04	$1.63 \cdot 10^{-8}$	$8.52 \cdot 10^{-8}$	$\mathbf{5.72 \cdot 10^{-9}}$
# `ConvIPF` iters.	–	**1**	9	5

Table E.2. Comparing different methods for achieving IPF convergence over three hours where the mobility data did not converge. The bolded entry in each row shows the optimal (in this case, lowest) number.

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). This intriguing observation is worthy of further study given that IPF only converges exponentially in certain settings (Pukelsheim & Simeone, 2009). 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.

Evaluating our convergence algorithm. Now, we test our convergence algorithm, `ConvIPF`, on these three hours where IPF did not converge. Recall that our convergence algorithm enables IPF to converge by making minimal modifications (edge additions) to the time-aggregated network $\bar{X}$. We test our two definitions of `MODIFY-X` (Section D.3), where we aim to either minimize the number of new edges added or minimize the change in the largest eigenvalue λ_1. For the first objective, we showed that any row in the blocking set S can be used, so, as two tie-breaking rules, we experiment with taking the row with the largest p_i and the smallest p_i. We compare our solutions to the typical solution for IPF non-convergence, which is to replace all zeros with a very small value ϵ (Lomax & Norman, 2015; Lovelace et al., 2015).

We present results in Table E.2. First, we find that all three versions of `ConvIPF` massively outperform the typical solution, for both metrics of minimizing the number of new edges and minimizing change in λ_1. Second, we find that in practice, `ConvIPF` only runs for a handful of iterations (typically, 5 or below, and at most, 16), so it terminates quickly and does not need to unblock that many blocking sets. However, each blocking set can be complex: `BLOCKING-SET` often

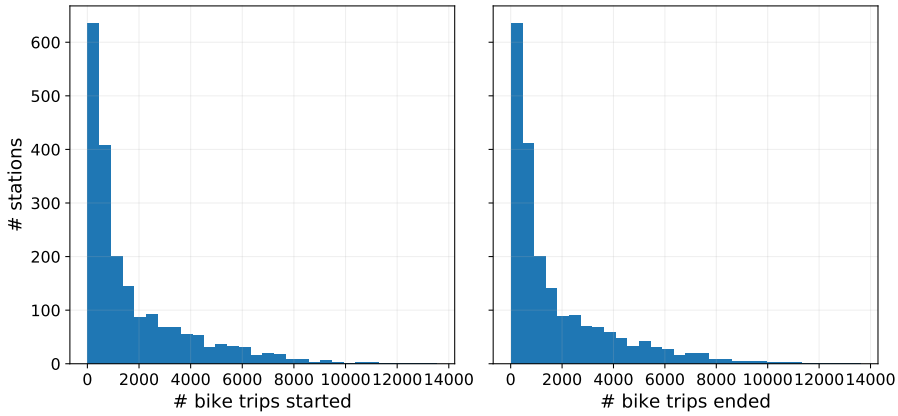


Figure E.11. Distribution of bike trips over start stations (left) and end stations (right).

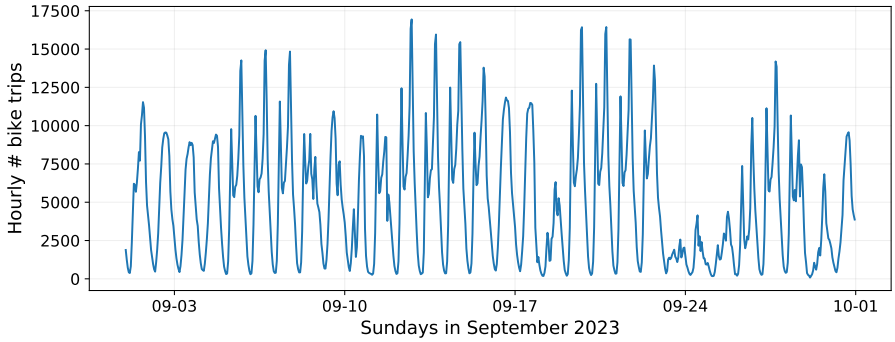


Figure E.12. Distribution of bike trips over hours.

Figure E.13. Bike trips from NYC Citibike data, from September 1 12AM to September 30 11PM, 2023.

finds blocking sets containing hundreds of rows, demonstrating the need for our efficient algorithm to find a blocking set, compared to the simple solution of iterating through row subsets. Third, we find that the choice of objective in `ConvIPF` makes a difference: choosing to minimize the number of edges (with largest p_i) always results in the fewest number of edges added and choosing to minimize the change in λ_1 always results in the smallest change in λ_1, by an order of magnitude compared to the other `ConvIPF` variants. This latter result also shows that, even with our approximation of λ_1 (Appendix D.3), the algorithm is still effective. Finally, we find that within the objective of minimizing number of edges added, taking the row with the largest p_i is consistently better than smallest p_i for minimizing both the overall number of edges added and the number of `ConvIPF` iterations. We hypothesize that this is because, within a blocking row set S, the row with the largest p_i is likely contributing more to the condition violation, $\sum_{i \in S} p_i > \sum_{j \in N_X(S)} q_j$, so modifying its connections improves the likelihood that there will not be a subset of S that still needs to be unblocked. This theory reveals an open question for future work, which is analyzing the optimal order of unblocking row sets in order to minimize the number of `ConvIPF` iterations.

E.3. Experiments with bikeshare data

In this section, we consider the case where we have *ground-truth* hourly networks. We use data from New York City's CitiBike system[9], which contains individual trips, including each trip's start time, end time, start station, and end station. For these experiments, we use trips from September 1 to September 30, 2023, which contains around 3.5 million trips. If we take the union of all start stations and end stations as the set of stations, there are $m = n = 2036$ stations in total.

The distribution of trips over stations is highly skewed (Figure E.11). To aggregate the individual trips into hourly network,

[9]We downloaded `202309-citibike-tripdata.csv` from https://citibikenyc.com/system-data.

we need to assign each trip to an hour. In 79% of trips, the trip starts and ends in the same hour, so the assignment is unambiguous. For the remaining 21% of trips, we assign the trip to the hour of its midway point. So, in our hourly network $X^{(t)}$, each $X_{ij}^{(t)}$ represents the total number of trips from station i to station j with its midpoint falling in hour t. In Figure E.12, we plot the distribution of trips over hours. We can see clear temporal patterns, e.g., fewer trips at night, two peaks on weekdays for getting to and getting from work. As usual, $p^{(t)}$ and $q^{(t)}$ are the row sums and column sums, respectively, of $X^{(t)}$. For the time-aggregated network $\bar{X}$, we explore different temporal aggregations (month, week, and day), so that we can experiment with IPF performance as the input network becomes increasingly time-aggregated.

E.3.1. Evaluating IPF performance on ground-truth networks

In these experiments, we evaluate IPF's performance at estimating the ground-truth hourly bikeshare networks, over 24 hours on September 1, 2023. We try three versions of IPF and seven baseline methods for estimating the hourly network:

1. IPF on the month-aggregated network (September 1 to September 30, 2023) and hourly marginals;
2. IPF on the week-aggregated network (September 1 to September 7, 2023) and hourly marginals;
3. IPF on the day-aggregated network (September 1, 2023) and hourly marginals;
4. IPF on the distance-based matrix and hourly marginals, equivalent to the gravity model (see details below);
5. An ablation that removes $\bar{X}$ and distributes $\sum_{ij} X^{(t)}$ proportional to $p_i^{(t)} q_j^{(t)}$;
6. An ablation that removes $q^{(t)}$ and distributes $p_i^{(t)}$ within each row proportional to $\bar{X}_{ij} / \sum_j \bar{X}_{ij}$;
7. An ablation that removes $p^{(t)}$ and distributes $q_j^{(t)}$ within each column proportional to $\bar{X}_{ij} / \sum_i \bar{X}_{ij}$.
8. A baseline where we use the month-aggregated network, scaled to match $\sum_{ij} X^{(t)}$.
9. A baseline where we use the week-aggregated network, scaled to match $\sum_{ij} X^{(t)}$.
10. A baseline where we use the day-aggregated network, scaled to match $\sum_{ij} X^{(t)}$.

Each method produces an estimate, $\hat{X}^{(t)}$, of the true hourly network, $X^{(t)}$ We use cosine similarity as a scale-invariant measure of the similarity between the two networks (41).

Details on gravity model. The gravity model (Zipf, 1946) is a classic model of human mobility, inspired by Newton's law of gravity, that assumes that the amount of travel between two regions is proportional to their population sizes divided by their geographical distance. Navick & Furth (1994) show that a "doubly constrained" gravity model of origin-destination matrices—where the total number of departures and arrivals are observed and must be matched—is equivalent to running IPF, where the initial matrix is replaced with a distance matrix with $D_{ij} = f(d_{ij})$, where $f(\cdot)$ is a function representing how trip propensities vary with distance. This convenient connection between IPF and the gravity model enables us to use the latter as a baseline, by running IPF on D_{ij} and hourly marginals $p^{(t)}$ and $q^{(t)}$.

To compute distances between bike stations, we first estimate the latitude/longitude of each station. CitiBike reports the start latitude/longitude and end latitude/longitude of each trip, so for each station, we take the mean of the start latitude/longitude where the station is start station and end latitude/longitude where the station is the end station. Then we compute distance between all pairs of stations using Euclidean distance, which is a reasonable proxy for short distances (all trips are within New York City). To prevent $d_{ij} = 0$ for trips from and to the same station (which would force $f(d_{ij})$ to be 0), we replace those distances with $\epsilon = \min(\{d_{ij} | d_{ij} > 0\})/2$; we consider these trips still meaningful, since a rider may want to pick up and drop off their bike in the same place (e.g., if they live near that station).

Then, following Navick & Furth (1994), we define $f(\cdot)$ as

$$f(d_{ij}) = d_{ij}^{\alpha} \exp(-d_{ij}\beta), \tag{46}$$

where α and β are learnable parameters. First, we estimate the empirical distribution by dividing station pairs into distance intervals (each one of size 0.001) and calculating the mean number of bike trips in the month-aggregated network $\bar{X}$ for that

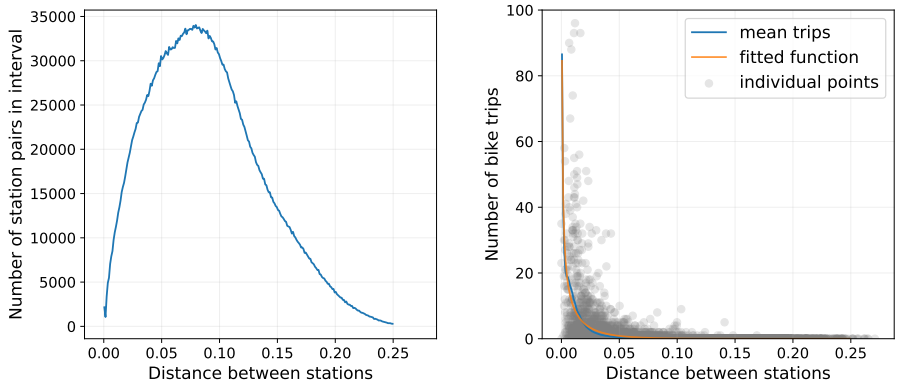


Figure E.14. Number of station pairs (left) and number of bike trips (right) over

IPF (month)	IPF (week)	IPF (day)	Gravity	No $\bar{X}$
0.363	0.389	0.496	0.204	0.116
No $p^{(t)}$	No $q^{(t)}$	Scale month	Scale week	Scale day
0.278	0.275	0.201	0.215	0.278

Table E.3. Mean cosine similarity between $X^{(t)}$ and $\hat{X}^{(t)}$, over 24 hours on September 1, 2024, for ten different methods.

interval. Then, we fit (46) to this empirical distribution, obtaining $\hat{\alpha} = -0.5863$ and $\hat{\beta} = 38.52$. The much larger magnitude of $\hat{\beta}$ is interesting, suggesting the distribution is much closer to an exponential distribution than a power distribution. In Figure E.14 (right), we plot the empirical distribution and our fitted function, along with a sample of individual data points (i.e., individual station pairs). Our fitted function matches the empirical distribution well and, as expected, the number of bike trips decreases with distance. However, from the individual data points, we can see that there is also substantial variance in the number of bike trips over station pairs of the same distance, revealing limitations of the gravity model.

Results (Table E.3). In our main text (Figure 3), we visualized results from the first seven methods listed above. First, we find that IPF strongly outperforms the gravity model, with a 78% improvement in cosine similarity even when using the month-aggregated network, demonstrating the value of using a time-aggregated network over a distance-based model. Second, IPF also outperforms the ablation baselines: when $\bar{X}$ is the month-aggregated network, IPF outperforms the ablation without $\bar{X}$ by 214% and the ablations without $p^{(t)}$ or $q^{(t)}$ by around 31%, showing the value of using all three pieces of information. Notably, removing $\bar{X}$ is much more detrimental to performance than removing $p^{(t)}$ or $q^{(t)}$, revealing which piece of information is more informative. Third, finer temporal granularity in the time-aggregated network improves IPF's performance, but the relative improvement is much larger from week to day (+27%) than from month to week (+7%). This suggests that bike trips vary more at a daily scale within the week (e.g., weekday vs. weekend) than a weekly scale within the month. However, the improvement in IPF performance at the daily level comes with a cost in convergence time, which we discuss more in Section E.3.2. In Figure E.15, we provide extended results on the scaling baselines. We find that, at all levels of time aggregation, running IPF on the time-aggregated network and hourly marginals greatly outperforms scaling the time-aggregated network to the hourly total. We also find that the day-aggregated network is much more similar to the hourly network than the week-aggregated or month-aggregated networks, further supporting the theory above that bike trips vary more at a daily than a weekly scale. Interestingly, IPF still outperforms the day-aggregated network, even when it only has access to the *month-aggregated network*, with a 31% improvement in cosine similarity. This finding reveals that, while the day-aggregated network is closer to the hourly network than the month-aggregated network, the relative information gain from the hourly marginals is still higher. This finding may have interesting implications for data providers, when deciding how to aggregate their data.

E.3.2. Convergence behavior

We observe a tradeoff between accuracy and convergence time. Using the day-aggregated network for IPF consistently outperforms the week-aggregated or month-aggregated network at recovering the true hourly network (Table E.3), but

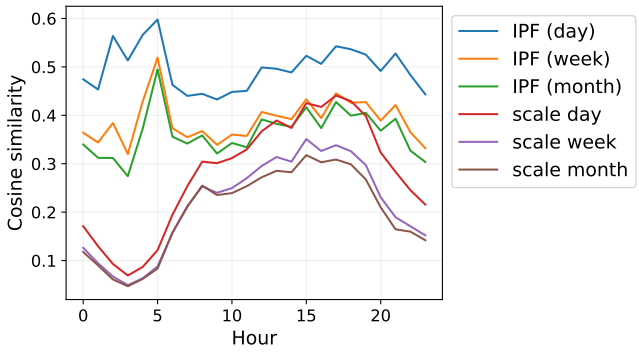


Figure E.15. Cosine similarity between ground-truth hourly networks $X^{(t)}$ from bikeshare data and inferred networks $\hat{X}^{(t)}$ from IPF and scaling baselines.

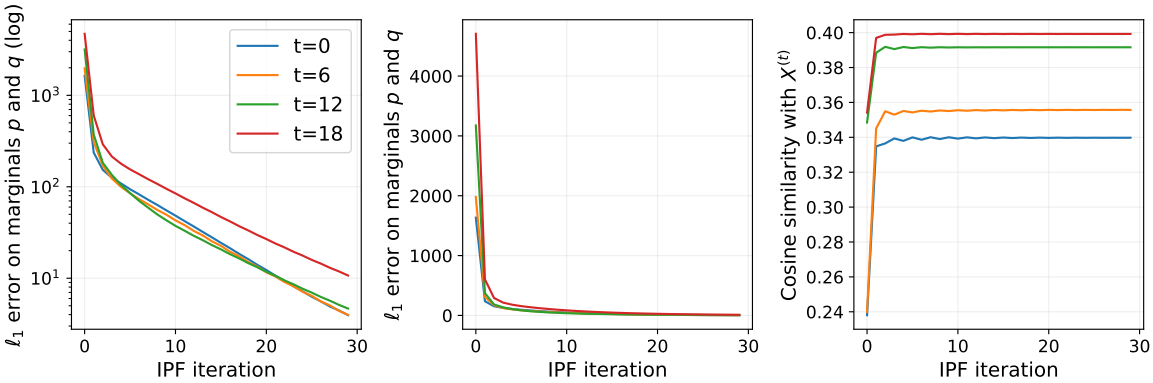


Figure E.16. IPF performance over iterations: ℓ_1 error on marginals $p^{(t)}$ and $q^{(t)}$ (with and without log scaling, left and middle); cosine similarity between $X^{(t)}$ and IPF's estimate $\hat{X}^{(t)}$ (right).

running IPF on the day-aggregated network takes substantially more iterations than using the week-aggregated network or month-aggregated network. Over the 24 hours we tested, the mean and median number of iterations when using the day-aggregated network were 7643.29 and 10000, respectively, while they were 920.79 and 419 for the week-aggregated network and 537.83 and 372 for the month-aggregated network.[10] This is not surprising, since the day-aggregated network is far sparser (2.04% nonzero) than the week-aggregated network (7.05% nonzero) or month-aggregated network (13.72% nonzero), and we know that IPF iterations increase with sparsity, which we also saw with synthetic data (Figure 2, left).

We can also evaluate IPF's performance over iterations, both in terms of its error on the target marginals as well as the cosine similarity between its estimated hourly network and the true hourly network. We test IPF on the month-aggregated network and hourly marginals, for four hours on September 1, 2023 (12 AM, 6 AM, 12 PM, and 6 PM). First, we observe that the ℓ_1 error on the marginals drops exponentially, but exhibits two different rates, one larger for the first few iterations then a slower rate for the remaining iterations (Figure E.16, left and middle). This is similar to what we found on the SafeGraph mobility data, where the convergence rate exhibited a one-time bend (Figure E.10). We show the ℓ_1 error with and without log scaling since log scaling helps to see the change in convergence rate but without log scaling emphasizes that most of the ℓ_1 error is reduced in the first few iterations. Second, when we consider the cosine similarity between the true network and IPF estimated network, we find that *all* of the improvement occurs in the first few iterations, and after that, the similarity does not improve (Figure E.16, right). Both of these analyses show that the vast majority of IPF's estimation capability is reached in the first few iterations, and running the next hundreds (or even thousands) of iterations will help IPF converge, but will not effectively improve the estimated network. This is a useful finding if efficiency is desired in network estimation; for example, if IPF is used in real-time to estimate time-varying networks, e.g., for transportation planning.

[10]These are actually underestimates, since we used a maximum of 10,000 iterations. Using the month-aggregated network never reached this maximum, week-aggregated reached it in one hour out of 24, and day-aggregated reached it in most hours.

Time-aggregation	Mean $\hat{\phi}$	25th percentile $\hat{\phi}$	75th percentile $\hat{\phi}$
Month	1.093	1.027	1.190
Week	1.086	1.046	1.159
Day	1.066	1.032	1.099

Table E.4. Distribution of dispersion parameter $\hat{\phi}$ when running IPF over 24 hours on September 1, 2023, and different time-aggregated networks.

E.3.3. Examining model assumptions on bikeshare data

In this final set of experiments, we leverage our opportunity with ground-truth hourly networks to test how reasonable the assumptions of the biproportional Poisson model (4) are.

Stationarity assumption. In Appendix B.4, we discussed how our model, which solves the network inference problem in a decoupled fashion (separately estimating parameters per t), leaves out a potential piece of information, which is that $\bar{X} = \sum_{t=1}^{T} X^{(t)}$, for some large T. We showed that the joint problem, which incorporates the constraint that the inferred $X^{(t)}$'s sum to $\bar{X}$, reduces the decoupled problem without the constraint when the following stationarity assumption is true:

$$\sum_{t=1}^{T} e^{u_{it} - v_{jt}} \approx c, \tag{47}$$

for some constant c, for all i, j where $\bar{X} > 0$. We are not able to perfectly verify this on our data, since we do not know the true parameters u or v (or whether the data comes from this model at all), but we can check the estimates for e^{u_i} and e^{-v_j}, which correspond to d_i^0 and d_j^1 in IPF, respectively. So, we want to see whether

$$\sum_{t=1}^{T} d_i^0(t) d_j^1(t) \approx c. \tag{48}$$

To check this, we fit IPF on the month-aggregated bikeshare network for all 720 hours in the month of September 2023. Then, for all i, j where $\bar{X}_{ij} > 0$, we compute $d_i^0(t) d_j^1(t)$ over all hours in the month. It turns out that the sums are quite close to each other, and close to 1, where the 5th to 95th percentile ranges from 0.88 to 1.19. These results motivate our stationarity assumption, which allows us to decouple the problem.

Overdispersion. A key assumption of our model is that the values in the hourly network follow Poisson distributions. A common issue with Poisson distributions is overdispersion, since we often find that real-world data has greater variance than the Poisson (which assumes variance is equal to the mean). To test for overdispersion, we investigate the Pearson residuals of our fitted model, which are $(y_i - \exp(\mathbf{x}_i \beta))/\sqrt{\exp(\mathbf{x}_i \beta)}$ for a generic Poisson regression model, which is equivalent to

$$r_{ij} = \frac{X_{ij}^{(t)} - d_i^0 \bar{X}_{ij} d_j^1}{\sqrt{d_i^0 \bar{X}_{ij} d_j^1}} \tag{49}$$

in our IPF setting. Recall that our set of Poisson observations $\mathcal{D}$ consist of all (i, j) where $\bar{X}_{ij} > 0$, $p_i^{(t)} > 0$, and $q_j^{(t)} > 0$. Then, we can estimate the dispersion parameter $\hat{\phi}$, which is the sum of the squared Pearson residuals divided by the degrees of freedom (number of observations minus number of model parameters):

$$\hat{\phi} = \frac{\sum_{(i,j) \in \mathcal{D}} r_{ij}^2}{|\mathcal{D}| - m - n}. \tag{50}$$

In the absence of overdispersion, the dispersion parameter should be close to 1; otherwise, it will be greater than 1. When we test the IPF estimates on the bikeshare data (24 hours on September 1, 2023), we find that the dispersion parameter is actually quite close to 1 for all three levels of time aggregations (Table E.4). Day-aggregated is the closest to 1, with a mean of 1.066, but week-aggregated and month-aggregated are still close, with means of 1.086 and 1.093, respectively. So, the data appears to be very slightly overdispersed, but not far from the Poisson assumptions.

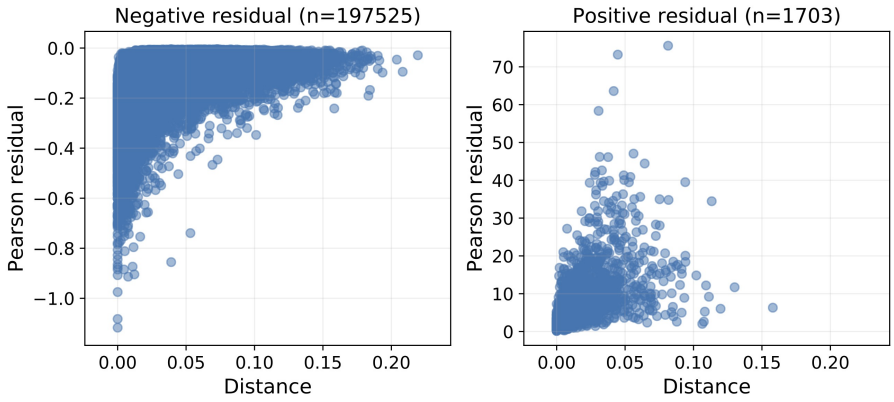


Figure E.17. Distance between stations vs. Pearson residual over station pairs i, j. Results are from IPF estimates on month-aggregated network and hourly marginals for 12 AM on September 1, 2023.

Independence assumptions and model fit. Another assumption of our model is that the values in the hourly network are independently sampled from their respective Poisson distributions. It is difficult for us to test all possible violations of this assumption, but we can test one natural dimension of correlation, which is spatial. For spatial relationships, our model should already capture spatial correlations between similar start stations (e.g., more traffic in certain neighborhoods at different times of day) and spatial correlations between similar end stations, since it learns a parameter for each row (i.e., start station) and column (i.e., end station). The question is whether there are additional interaction effects between start and end stations. Since we are interested in the interaction, we use the *distance* between the start and end stations as a dimension of interest. We described in Section E.3.1 how we compute distances between bike stations, in order to fit the gravity model, and visualized the relationship between distance and bike trips (Figure E.14).

We compare the Pearson residual r_{ij} to the distance between stations i and j for all station pairs i, j in $\mathcal{D}$. Over these pairs, we find that there *is* a relationship between distance and residuals. In Figure E.17, we show a representative example from running IPF on the month-aggregated network and hourly marginals for 12 AM on September 1, 2023. To make the visualization clearer, we split the data points into two groups: negative residuals and positive residuals. We can see that the vast majority (over 99%) of data points have negative residuals, which is expected since the true values, $X_{ij}^{(t)}$, are equal to zero most of the time, so any positive expected value will result in a negative residual. Within both the negative and positive groups, we observe a positive relationship between distances and residuals. This positive relationship is consistent over hours and time aggregations, although the correlation is weaker for smaller time aggregations.

The Pearson residuals (49) also allow us to test the Poisson model's goodness-of-fit. The numerator in (50) is known as the Pearson statistic, which follows a chi-square distribution with $n - k - 1$ degrees of freedom (where n is the number of observations and k is the number of model parameters). We find that the goodness-of-fit test is *rejected* for our model, meaning there is a statistically significant lack of fit. This is not entirely surprising, since our model is simple due to the very limited information we have about the network, but it is worth keeping in mind as a caveat when using IPF to infer networks. The lack of fit, along with the violation of independence assumptions, motivate the development of future methods that incorporate additional available information, e.g., interaction terms between features such as distance, while still only relying on the time-varying marginals and time-aggregated network to estimate the time-varying network.