# ChoiceRank: Identifying Preferences from Node Traffic in Networks

**Lucas Maystre** [1]  **Matthias Grossglauser** [1]

## Abstract

Understanding how users navigate in a network is of high interest in many applications. We consider a setting where only aggregate node-level traffic is observed and tackle the task of learning edge transition probabilities. We cast it as a preference learning problem, and we study a model where choices follow Luce's axiom. In this case, the $O(n)$ marginal counts of node visits are a sufficient statistic for the $O(n^2)$ transition probabilities. We show how to make the inference problem well-posed regardless of the network's structure, and we present ChoiceRank, an iterative algorithm that scales to networks that contains billions of nodes and edges. We apply the model to two clickstream datasets and show that it successfully recovers the transition probabilities using only the network structure and marginal (node-level) traffic data. Finally, we also consider an application to mobility networks and apply the model to one year of rides on New York City's bicycle-sharing system.

## 1  Introduction

Consider the problem of estimating click probabilities for links between pages of a website, given a hyperlink graph and aggregate statistics on the number of times each page has been visited. Naively, one might expect that the probability of clicking on a particular link should be roughly proportional to the traffic of the link's target. However, this neglects important structural effects: a page's traffic is influenced by *a)* the number of incoming links, *b)* the traffic at the pages that link to it, and *c)* the traffic absorbed by competing links. In order to successfully infer click probabilities, it is therefore necessary to disentangle the *preference* for a page (i.e., the intrinsic propensity of a user to click on a link pointing to it) from the page's *visibility* (the exposure

it gets from pages linking to it). Building upon recent work by Kumar et al. (2015), we present a statistical framework that tackles a general formulation of the problem: given a network (representing possible transitions between nodes) and the marginal traffic at each node, recover the transition probabilities. This problem is relevant to a number of scenarios (in social, information or transportation networks) where transition data is not available due to, e.g., privacy concerns or monitoring costs.

We begin by postulating the following model of traffic. Users navigate from node to node along the edges of the network by making a choice between adjacent nodes at each step, reminiscent of the random-surfer model introduced by Brin & Page (1998). Choices are assumed to be independent and generated according to Luce's model (Luce, 1959): each node in the network is chararacterized by a latent *strength* parameter, and (stochastic) choice outcomes tend to favor nodes with greater strengths. In this model, estimating the transition probabilities amounts to estimating the strength parameters. Unlike the setting in which choice models are traditionally studied (Train, 2009; Maystre & Grossglauser, 2015; Vojnovic & Yun, 2016), we do not observe distinct choices among well-identified sets of alternatives. Instead, we only have access to aggregate, marginal statistics about the traffic at each node in the network. In this setting, we make the following contributions.

1. We observe that marginal per-node traffic is a sufficient statistic for the strength parameters. That is, the parameters can be inferred from marginal traffic data without any loss of information.

2. We show that if the parameters are endowed with a prior distribution, the inference problem becomes well-posed regardless of the network structure. This is a crucial step in making the framework applicable to real-world datasets.

3. We show that model inference can scale to very large datasets. We present an iterative EM-type inference algorithm that enables a remarkably efficient implementation—each iteration requires the computational equivalent of two iterations of PageRank.

We evaluate two aspects of our framework using real-world networks. We begin by demonstrating that local preferences

---

[1]School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland. Correspondence to: Lucas Maystre <lucas.maystre@epfl.ch>.

can indeed be inferred from global traffic: we investigate the accuracy of the transition probabilities recovered by our model on three datasets for which we have ground-truth transition data. First, we consider two hyperlink graphs, representing the English Wikipedia (over two million nodes) and a Hungarian news portal (approximately 40 000 nodes), respectively. We model clickstream data as a sequence of independent choices over the links available at each page. Given only the structure of the graph and the marginal traffic at every node, we estimate the number of transitions between nodes, and we find that our estimate matches ground-truth edge-level transitions accurately in both instances. Second, we consider the network of New York City's bicycle-sharing service. For a given ride, given a pick-up station, we model the drop-off station as a choice out of a set of locations. Our model yields promising results, suggesting that our method can be useful beyond clickstream data. Next, we test the scalability of the inference algorithm. We show that the algorithm is able to process a snapshot of the WWW hyperlink graph containing over a hundred billion edges using a single machine.

**Organization of the paper.** In Section 2, we formalize the network choice model. In Section 3, we briefly review related literature. In Section 4, we present salient statistical properties of the model and its maximum-likelihood estimator, and we propose a prior distribution that makes the inference problem well-posed. In Section 5, we describe an inference algorithm that enables an efficient implementation. We evaluate the model and the inference algorithm in Section 6, before concluding in Section 7. In the supplementary material, we provide a more in-depth discussion of our model and algorithm, and we present proofs for all the theorems stated in the main text.

## 2  Network Choice Model

Let $G = (V, E)$ be a directed graph on $n$ nodes (corresponding to items) and $m$ edges. We denote the out-neighborhood of node $i$ by $N_i^+$ and its in-neighborhood by $N_i^-$. We consider the following choice process on $G$. A user starts at a node $i$ and is faced with alternatives $N_i^+$. The user chooses item $j$ and moves to the corresponding node. At node $j$, the user is faced with alternatives $N_j^+$ and chooses $k$, and so on. At any time, the user can stop. Figure 1 gives an example of a graph and the alternatives available at a step of the process.

To define the transition probabilities, we posit Luce's well-known choice axiom that states that the odds of choosing item $j$ over item $j'$ do not depend on the rest of the alternatives (Luce, 1959). This axiom leads to a unique probabilistic model of choice. For every node $i$ and every $j \in N_i^+$, the probability that $j$ is selected among alternatives $N_i^+$ can
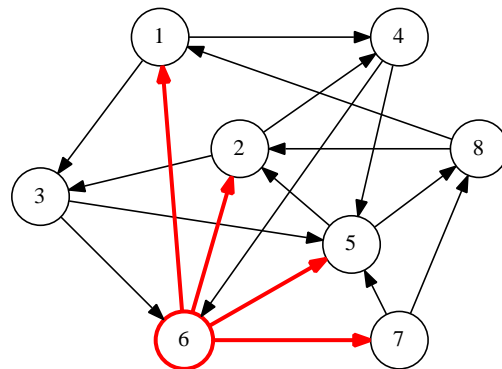


*Figure 1.* An illustration of one step of the process. The user is at node 6 and can reach nodes $N_6^+ = \{1, 2, 5, 7\}$.

be written as

$$p_{ij} = \frac{\lambda_j}{\sum_{k \in N_i^+} \lambda_k} \qquad (1)$$

for some parameter vector $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \cdots & \lambda_n \end{bmatrix}^\top \in \mathbf{R}_{>0}^n$. Intuitively, the parameter $\lambda_i$ can be interpreted as the strength (or utility) of item $i$. Note that $p_{ij}$ depends only on the out-neighborhood of node $i$. As such, the choice process satisfies the Markov property, and we can think of the sequence of choices as a trajectory in a Markov chain. In the context of this model, we can formulate the inference problem as follows. Given a directed graph $G = (V, E)$ and data on the aggregate traffic at each node, find a parameter vector $\boldsymbol{\lambda}$ that fits the data.

## 3  Related Work

A variant of the network choice model was recently introduced by Kumar et al. (2015), in an article that lays much of the groundwork for the present paper. Their generative model of traffic and the parametrization of transition probabilities based on Luce's axiom form the basis of our work. Kumar et al. define the *steady-state inversion* problem as follows: Given a graph $G$ and a target stationary distribution, find transition probabilities that lead to the desired stationary distribution. This problem formulation assumes that $G$ satisfies restrictive structural properties (strong-connectedness, aperiodicity) and is valid only asymptotically, when the sequences of choices made by users are very long. Our formulation is, in contrast, more general. In particular, we eliminate any assumptions about the structure of $G$ and cope with finite data in a principled way—in fact, our derivations are valid for choice sequences of any length. One of our contributions is to explain the steady-state inversion problem in terms of (asymptotic) maximum-likelihood inference in the network choice model. Furthermore, the statistical viewpoint that we develop also leads to *a)* a robust regularization scheme, and *b)* a simple and efficient EM-type

inference algorithm. These important extensions make the model easier to apply to real-world data.

**Luce's choice axiom.** The general problem of estimating parameters of models based on Luce's axiom has received considerable attention. Several decades before Luce's seminal book (Luce, 1959), Zermelo (1928) proposed a model and an algorithm that estimates the strengths of chess players based on pairwise comparison outcomes (his model would later be rediscovered by Bradley & Terry (1952)). More recently, Hunter (2004) explained Zermelo's algorithm from the perspective of the minorization-maximization (MM) method. This method is easily generalized to other models that are based on Luce's axiom, and it yields simple, provably convergent algorithms for maximum-likelihood (ML) or maximum-a-posteriori point estimates. Caron & Doucet (2012) observe that these MM algorithms can be further recast as expectation-maximization (EM) algorithms by introducing suitable latent variables. They use this observation to derive Gibbs samplers for a wide family of models. We take advantage of this long line of work in Section 5 when developing an inference algorithm for the network choice model. In recent years, several authors have also analyzed the sample complexity of the ML estimate in Luce's choice model (Hajek et al., 2014; Vojnovic & Yun, 2016) and investigated alternative spectral inference methods (Negahban et al., 2012; Azari Soufiani et al., 2013; Maystre & Grossglauser, 2015). Some of these results could be applied to our setting, but in general they require observing choices among well-identified sets of alternatives. Finally, we note that models based on Luce's axiom have been successfully applied to problems ranging from ranking players based on game outcomes (Zermelo, 1928; Elo, 1978) to understanding consumer behavior based on discrete choices (McFadden, 1973), and to discriminating among multiple classes based on the output of pairwise classifiers (Hastie & Tibshirani, 1998).

**Network analysis.** Understanding the preferences of users in networks is of significant interest in many domains. For brevity, we focus on literature related to hyperlink graphs. A method that has undoubtedly had a tremendous impact in this context is PageRank (Brin & Page, 1998). PageRank computes a set of scores that are proportional to the amount of time a surfer, who clicks on links randomly and uniformly, spends at each node. These scores are based only on the structure of the graph. The network choice model presented in this paper appears similar at first, but tackles a different problem. In addition to the structure of the graph, it uses the traffic at each page, and computes a set of scores that reflect the (non-uniform) probability of clicking on each link. Nevertheless, there are striking similarities in the implementation of the respective inference algorithms (see Section 6). The HOTness method proposed by Tomlin (2003) is somewhat related, but tries to tackle a harder problem. It attempts to estimate jointly the traffic and the probability of clicking on each link, by using a maximum-entropy approach. At the other end of the spectrum, BrowseRank (Liu et al., 2008) uses detailed data collected in users' browsers to improve on PageRank. Our method uses only marginal traffic data that can be obtained without tracking users.

## 4 Statistical Properties

In this section, we describe some important statistical properties of the network choice model. We begin by observing that $O(n)$ values summarizing the traffic at each node is a sufficient statistic for the $O(n^2)$ entries of the Markov-chain transition matrix. We then connect our statistical model to the steady-state inversion problem defined by Kumar et al. (2015). Guided by this connection, we study the maximum-likelihood (ML) estimate of model parameters, but find that the estimate is likely to be ill-defined in many scenarios of practical interest. Lastly, we study how to overcome this issue by introducing a prior distribution on the parameters $\boldsymbol{\lambda}$; the prior guarantees that the inference problem is well-posed.

For simplicity of exposition, we present our results for Luce's standard choice model defined in (1). Our developments extend to the model variant proposed by Kumar et al. (2015), where choice probabilities can be modulated by edge weights. In the supplementary material, we describe this variant and give the necessary adjustments to our developments.

### 4.1 Aggregate Traffic Is a Sufficient Statistic

Let $c_{ij}$ denote the number of transitions that occurred along edge $(i, j) \in E$. Starting from the transition probability defined in (1), we can write the log-likelihood of $\boldsymbol{\lambda}$ given data $\mathcal{D} = \{c_{ij} \mid (i, j) \in E\}$ as

$$
\begin{aligned}
\ell(\boldsymbol{\lambda}; \mathcal{D}) &= \sum_{(i,j) \in E} c_{ij} \left[ \log \lambda_j - \log \sum_{k \in N_i^+} \lambda_k \right] \\
&= \sum_{j=1}^{n} \sum_{i \in N_j^-} c_{ij} \log \lambda_j - \sum_{i=1}^{n} \sum_{j \in N_i^+} c_{ij} \log \sum_{k \in N_i^+} \lambda_k \\
&= \sum_{i=1}^{n} \left[ c_i^- \log \lambda_i - c_i^+ \log \sum_{k \in N_i^+} \lambda_k \right], \quad (2)
\end{aligned}
$$

where $c_i^- = \sum_{j \in N_i^-} c_{ji}$ and $c_i^+ = \sum_{j \in N_i^+} c_{ij}$ is the aggregate number of transitions arriving in and originating from $i$, respectively. This formulation of the log-likelihood exhibits a key feature of the model: the set of $2n$ counts $\{(c_i^-, c_i^+) \mid i \in V\}$ is a sufficient statistic of the $O(n^2)$ counts $\{c_{ij} \mid (i, j) \in E\}$ for the parameters $\boldsymbol{\lambda}$. (In the supplementary material, we show that it is in fact minimally sufficient.) In other words, it is enough to observe marginal

information about the number of arrivals and departures at each node—we collectively call this data the *traffic* at a node—and no additional information can be gained by observing the full choice process. This makes the model particularly attractive, because it means that it is unnecessary to track users across nodes. In several applications of practical interest, tracking users is undesirable, difficult, or outright impossible, due to *a*) privacy reasons, *b*) monitoring costs, or *c*) lack of data in existing datasets.

Note that if we make the additional assumption that the flow in the network is conserved, then $c_i^- = c_i^+$. If users' typical trajectories consist of many hops, it is reasonable to approximate $c_i^-$ or $c_i^+$ using that assumption, should one of the two quantities be missing.

## 4.2 Connection to the Steady-State Inversion Problem

In recent work, Kumar et al. (2015) define the problem of *steady-state inversion* as follows: Given a strongly-connected directed graph $G = (V, E)$ and a target distribution over the nodes $\boldsymbol{\pi}$, find a Markov chain on $G$ with stationary distribution $\boldsymbol{\pi}$. As there are $m = O(n^2)$ degrees of freedom (the transition probabilities) for $n$ constraints (the stationary distribution), the problem is in most cases underdetermined. Following Luce's ideas, the transition probabilities are constrained to be proportional to a latent score of the destination node as per (1), thus reducing the number of parameters from $m$ to $n$. Denote by $P(\boldsymbol{s})$ the Markov-chain transition matrix parametrized with scores $\boldsymbol{s}$. The score vector $\boldsymbol{s}$ is a solution for the steady-state inversion problem if and only if $\boldsymbol{\pi} = \boldsymbol{\pi} P(\boldsymbol{s})$, or equivalently

$$\pi_i = \sum_{j \in N_i^-} \frac{s_i}{\sum_{k \in N_j^+} s_k} \pi_j \quad \forall i. \tag{3}$$

In order to formalize the connection between Kumar et al.'s work and ours, we now express the steady-state inversion problem as that of asymptotic maximum-likelihood estimation in the network choice model. Suppose that we observe node-level traffic data $\mathcal{D} = \{(c_i^-, c_i^+) \mid i \in V\}$ about a trajectory of length $T$ starting at an arbitrary node. We want to obtain an estimate of the parameters $\boldsymbol{\lambda}^\star$ by maximizing the average log-likelihood $\hat{\ell}(\boldsymbol{\lambda}) = \frac{1}{T}\ell(\boldsymbol{\lambda}; \mathcal{D})$. From standard convergence results for Markov chains (Kemeny & Snell, 1976), it follows that as $G$ is strongly connected, $\lim_{T \to \infty} c_i^-/T = \lim_{T \to \infty} c_i^+/T = \pi_i$. Therefore,

$$\hat{\ell}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \left[ \frac{c_i^-}{T} \log \lambda_i - \frac{c_i^+}{T} \log \sum_{k \in N_i^+} \lambda_k \right]$$
$$\xrightarrow{T \to \infty} \sum_{i=1}^{n} \pi_i \left[ \log \lambda_i - \log \sum_{k \in N_i^+} \lambda_k \right].$$

Let $\boldsymbol{\lambda}^\star$ be a maximizer of the average log-likelihood. When $T \to \infty$, the optimality condition $\nabla \hat{\ell}(\boldsymbol{\lambda}^\star) = \mathbf{0}$ implies

$$\frac{\partial \hat{\ell}(\boldsymbol{\lambda})}{\partial \lambda_i}\bigg|_{\boldsymbol{\lambda} = \boldsymbol{\lambda}^\star} = \frac{\pi_i}{\lambda_i^\star} - \sum_{j \in N_i^-} \frac{\pi_j}{\sum_{k \in N_j^+} \lambda_k^\star} = 0$$
$$\iff \pi_i = \sum_{j \in N_i^-} \frac{\lambda_i^\star}{\sum_{k \in N_j^+} \lambda_k^\star} \pi_j \quad \forall i. \tag{4}$$

Comparing (4) to (3), it is clear that $\boldsymbol{\lambda}^\star$ is a solution of the steady-state inversion problem. As such, the network choice model presented in this paper can be viewed as a principled extension of the steady-state inversion problem to the finite-data case.

## 4.3 Maximum-Likelihood Estimate

The log-likelihood (2) is not concave in $\boldsymbol{\lambda}$, but it can be made concave using the simple reparametrization $\lambda_i = e^{\theta_i}$. Therefore, any local minimum of the likelihood is a global minimum. Unfortunately, it turns out that the conditions guaranteeing that the ML estimate is well-defined (i.e., that it exists and is unique) are restrictive and impractical. We illustrate this by providing a necessary condition, and for brevity we defer the comprehensive analysis of the ML estimate to the supplementary material. We begin with a definition that uses the notion of *hypergraph*, a generalized graph where edges may be any non-empty subset of nodes.

**Definition** (Comparison hypergraph)**.** Given a directed graph $G = (V, E)$, the *comparison hypergraph* is the hypergraph $H = (V, \mathcal{A})$, with $\mathcal{A} = \{N_i^+ \mid i \in V\}$.

Intuitively, $H$ is the hypergraph induced by the sets of alternatives available at each node. Figure 2 provides an example of a graph and of its associated comparison hypergraph. Equipped with this definition, we can state the following theorem that is a reformulation of a well-known result for Luce's choice model (Hunter, 2004).

**Theorem 1.** *If the comparison hypergraph is not connected, then for any data $\mathcal{D}$ there are $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ such that $\boldsymbol{\lambda} \neq c\boldsymbol{\mu}$ for any $c \in \mathbf{R}_{>0}$ and $\ell(\boldsymbol{\lambda}; \mathcal{D}) = \ell(\boldsymbol{\mu}; \mathcal{D})$.*

In short, the proof shows that rescaling all the parameters in one of the connected components does not change the value of the likelihood function. The network of Figure 1 illustrates an instance where the condition fails: although the graph $G$ is strongly connected, its associated comparison hypergraph $H$ (depicted in Figure 2) is disconnected, and no matter what the data $\mathcal{D}$ is, the ML estimate will never be uniquely defined. In fact, in the supplementary material, we demonstrate that Theorem 1 is just the tip of the iceberg. We provide an example where the ML estimate *does not exist* even though the comparison hypergraph is connected, and we explain that verifying a necessary and sufficient condi-
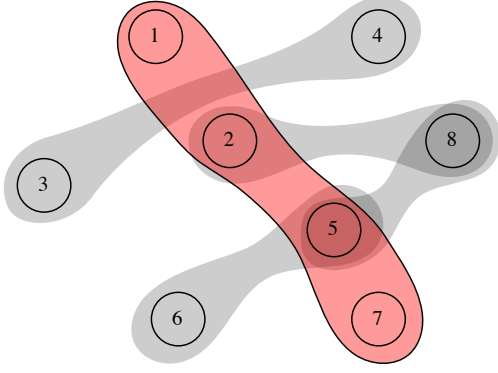
*Figure 2.* The comparison hypergraph associated to the network of Fig. 1. The hyperedge associated to $N_6^+$ is highlighted in red. Note that the component $\{3, 4\}$ is disconnected from the rest of the hypergraph.

tion for the existence of the ML estimate is computationally *more expensive* than solving the inference problem itself.

### 4.4 Well-Posed Inference

Following the ideas of Caron & Doucet (2012), we introduce an independent Gamma prior on each parameter, i.e., i.i.d. $\lambda_1, \ldots, \lambda_n \sim \text{Gamma}(\alpha, \beta)$. Adding the log-prior to the log-likelihood, we can write the log-posterior as

$$\log p(\boldsymbol{\lambda} \mid \mathcal{D}) = \sum_{i=1}^{n} \left[ (c_i^- + \alpha - 1) \log \lambda_i \right.$$
$$\left. - c_i^+ \log \sum_{k \in N_i^+} \lambda_k - \beta \lambda_i \right] + \kappa, \quad (5)$$

where $\kappa$ is a constant that is independent of $\boldsymbol{\lambda}$. The Gamma prior translates into a form of regularization that makes the inference problem well-posed, as shown by the following theorem.

**Theorem 2.** *If i.i.d.* $\lambda_1, \ldots, \lambda_n \sim \text{Gamma}(\alpha, \beta)$ *with* $\alpha > 1$*, then the log-posterior* (5) *always has a unique maximizer* $\boldsymbol{\lambda}^\star \in \mathbf{R}_{>0}^n$.

The condition $\alpha > 1$ ensures that the prior has a nonzero mode. In short, the proof of Theorem 2 shows that as a result of the Gamma prior, the log-posterior can be reparametrized into a strictly concave function with bounded super-level sets (if $\alpha > 1$). This guarantees that the log-posterior will always have exactly one maximizer. Unlike the results that we derive for the ML estimate, Theorem 2 does not impose any condition on the graph $G$ for the estimate to be well-defined.

**Remark.** Note that varying the rate $\beta$ in the Gamma prior simply rescales the parameters $\boldsymbol{\lambda}$. Furthermore, it is clear from (1) that such a rescaling affects neither the likelihood

of the observed data nor the prediction of future transitions. As a consequence, we may assume that $\beta = 1$ without loss of generality.

## 5 Inference Algorithm

The maximizer of the log-posterior does not have a closed-form solution. In the spirit of the algorithms of Hunter (2004) for variants of Luce's choice model, we develop a minorization-maximization (MM) algorithm. Simply put, the algorithm iteratively refines an estimate of the maximizer by solving a sequence of surrogates of the log-posterior. Using the inequality $\log x \le \log \tilde{x} + x/\tilde{x} - 1$ (with equality if and only if $x = \tilde{x}$), we can lower-bound the log-posterior (5) by

$$f^{(t)}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \left[ (c_i^- + \alpha - 1) \log \lambda_i \right.$$
$$\left. - c_i^+ \left( \log \sum_{k \in N_i^+} \lambda_k^{(t)} + \frac{\sum_{k \in N_i^+} \lambda_k}{\sum_{k \in N_i^+} \lambda_k^{(t)}} - 1 \right) - \beta \lambda_i \right] + \kappa,$$

with equality if and only if $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(t)}$. Starting with an arbitrary $\boldsymbol{\lambda}^{(0)} \in \mathbf{R}_{>0}^n$, we repeatedly solve the optimization problem

$$\boldsymbol{\lambda}^{(t+1)} = \arg\max_{\boldsymbol{\lambda}} f^{(t)}(\boldsymbol{\lambda}).$$

Unlike the maximization of the log-posterior, the surrogate optimization problem has a closed-form solution, obtained by setting $\nabla f^{(t)}$ to $\mathbf{0}$:

$$\lambda_i^{(t+1)} = \frac{c_i^- + \alpha - 1}{\sum_{j \in N_i^-} \gamma_j^{(t)} + \beta}, \ \gamma_j^{(t)} = \frac{c_j^+}{\sum_{k \in N_j^+} \lambda_k^{(t)}}. \quad (6)$$

The iterates provably converge to the maximizer of (5), as shown by the following theorem.

**Theorem 3.** *Let* $\boldsymbol{\lambda}^\star$ *be the unique maximum a-posteriori estimate. Then for any initial* $\boldsymbol{\lambda}^{(0)} \in \mathbf{R}_{>0}^n$ *the sequence of iterates defined by* (6) *converges to* $\boldsymbol{\lambda}^\star$.

Theorem 3 follows from a standard result on the convergence of MM algorithms and uses the fact that the log-posterior increases after each iteration. Furthermore, it is known that MM algorithms exhibit geometric convergence in a neighborhood of the maximizer (Lange et al., 2000). A thorough investigation of the convergence properties is left for future work.

The structure of the updates in (6) leads to an extremely simple and efficient implementation, given in Algorithm 1: we call it ChoiceRank. A graphical representation of an iteration from the perspective of a single node is given in Figure 3. Each iteration consists of two phases of message

**Algorithm 1** ChoiceRank

**Require:** graph $G = (V, E)$, counts $\{(c_i^-, c_i^+)\}$
1: $\boldsymbol{\lambda} \leftarrow [1, \ldots, 1]$
2: **repeat**
3:      $\boldsymbol{z} \leftarrow \boldsymbol{0}_n$                      ▷ Recompute $\boldsymbol{\gamma}$
4:      **for** $(i, j) \in E$ **do** $z_i \leftarrow z_i + \lambda_j$
5:      **for** $i \in V$ **do** $\gamma_i \leftarrow c_i^+ / z_i$
6:      $\boldsymbol{z} \leftarrow \boldsymbol{0}_n$                      ▷ Recompute $\boldsymbol{\lambda}$
7:      **for** $(i, j) \in E$ **do** $z_j \leftarrow z_j + \gamma_i$
8:      **for** $i \in V$ **do** $\lambda_i \leftarrow (c_i^- + \alpha - 1)/(z_i + \beta)$
9: **until** $\boldsymbol{\lambda}$ has converged



$$\gamma_2^{(t)} = \frac{c_2^+}{\lambda_1^{(t)} + \lambda_3^{(t)}}$$

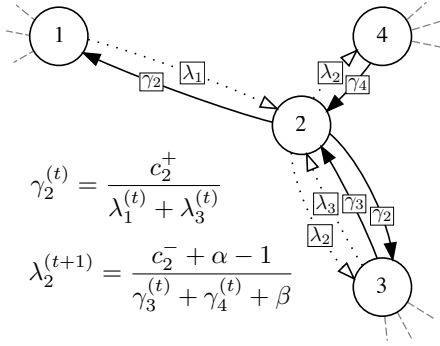$$\lambda_2^{(t+1)} = \frac{c_2^- + \alpha - 1}{\gamma_3^{(t)} + \gamma_4^{(t)} + \beta}$$

*Figure 3.* One iteration of ChoiceRank from the perspective of node 2. Messages flow in both directions along the edges of the graph $G$, first in the reverse direction (in dotted) then in the forward direction (in solid).

passing, with $\gamma_i$ flowing towards in-neighbors $N_i^-$, then $\lambda_i$ flowing towards out-neighbors $N_i^+$. The updates to a node's state are a function of the sum of the messages. As the algorithm does two passes over the edges and two passes over the vertices, an iteration takes $O(m+n)$ time. The edges can be processed in any order, and the algorithm maintains a state over only $O(n)$ values associated with the vertices. Furthermore, the algorithm can be conveniently expressed in the well-known vertex-centric programming model (Malewicz et al., 2010). This makes it easy to implement ChoiceRank inside scalable, optimized graph-processing systems such as Apache Spark (Gonzalez et al., 2014).

**EM viewpoint.** The update (6) can also be explained from an expectation-maximization (EM) viewpoint, by introducing suitable latent variables (Caron & Doucet, 2012). This viewpoint enables a Gibbs sampler that can be used for Bayesian inference. We present the EM derivation in the supplementary material, but leave a study of fully Bayesian inference in the network choice model for future work.

# 6 Experimental Evaluation

In this section, we investigate *a)* the ability of the network choice model to accurately recover transitions in real-world

scenarios, and *b)* the potential of ChoiceRank to scale to very large networks.

## 6.1 Accuracy on Real-World Data

We evaluate the network choice model on three datasets that are representative of two distinct application domains. Each dataset can be represented as a set of transition counts $\{c_{ij}\}$ on a directed graph $G = (V, E)$. We aggregate the transition counts into marginal traffic data $\{(c_i^-, c_i^+) \mid i \in V\}$ and fit a network choice model by using ChoiceRank. We set $\alpha = 2.0$ and $\beta = 1.0$ (these small values simply guarantee the convergence of the algorithm) and declare convergence when $\|\boldsymbol{\lambda}^{(t)} - \boldsymbol{\lambda}^{(t-1)}\|_1/n < 10^{-8}$. Given $\boldsymbol{\lambda}$, we estimate transition probabilities using $p_{ij} \propto \lambda_j$ as given by (1). To the best of our knowledge, there is no other published method tackling the problem of estimating transition probabilities from marginal traffic data. Therefore, we compare our method to three baselines based on simple heuristics.

**Traffic** Transitions probabilities are proportional to the traffic of the target node: $q_{ij}^T \propto c_j^-$.

**PageRank** Transition probabilities are proportional to the PageRank score of the target node: $q_{ij}^P \propto \text{PR}_j$.

**Uniform** Any transition is equiprobable: $q_{ij}^U \propto 1$.

The four estimates are compared against ground-truth transition probabilities derived from the edge traffic data: $p_{ij}^\star \propto c_{ij}$. We emphasize that although per-edge transition counts $\{c_{ij}\}$ are needed to *evaluate* the accuracy of the network choice model (and the baselines), these counts are not necessary for *learning* the model—per-node marginal counts are sufficient.

Given a node $i$, we measure the accuracy of a distribution $\boldsymbol{q}_i$ over outgoing transitions using two error metrics, the KL-divergence and the (normalized) rank displacement:

$$D_{\text{KL}}(\boldsymbol{p}_i^\star, \boldsymbol{q}_i) = \sum_{j \in N_i^+} p_{ij}^\star \log \frac{p_{ij}^\star}{q_{ij}},$$

$$D_{\text{FR}}(\boldsymbol{p}_i^\star, \boldsymbol{q}_i) = \frac{1}{|N_i^+|^2} \sum_{j \in N_i^+} |\sigma_i^\star(j) - \hat{\sigma}_i(j)|,$$

where $\sigma_i^\star$ (respectively $\hat{\sigma}_i$) is the ranking of elements in $N_i^+$ by decreasing order of $p_{ij}^\star$ (respectively $q_{ij}$). We report the distribution of errors "over choices", i.e., the error at each node $i$ is weighted by the number of outgoing transitions $c_i^+$.

### 6.1.1 Clickstream Data

**Wikipedia** The Wikimedia Foundation has a long history of publicly sharing aggregate, page-level web traffic data[1].

---

[1] See: https://stats.wikimedia.org/.

Recently, it also released clickstream data from the English version of Wikipedia (Wulczyn & Taraborelli, 2016), providing us with essential ground-truth transition-level data. We consider a dataset that contains information, extracted from the server logs, about the traffic each page of the English Wikipedia received during the month of March 2016. Each page's incoming traffic is grouped by HTTP referrer, i.e., by the page visited prior to the request. We ignore the traffic generated by external Web sites such as search engines and keep only the internal traffic (18% of the total traffic in the dataset). In summary, we obtain counts of transitions on the hyperlink graph of English Wikipedia articles. The graph contains $n = 2\,316\,032$ nodes and $m = 13\,181\,698$ edges, and we consider slightly over 1.2 billion transitions over the edges. On this dataset, ChoiceRank converges after 795 iterations.

**Kosarak** We also consider a second clickstream dataset from a Hungarian online news portal[2]. The data consists of $7\,029\,013$ transitions on a graph containing $n = 41001$ nodes and $m = 974\,560$ edges. ChoiceRank converges after 625 iterations.

The four leftmost plots of Figure 4 show the error distributions. ChoiceRank significantly improves on the baselines, both in terms of KL-divergence and rank displacement. These results give compelling evidence that transitions do not occur proportionally with the target's page traffic: in terms of KL-divergence, ChoiceRank improves on Traffic by a factor $3\times$ and $2\times$, respectively. PageRank scores, while reflecting some notion of importance of a page, are not designed to estimate transitions, and understandably the corresponding baseline performs poorly. Uniform (perhaps the simplest of our baselines) is (by design) unable to distinguish among transitions, resulting in a large displacement error. We believe that its comparatively better performance in terms of KL-divergence (for Wikipedia) is mostly an artifact of the metric, which encourages "prudent" estimates. Finally, in Figure 5 we observe that ChoiceRank seems to perform comparatively better as the number of possible transition increases.

### 6.1.2 NYC BICYCLE-SHARING DATA

Next, we consider trip data from Citi Bike, New York City's bicycle-sharing system[3]. For each ride on the system made during the year 2015, we extract the pick-up and drop-off stations and the duration of the ride. Because we want to focus on direct trips, we exclude rides that last more than one hour. We also exclude source-destinations pairs which have less than 1 ride per day on average (a majority of source-destination pairs appears at least once in the dataset). The resulting data consists of 3.4 million rides on a graph containing $n = 497$ nodes and $m = 5209$ edges. ChoiceRank converges after 7508 iterations. We compute the error distribution in the same way as for the clickstream datasets.

The two rightmost plots of Figure 4 display the results. The observations made on the clickstream datasets carry over to this mobility dataset, albeit to a lesser degree. A significant difference between clicking a link and taking a bicycle trip is that in the latter case, there is a non-uniform "cost" of a transition due to the distance between source and target. In future work, one might consider incorporating edge weights and using the weighted network choice model presented in the supplementary material.

### 6.2 Scaling ChoiceRank to Billions of Nodes

To demonstrate ChoiceRank's scalability, we develop a simple implementation in the Rust programming language, based on the ideas of COST (McSherry et al., 2015). Our code is publicly available online[4]. The implementation repeatedly streams edges from disk and keeps four floating-point values per node in memory: the counts $c_i^-$ and $c_i^+$, the sum of messages $z_i$, and either $\gamma_i$ or $\lambda_i$ (depending on the stage in the iteration). As edges can be processed in any order, it can be beneficial to reorder the edges in a way that accelerates the computation. For this reason, our implementation preprocesses the list of edges and reorders them in Hilbert curve order[5]. This results in better cache locality and yields a significant speedup.

We test our implementation on a hyperlink graph extracted from the 2012 Common Crawl web corpus[6] that contains over 3.5 billion nodes and 128 billion edges (Meusel et al., 2014). The edge list alone requires about 1 TB of uncompressed storage. There is no publicly available information on the traffic at each page, therefore we generate a value $c_i$ for every node $i$ randomly and uniformly between 100 and 500, and set both $c_i^-$ and $c_i^+$ to $c_i$. As such, this experiment does not attempt to measure the validity of the model (unlike the experiments of Section 6.1). Instead, it focuses on testing the algorithm's potential to scale to to very large networks.

**Results.** We run 20 iterations of ChoiceRank on a dual Intel Xeon E5-2680 v3 machine, with 256 GB of RAM and 6 HDDs configured in RAID 0. We arbitrarily set $\alpha = 2.0$ and $\beta = 1.0$ (but this choice has no impact on the results). Only about 65 GB of memory is used, all to store the nodes'

---

[2]The data is publicly available at `http://fimi.ua.ac.be/data/`.

[3]The data is available at `https://www.citibikenyc.com/system-data`.

[4]See: `http://lucas.maystre.ch/choicerank`.

[5]A Hilbert space-filling curve visits all the entries of the adjacency matrix of the graph, in a way that preserves locality of both source and destination of the edges.

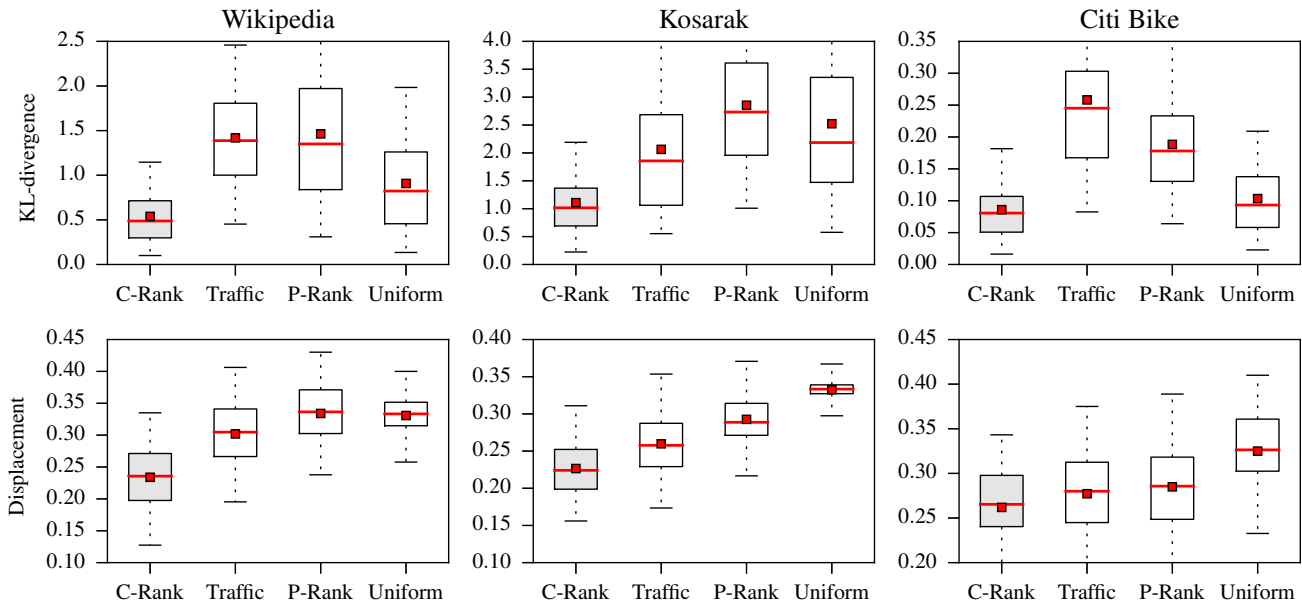[6] The data is available at `http://webdatacommons.org/hyperlinkgraph/`.

*Figure 4.* Error distributions of the network choice model and three baselines for the Wikipedia (WP) and Citi Bike (CB) datasets. The boxes show the interquartile range, the whiskers show the $5^{th}$ and $95^{th}$ percentiles, the red horizontal bars show the median and the red squares show the mean.
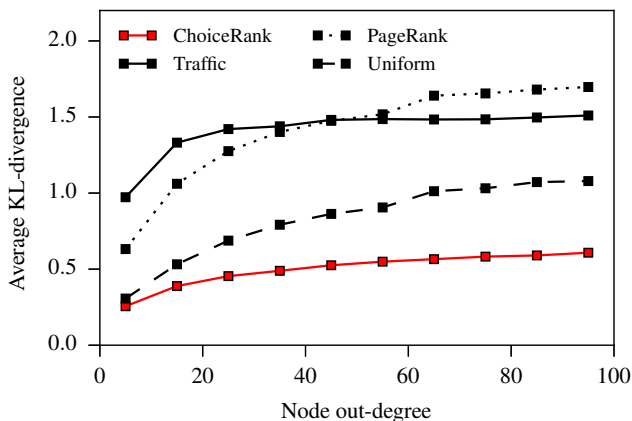


*Figure 5.* Average KL-divergence as a function of the number of possible transitions for the Wikipedia dataset. ChoiceRank performs comparatively better in the case where a node's out-degree is large.

state ($4 \times 4$ bytes per node). The algorithm takes a little less than 39 minutes per iteration on average. Collectively, these results validate the feasibility of model inference for very large datasets.

It is worth noting that despite tackling different problems, the ChoiceRank algorithm exhibits interesting similarities with a message-passing implementation of PageRank commonly used in scalable graph-parallel systems such as Pregel (Malewicz et al., 2010) and Spark (Gonzalez et al., 2014). For comparison, using the COST code (McSherry et al.,

2015) we run 20 iterations of PageRank on the same hardware and data. PageRank uses slightly less memory (about 50 GB, or one less floating-point number per node) and takes about half of the time per iteration (a little over 20 minutes). This is consistent with the fact that ChoiceRank requires two passes over the edges per iteration, whereas PageRank requires one. The similarities between the two algorithms lead us to believe that in general, ChoiceRank can benefit from any new system optimizations developed for PageRank.

## 7 Conclusion

In this paper, we present a method that tackles the problem of finding the transition probabilities along the edges of a network, given only the network's structure and aggregate node-level traffic data. This method generalizes and extends ideas recently presented by Kumar et al. (2015). We demonstrate that in spite of the strong model assumptions needed to learn $O(n^2)$ probabilities from $O(n)$ observations, the method still manages to recover the transition probabilities to a good level of accuracy on two clickstream datasets, and shows promise for applications beyond clickstream data. To sum up, we believe that our method will be useful to pracitioners interested in understanding patterns of navigation in networks from aggregate traffic data, commonly available, e.g., in public datasets.

## Acknowledgments

## References

Azari Soufiani, H., Chen, W. Z., Parkes, D. C., and Xia, L. Generalized Method-of-Moments for Rank Aggregation. In *NIPS 2013*, Lake Tahoe, CA, 2013.

Bradley, R. A. and Terry, M. E. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brin, S. and Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *WWW'98*, Brisbane, Australia, 1998.

Caron, F. and Doucet, A. Efficient Bayesian Inference for Generalized Bradley–Terry models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012.

Elo, A. *The Rating Of Chess Players, Past & Present*. Arco, 1978.

Gonzalez, J. E., Xin, R. S., Dave, A., Crankshaw, D., Franklin, M. J., and Stoica, I. Graphx: Graph Processing in a Distributed Dataflow Framework. In *OSDI'14*, pp. 599–613, 2014.

Hajek, B., Oh, S., and Xu, J. Minimax-optimal Inference from Partial Rankings. In *NIPS 2014*, Montreal, QC, Canada, 2014.

Hastie, T. and Tibshirani, R. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.

Hunter, D. R. MM algorithms for generalized Bradley–Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.

Kemeny, J. G. and Snell, J. L. *Finite Markov Chains*. Springer-Verlag, 1976.

Kumar, Ravi, Tomkins, Andrew, Vassilvitskii, Sergei, and Vee, Erik. Inverting a Steady-State. In *WSDM'15*, pp. 359–368. ACM, 2015.

Lange, K., Hunter, D. R., and Yang, I. Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

Liu, Y., Gao, B., Liu, T.-Y., Zhang, Y., Ma, Z., He, S., and Li, H. BrowseRank: Letting Web Users Vote For Page Importance. In *SIGIR'08*, pp. 451–458. ACM, 2008.

Luce, R. D. *Individual Choice behavior: A Theoretical Analysis*. Wiley, 1959.

Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., and Czajkowski, G. Pregel: A System for Large-Scale Graph Processing. In *SIGMOD'10*, pp. 135–145. ACM, 2010.

Maystre, L. and Grossglauser, M. Fast and Accurate Inference of Plackett–Luce Models. In *NIPS 2015*, Montreal, Canada, 2015.

McFadden, D. Conditional logit analysis of qualitative choice behavior. In Zarembka, P. (ed.), *Frontiers in Econometrics*, pp. 105–142. Academic Press, 1973.

McSherry, F., Isard, M., and Murray, D. G. Scalability! But at what COST? In *HotOS XV*, 2015.

Meusel, Robert, Vigna, Sebastiano, Lehmberg, Oliver, and Bizer, Christian. Graph Structure in the Web—Revisited: A Trick of the Heavy Tail. In *WWW'14 Companion*, pp. 427–432, 2014.

Negahban, S., Oh, S., and Shah, D. Iterative Ranking from Pair-wise Comparisons. In *NIPS 2012*, Lake Tahoe, CA, 2012.

Tomlin, J. A. A New Paradigm for Ranking Pages on the World Wide Web. In *WWW'03*, pp. 350–355. ACM, 2003.

Train, K. E. *Discrete Choice Methods with Simulation*. Cambridge University Press, second edition, 2009.

Vojnovic, M. and Yun, S.-Y. Parameter Estimation for Generalized Thurstone Choice Models. In *ICML 2016*, pp. 498–506, 2016.

Wulczyn, E. and Taraborelli, D. Wikipedia Clickstream. April 2016. URL https://dx.doi.org/10.6084/m9.figshare.1305770.v16.

Zermelo, E. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1928.