

Inverting a Steady-State

Ravi Kumar

Andrew Tomkins

Sergei Vassilvitskii

Erik Vee

Google

Mountain View, CA

{ravi.k53, atomkins, sergeiv, erikvee}@gmail.com

ABSTRACT

We consider the problem of inferring choices made by users based only on aggregate data containing the relative popularity of each item. We propose a framework that models the problem as that of inferring a Markov chain given a stationary distribution. Formally, we are given a graph and a target steady-state distribution on its nodes. We are also given a mapping from per-node scores to a transition matrix, from a broad family of such mappings. The goal is to set the scores of each node such that the resulting transition matrix induces the desired steady state. We prove sufficient conditions under which this problem is feasible and, for the feasible instances, obtain a simple algorithm for a generic version of the problem. This iterative algorithm provably finds the unique solution to this problem and has a polynomial rate of convergence; in practice we find that the algorithm converges after fewer than ten iterations. We then apply this framework to choice problems in online settings and show that our algorithm is able to explain the observed data and predict the user choices much better than other competing baselines across a variety of diverse datasets.

1. INTRODUCTION

The theory of Markov chains provides a beautiful algebraic formulation of the conditions under which a steady state exists for a random walk, and the nature of that steady state. Given a transition matrix of the Markov chain, one may then determine whether it meets the conditions, and compute the steady state if it exists.

In practice, however, it may be easier to observe the steady state than the underlying transitions. Consider for example a video sharing website that recommends a number of follow-on videos to the viewer of a particular video, and also displays the number of times each video has been viewed. The popularity of each video is directly proportional to the steady state distribution in the random walk defined by the follow-on videos, but an outside observer does not know which edges have been traversed to attain this steady state.

Thus, rather than determining the steady state from the matrix, we are often faced with the inverse problem:

Given the steady state, can we recover some properties of the underlying process?

Such a problem is highly under-constrained, as there are numerous ways to set potentially quadratically-many transition probabilities so as to produce the desired steady state. Our main result is a solution to this problem under a natural set of constraints.

We will assume first that the graph of the Markov chain is known. This is tantamount to assuming that certain transitions are known to be zero, or equivalently, from any state, there is a known set of available alternatives from which the user must choose. In many situations, these alternatives are present explicitly as a set of “related items” on a page: related videos on `youtube.com`, related restaurants on `yelp.com`, various types of related products on `amazon.com`, and so forth.

So far, the assumption is quite weak. We now introduce a stronger axiom about the transition matrix, which somewhat surprisingly will in general result in a *unique* Markov chain that produces the given steady state. Our axiom can be described as follows.

First, we assume that each state i of the Markov process has an unknown *score* s_i , and there is a known mapping from the the scores to the full transition matrix. We require further that this mapping is *continuous, monotonic, and unbounded*. That is, all else held constant, an increase in the score of i will always increase the transition probability of an edge to i (monotonicity), and as the score becomes sufficiently high, this transition probability will become arbitrarily close to 1 (unboundedness). Note that the most natural mapping satisfying this condition makes the probability of transition from a node to its neighbor be proportional to the neighbor’s score. To the best of our knowledge, this kind of inverse problem has not been considered in the literature before; see Section 2.

Our main result states that in this setting, if a natural condition on the graph of the Markov process holds with respect to the given steady-state distribution, then:

1. For any desired steady state, there is an assignment of scores to the nodes that attains this steady state.
2. The scores yielding a particular steady state are unique up to scalar multiplication.
3. An efficient algorithm exists to find these scores.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from `permissions@acm.org`.
WSDM’15, February 2–6, 2015, Shanghai, China.
Copyright 2015 ACM 978-1-4503-3317-7/15/02 ...\$15.00.
<http://dx.doi.org/10.1145/2684822.2685310>.

Let us give a couple of examples to develop the intuition for this model. First, consider a music recommendation site that allows users to navigate from one song to any of a set of related songs. We model that user’s likelihood of transitioning from song A to B as proportional to $\text{similarity}(A, B) \cdot \text{score}(B)$, which is monotonic and unbounded, as long as the similarity scores are positive. If we know the overall visit popularity of the songs, we can therefore learn the scores that attain these visit popularities. As another example, consider a user of a recommendation site. The user picks a next object to consider from among a known slate of candidates with probability proportional to some (increasing and unbounded) function of the candidate object’s overall score. The increasing function may be postulated or identified empirically from data. Our techniques may then be applied to learn the score of each item.

We apply our framework to four diverse choice settings, from choosing which link to follow in a Wikipedia article to reach a particular webpage, to natural restaurant selection problems, to deciding which comedy clip is funnier. We show that by only considering the *structure* of the graph and ignoring the metadata, the model can explain the data and predict the individual user choices 10 to 40% better than strong baselines based on popularity, pagerank, or maximum entropy approaches (see below for a broader discussion of related work).

2. RELATED WORK

To the best of our knowledge, the problem of inverting a steady-state distribution has not been considered in the literature before. Tomlin proposed a notion of *Temperature Rank* [31], which is related to our work. In his model, we are given the link structure on the web pages along with the steady-state traffic on all the web pages and the goal is to identify the transition probabilities on the edges by enforcing flow conservation of the traffic. Since the optimization problem is under-constrained, the idea is to use maximum-entropy regularization and a practical heuristic based on iterative matrix scaling [29] to compute the probabilities. Importantly, the transition probabilities do not depend on the neighborhood of a node but depend only on the endpoints of an edge. Even though at the end the method produces a score for each node in the graph, it is a simple heuristic with no provable guarantees. Our model, on the other hand, is based on a concrete transition probability model that takes the neighborhood of a node into account and does not need any form of regularization since the number of parameters is exactly the number of nodes. Furthermore, our algorithm is provably correct and has many additional desirable properties including uniqueness and a polynomial rate of convergence. We will use Temperature Rank as one of our baselines.

Markov chains have been studied in the literature for over a century; see the books by Kemeny and Snell [17] and Levin, Perese, and Wilmer [21]. Typical problems in Markov chains include studying the stationary distribution and mixing times. As we mentioned earlier, as far as we know, the problem of steady-state inversion has not been directly addressed in the Markov chain literature.

PageRank [7, 28] popularized Markov chains in the web search and data mining communities. PageRank postulates a random surfer model in which a uniform random walk is conducted on the link structure of the web graph, after

making it ergodic by adding a random teleportation step. The stationary distribution of this random walk yields a score, called PageRank, for each node in the graph. Our method also yields a score for each node, but both our model and goals are different from PageRank. In PageRank, the random walk (modulo the teleportation) is uniform on the neighborhood of a source node, whereas in our case, the walk is weighted by the score of a destination node. Hence, the “normalization” factor for the random walk in PageRank is proportional to the size of the out-neighborhood of a source node whereas in our case, it is the sum of the scores of all nodes in the out-neighborhood of the source node. We will in fact use PageRank as one of our baselines and show that the more general score-based method better explains the data. There have been several papers on PageRank computation (e.g., [8, 13, 15, 16, 20, 22, 25]) and other link analysis methods (e.g., [1, 5, 18, 27, 30]). For a detailed introduction to PageRank and its related methods, see the surveys by Berkhin [4], and Langville and Meyer [19].

Another work that is related in spirit to ours is the classical Metropolis–Hastings method [12, 26]. In this method, we are given a graph and a desired stationary distribution on the nodes of the graph and the goal is to design a random walk (Markov chain) such that the stationary distribution of this random walk is exactly the desired stationary distribution. This is achieved by carefully adjusting the transition probabilities in the walk. Unlike our problem, the main goal in Metropolis–Hastings is to compute the probabilities on the edges to achieve the desired node probabilities. Even though Metropolis–Hastings has been studied for more than half a century, very little is known about it from a formal point of view, especially, its mixing time and rate of convergence. For more details, see the nice survey by Chib and Greenberg [9]. In general, the problem of designing transition probabilities (or more generally, edge weights) to achieve a global objective has been studied in the optimization community. For example, Ghosh, Boyd, and Saberi [10] considered the global objective of minimizing the total effective resistance and Boyd, Diaconis, and Xiao [6] considered the objective of minimizing the mixing time. These problems are typically cast as convex optimization problems and solved directly instead of trying to define a score for each node.

Discrete choice theory is a classic topic in statistics and behavioral economics and is the closest body of work to our objective. The goal of the theory to explain and predict choices between two or more discrete alternatives. Discrete choice models are characterized by the nature of the set of alternatives (choice set), the process by which the choice probabilities are defined, and the utility derived by making a particular choice. See the books by Ben-Akiva and Lerman [3] and by Hensher, Rose, and Greene [14]. There is a variety of discrete choice models depending on what we assume about the alternatives. Our choice of choice probabilities (no recursion intended!) is loosely motivated by the axioms formulated by Duncan Luce [23, 24], which state that a user’s likelihood to select one object over another is unaffected by the nature of the other alternatives. Our framework is reminiscent of multinomial choice models: the choice set is given by the neighborhood of a node in the graph, the choice probabilities are defined according to the score of the nodes, and the implicit utility is to match the given steady-state distribution. However, our framework is

more general than choice models. First, it does not need to know which item was chosen, rather, it relies on aiming to match a desired steady-state distribution. Second, it does not assume that the random variables follow given distributions, a very common assumption in the statistical modeling of choice. Third, it has provable convergence guarantees, which are hard to obtain for many heuristic and optimization methods used in discrete choice theory. Last, it works with weighted graphs and also for a broad family of functions that can be applied to the scores of a neighborhood. As we will show formally, the unweighted case of the steady-state inversion problem is related to the choice model.

3. PRELIMINARIES

Let $G = (V, E)$ be a weighted directed graph where $V = \{1, \dots, n\}$ and $w_{ij} > 0$ is the *weight* of an edge $(i, j) \in E$. For a directed edge $(i, j) \in E$, we call j the out-neighbor of i and i the in-neighbor of j . Let $\Gamma_{\text{out}}(i)$ denote the set of out-neighbors of i and let $\Gamma_{\text{in}}(i)$ denote the set of in-neighbors of i . For a set $S \subseteq V$ let $\Gamma_{\text{in}}(S) = \cup_{v \in S} \Gamma_{\text{in}}(v)$ and $\Gamma_{\text{out}}(S) = \cup_{v \in S} \Gamma_{\text{out}}(v)$. Let G be strongly connected, i.e., any node is reachable from every node.

Let $s : V \rightarrow \mathbb{R}^{>0}$ be a *score* function on the nodes V . Such a score function naturally defines a Markov chain on V given by the following transition probability from i to j :

$$T_{G,s}(i, j) = \frac{w_{ij}s_j}{\sum_{j' \in \Gamma_{\text{out}}(i)} w_{ij'}s_{j'}}.$$

Note that the strong-connectedness of G guarantees that the Markov chain is recurrent. If the Markov chain is also aperiodic then the stationary distribution is unique. Given a score function s , let $\pi_{G,s}$ denote the stationary distribution of the Markov chain.

Let $p : V \rightarrow (0, 1]$ be a given probability distribution over the nodes of G so that $\sum_{i \in V} p_i = 1$. By removing each node i where $p_i = 0$, we will assume without loss of generality that $p_i > 0$ for all $i \in V$.

PROBLEM 1 (STEADY-STATE INVERSION). *Given a strongly connected weighted graph $G = (V, E)$, and a distribution p on V , find a score function $s : V \rightarrow \mathbb{R}^{>0}$ such that $\pi_{G,s} = p$.*

Note that a priori it is unclear if such a score function always exists or if it is unique.

4. FEASIBILITY AND SOLVABILITY

In this section we study the necessary and sufficient conditions for the feasibility of the steady-state inversion problem and show how to find the scores in polynomial time if the problem is feasible.

We first illustrate a simple example to show that a solution to the steady-state inversion problem may not always exist. Let G be the directed three node cycle $A \rightarrow B \rightarrow C \rightarrow A$, with an additional self loop on A . For any score function s , it is easy to see that the stationary distributions on B and C must be equal. If not, the the steady-state inversion problem on this graph is infeasible.

4.1 Reduction to the bipartite case

To study the conditions under which a solution exists and to find a solution, we convert the steady-state inversion problem on general graphs to a related problem on directed weighted bipartite graphs.

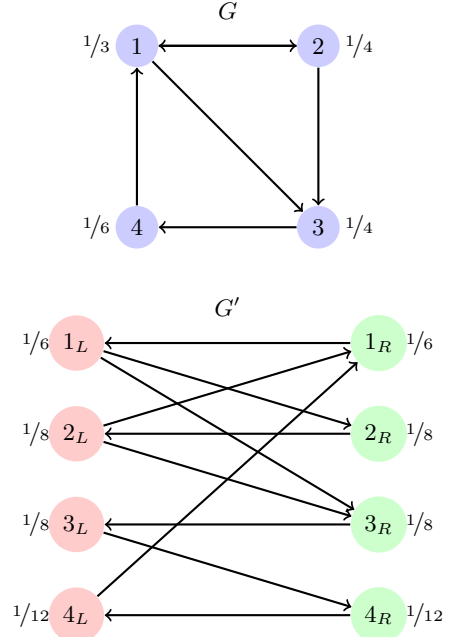


Figure 1: Reduction to the bipartite case: original graph G and the given steady-state distribution p and the bipartite form G' and the new distribution p' .

Let $G = (L, R, E)$ be a directed weighted bipartite graph. Let $T'_{G,s} : L \times R \rightarrow [0, 1]$ be a submatrix of the transition matrix defined as $T'_{G,s}(i_L, j_R) = T_{G,s}(i_L, j_R)$ for $i_L \in L$ and $j_R \in R$. Note that since G is bipartite, $T'_{G,s}$ is itself a transition matrix. Let p be a probability distribution on the nodes $L \cup R$. Let p_L (resp. p_R) be the conditional distribution on the nodes in L (resp. R). We define the following problem.

PROBLEM 2 (BIPARTITE STEADY-STATE INVERSION). *Given a directed bipartite graph $G = (L, R, E)$, and probability distributions p_L on L and p_R on R , find a score function $s : R \rightarrow \mathbb{R}^{>0}$ such that $p_R = p_L \cdot T'_{G,s}$.*

We now show how to reduce the steady-state inversion problem on graphs to bipartite steady-state inversion. Let $G = (V, E)$ be the given instance of the steady-state inversion problem and let p be the given distribution on V .

Let $G' = (L, R, E')$ be a bipartite graph derived from G in the following manner. For each node $i \in V$, create two nodes i_L and i_R : $L = \{i_L \mid i \in V\}$ and $R = \{i_R \mid i \in V\}$. For each edge $(i, j) \in E$ with weight w_{ij} , add an edge (i_L, j_R) with weight w_{ij} to E' . For each node $i \in V$, add a back edge (i_R, i_L) of weight 1 to E' . Let p' be a distribution on $L \cup R$ given by $p'(i_L) = p'(i_R) = p(i)/2$. We give an example of the reduction in Figure 1.

LEMMA 3. *Let s' be the solution to the bipartite steady-state inversion problem on G' and p' . Then, s defined by $s(i) = s'(i_R)$ is a solution to the steady-state inversion problem on G .*

PROOF. By construction, we have $p'(i_R) = p(i)/2 = p'(i_L)$. Also, by the construction of G' from G and the definition of

s from s' , we have $T'_{G',s'} = T_{G,s}$ and hence,

$$p(i) = 2p'(i_R) = 2p'(i_L) \cdot T'_{G',s'} = p(i) \cdot T_{G,s},$$

where the second equality follows since s' is a solution to the bipartite steady-state inversion problem. Therefore, s is a solution to the steady-state inversion problem on G and p . \square

4.2 Algorithm for bipartite graphs

Let $G = (L, R, E)$ be a bipartite graph and let p be a distribution on $L \cup R$. We define a notion of graph consistency and show that the steady-state inversion problem is feasible if and only if the bipartite graph is consistent.

DEFINITION 4 (CONSISTENCY). *A directed bipartite graph $G = (L, R, E)$ is consistent with a distribution p on $L \cup R$ if*

$$\sum_{i \in L} p_i = \frac{1}{2} = \sum_{j \in R} p_j,$$

and there exist values $\alpha_{ij} \geq 0$ for all $(i, j) \in E$ so that for every node $i \in L$,

$$p_i = \sum_{j \in \Gamma_{\text{out}}(i)} \alpha_{ij},$$

and for every node $j \in R$,

$$p_j = \sum_{i \in \Gamma_{\text{in}}(j)} \alpha_{ij}.$$

The consistency condition is equivalent to saying that the stationary distribution p is a result of specific choices (namely the α_{ij} 's) made by each node. Consistency is a condition that is ensured in many natural settings. For example, a bipartite graph obtained (according to the reduction in Section 4.1) from a strongly connected directed graph with self-loops at all nodes, is consistent with any distribution. Another example of a bipartite graph that is consistent with any distribution is one in which for any subset $L' \subseteq L$, we have $\sum_{i \in L'} p_i \leq \sum_{j \in R'} p_j$, where $R' = \cup_{i \in L'} \Gamma_{\text{out}}(i)$; this is related to the Hall's theorem condition [11] for the existence of a perfect matching in bipartite graphs. Observe that the three node cycle example above is consistent only when the stationary distribution on nodes B and C is the same. Clearly if the consistency condition is not met, there is no way to satisfy the steady-state equations in the original graph. However, surprisingly, the converse is also true: if the consistency condition is met, then the node scores can be set so as to realize the target steady state.

This is our main result, which we now state and prove.

THEOREM 5. *Let $G = (L, R, E)$ be a weighted bipartite graph and p be a distribution on $L \cup R$ that is consistent with G . Then for any $\epsilon > 0$, there exists a score function s such that*

$$(1 - \epsilon)p_R \leq p_L \cdot T'_{G,s} \leq (1 + \epsilon)p_R;$$

moreover, s can be computed in polynomial time.

We prove this Theorem by giving an algorithm to find the desired scores s . The algorithm will begin with some assignment of the scores, and then increase the scores for the nodes whose realized probability is far below the desired probability. To state the algorithm, for a node $j \in R$, let

$$q_j(s) = (p_L \cdot T'_{G,s})_j,$$

be the probability mass at node j after one step of the random walk from P_L , based on the scores s .

Let $\epsilon > \delta > 0$ and consider the following algorithm, which obtains a sequence of approximations to the score function. (For simplicity of exposition, we have separated the roles of δ and ϵ ; later in Theorem 10, we will set $\delta = \epsilon/2$.)

Algorithm 1 Bipartite steady-state inversion

```

1:  $t \leftarrow 0$ 
2: for  $j \in R$  do
3:   Initialize  $s_j^t = p_j$ 
4: end for
5: repeat
6:    $t \leftarrow t + 1$ 
7:   for  $j \in R$  do
8:     if  $q_j(s^{t-1}) \geq p_j(1 - \epsilon)$  then
9:        $s_j^t \leftarrow s_j^{t-1}$  {Leave score unchanged}
10:    else
11:      Solve the following for  $x$ :
12:         $q_j(s_1^{t-1}, \dots, s_{j-1}^{t-1}, x, s_{j+1}^{t-1}, \dots, s_n^{t-1}) = p_j(1 - \delta)$ 
13:       $s_j^t \leftarrow x$  {Increase score}
14:    end if
15:   end for
16: until  $\forall j \in R, s_j^t = s_j^{t-1}$ 

```

4.3 Algorithm Analysis

LEMMA 6. *Algorithm 1 is well defined, i.e., s_j^{t+1} exists as defined for every t . Furthermore, the algorithm produces a sequence of score functions s^0, s^1, \dots that is increasing in each coordinate, i.e., for all $j \in R$, $s_j^0 \leq s_j^1 \leq \dots$.*

PROOF. To see that s_j^t exists for all t and j , first note that when $q_j(s^{t-1}) \geq p_j(1 - \epsilon)$, it follows trivially. So consider the case that $q_j(s^{t-1}) < p_j(1 - \epsilon)$. We need to show that the equation

$$q_j(s_1^{t-1}, \dots, s_{j-1}^{t-1}, x, s_{j+1}^{t-1}, \dots, s_n^{t-1}) = p_j(1 - \delta)$$

has a solution. For convenience, let

$$Q_j(x) = q_j(s_1^{t-1}, \dots, s_{j-1}^{t-1}, x, s_{j+1}^{t-1}, \dots, s_n^{t-1}).$$

Notice that

$$\lim_{x \rightarrow \infty} Q_j(x) = \sum_{i \in \Gamma_{\text{in}}(j)} p_i.$$

Since p is consistent with G , this sum either surpasses p_j or approaches p_j in the limit. Since $Q_j(s_j^{t-1}) < p_j(1 - \epsilon)$, and Q_j is increasing in x and continuous, there is some $x \geq s_j^{t-1}$ for which $Q_j(x) = p_j(1 - \delta)$, as desired. Note that this also shows that the sequence is non-decreasing. \square

Thus, we have a sequence of score functions that is non-decreasing in each coordinate. We will show that each coordinate is upper bounded. At a high level, the proof argues that any node whose score increases beyond bound must already be taking almost all of the probability mass from its in-neighbors (by unboundedness), but that these neighbors must among them have enough probability mass (by consistency), thus reaching a contradiction.

LEMMA 7. *For all $j \in R$ there is an upper bound u_j such that $s_j^t \leq u_j$ for all $t \geq 0$.*

PROOF. Let $Z \subseteq R$ be the set of nodes for which the upper bound exists. We will show that $Z = R$. We begin by showing that Z is always non-empty.

Fix a time t , and let Z' be the set of nodes for which $q_j(s^t) > p_j$. Note that if $q_j(s^t) \leq (1 - \epsilon)p_j$ for some $t = t_0$, then $q_j(s^t) \leq (1 - \delta)p_j$ for all $t \geq t_0$. This follows from the fact that $Q_j(x)$ is non-increasing for all coordinates $k \neq j$. Thus if Z' is empty, then the total probability mass on the nodes in R is:

$$\sum_{j \in R} q_j(s^t) \leq \sum_{j \in R} (1 - \delta)p_j < 1,$$

a contradiction. Therefore Z' and $Z \supseteq Z'$ are non-empty.

Since Z is non-empty, consider what happens when $Z \neq R$. Intuitively, since the scores for these nodes are increasing, it must be the case that their probability mass is below the target. We will again show that not all of the nodes can have their probability mass below the target simultaneously.

Assume without loss of generality that $\min w_{ij} = 1$, and let $W = \max w_{ij}$. Let $s_{\max} = \max_{j \in Z} s_j$ be the maximum score for any node in Z , and denote by

$$M = \frac{s_{\max} n^2 W}{\epsilon p_{\min}},$$

where $p_{\min} = \min_{j \in R} p_j$. Consider the time t when the scores for all nodes in $R \setminus Z$ are at least M . Note that since scores in $R \setminus Z$ tend to infinity such a time must exist.

We partition the nodes in L into three sets, those that are not connected to any node in Z ,

$$A = \{v_i \in L \mid \Gamma_{\text{out}}(v_i) \cap Z = \emptyset\};$$

those connected to nodes in Z and in $R \setminus Z$,

$$B = \{v_i \in L \mid \Gamma_{\text{out}}(v_i) \cap Z \neq \emptyset, \Gamma_{\text{out}}(v_i) \cap R \setminus Z \neq \emptyset\};$$

and those connected only to Z ,

$$C = \{v_i \in L \mid \Gamma_{\text{out}}(v_i) \subseteq Z\}.$$

Since the nodes in $R \setminus Z$ have scores that are increasing, it must be the case that:

$$\sum_{j \in R \setminus Z} q_j(s^t) < (1 - \epsilon) \sum_{j \in R \setminus Z} p_j. \quad (1)$$

Now consider the total probability mass for nodes in $R \setminus Z$

$$\begin{aligned} \sum_{j \in R \setminus Z} q_j(s^t) &= \sum_{i \in A \cup B} p_i \frac{\sum_{j \in \Gamma_{\text{out}}(v_i) \cap R \setminus Z} w_{ij} s_j}{\sum_{j \in \Gamma_{\text{out}}(v_i)} w_{ij} s_j} \\ &= \sum_{i \in A} p_i + \sum_{i \in B} p_i \left(1 - \frac{\sum_{j \in \Gamma_{\text{out}}(v_i) \cap Z} w_{ij} s_j}{\sum_{j \in \Gamma_{\text{out}}(v_i)} w_{ij} s_j} \right) \\ &> \sum_{i \in A} p_i + \sum_{i \in B} p_i \left(1 - \frac{\sum_{j \in \Gamma_{\text{out}}(v_i) \cap Z} W s_{\max}}{M} \right) \\ &\geq \sum_{i \in A} p_i + \sum_{i \in B} p_i \left(1 - n W s_{\max} \cdot \frac{\epsilon p_{\min}}{n^2 W s_{\max}} \right) \\ &\geq \sum_{i \in A \cup B} p_i - \epsilon p_{\min}. \end{aligned} \quad (2)$$

Finally, consider the total probability mass for nodes in Z . Intuitively the nodes in Z inherit all of the probability mass in from nodes in C , and some of the probability mass from

those in B ; formally, since p is consistent with G , we have

$$\sum_{j \in Z} p_j \geq \sum_{i \in C} p_i. \quad (3)$$

Combining (1) and (2), we have

$$(1 - \epsilon) \sum_{j \in R \setminus Z} p_j > \sum_{i \in A \cup B} p_i - \epsilon p_{\min}.$$

Since by consistency

$$\sum_{j \in R} p_j = \sum_{i \in A \cup B \cup C} p_i = \frac{1}{2},$$

this is equivalent to

$$\left(\frac{1}{2} - \sum_{j \in Z} p_j \right) - \epsilon \sum_{j \in R \setminus Z} p_j > \left(\frac{1}{2} - \sum_{i \in C} p_i \right) - \epsilon p_{\min}.$$

Rewriting, we obtain

$$\sum_{i \in C} p_i + \epsilon \left(p_{\min} - \sum_{j \in R \setminus Z} p_j \right) > \sum_{j \in Z} p_j.$$

Since $R \setminus Z$ is non-empty, this implies

$$\sum_{i \in C} p_i > \sum_{j \in Z} p_j,$$

which directly contradicts (3). \square

4.4 Rate of convergence

In order to analyze the convergence time of the algorithm we give an explicit upper bound on the value of the scores, and then give a lower bound on the amount by which the scores increase in every iteration.

LEMMA 8. Let s^t and s^{t+1} be the score functions at time t and $t+1$ respectively. Then for each $j \in R$ either $s_j^t = s_j^{t+1}$ or

$$\frac{s_j^{t+1}}{s_j^t} \geq \frac{1 - \delta}{1 - \epsilon} > 1.$$

PROOF. If s_j^t increases from time t to $t+1$, then we know that

$$\sum_{i \in \Gamma_{\text{in}}(j)} p_i \frac{s_j^t w_{ij}}{\sum_{j' \in \Gamma_{\text{out}}(i)} s_{j'}^t w_{ij'}} = Q_j(s_j^t) < (1 - \epsilon)p_j,$$

and

$$\sum_{i \in \Gamma_{\text{in}}(j)} p_i \frac{s_j^{t+1} w_{ij}}{\sum_{j' \in \Gamma_{\text{out}}(i)} s_{j'}^{t+1} w_{ij'}} = Q_j(s_j^{t+1}) = (1 - \delta)p_j,$$

i.e.,

$$\frac{Q_j(s_j^{t+1})}{Q_j(s_j^t)} \geq \frac{1 - \delta}{1 - \epsilon}.$$

By an averaging argument, there exists an $i \in \Gamma_{\text{in}}(j)$ such that

$$\left(\frac{s_j^t w_{ij}}{\sum_{j' \in \Gamma_{\text{out}}(i)} s_{j'}^t w_{ij'}} \Big/ \frac{s_j^{t+1} w_{ij}}{\sum_{j' \in \Gamma_{\text{out}}(i)} s_{j'}^{t+1} w_{ij'}} \right) \geq \frac{1 - \delta}{1 - \epsilon}.$$

We will focus on this i . Note that $j \in \Gamma_{\text{out}}(i)$. Hence, the above expression can be written as

$$\left(\frac{s_j^t}{s_j^t + Z} \Big/ \frac{s_j^{t+1}}{s_j^{t+1} + Z} \right) \geq \frac{1 - \delta}{1 - \epsilon},$$

where $Z = (1/w_{ij}) \cdot \sum_{j' \in \Gamma_{\text{out}}(i) \setminus \{j\}} s_{j'}^t w_{ij'} \geq 0$ and we used the fact that $s_{j'}^{t+1} = s_{j'}^t$ for $j' \in \Gamma_{\text{out}}(i) \setminus \{j\}$. We now claim the following: for $z \geq 0$ and $\rho > 1$,

$$\frac{a}{a+z} \Big/ \frac{b}{b+z} \geq \rho \implies \frac{a}{b} \geq \rho.$$

Given this claim, the proof is complete by setting $z = Z$, $a = s_j^t$, $b = s_j^{t+1}$, and $\rho = (1 - \delta)/(1 - \epsilon)$.

The claim is easy to see: indeed, after algebraic manipulations, it is equivalent to

$$az \geq \rho bz + ab(\rho - 1).$$

Using $\rho > 1$ and $z \geq 0$, we see that $a \geq \rho b$ and the claim is hence proved. \square

For the explicit upper bound, let $M_1 = 1$ and for $i > 1$, $M_k = \left(\frac{n^2 W}{\epsilon p_{\min}} \right)^{k-1}$. We will show that any point of the algorithm at most $n - k$ nodes can have value of M_k .

LEMMA 9. *At any time t , for all $k \in \{1, \dots, n\}$ and all nodes $j \in R$:*

$$|\{s_j \geq M_k\}| \leq n - k.$$

PROOF. We prove the lemma by induction. The scores are initialized so that $s_i = p_i$. As we saw in the proof of Lemma 6 at least one of the initial scores is such that $Q_j(s_i) \geq (1 - \epsilon)p_i$ and it never increases. Since $p_i < 1$ the statement holds.

Now suppose the statement holds for all $k^* < k$. Let Z be the set of nodes with scores less than M_{k-1} . Lemma 6 states that not all of the scores in $R \setminus Z$ can be larger than $M_{k-1} \cdot \frac{n^2 W}{\epsilon p_{\min}}$. Therefore at least one of the nodes in $R \setminus Z$ will have a score bounded by $M_{k-1} \cdot \frac{n^2 W}{\epsilon p_{\min}} = M_k$, and the proof follows. \square

THEOREM 10. *Let $\delta = \epsilon/2$. Then the algorithm terminates after at most*

$$O\left(\frac{n^2}{\epsilon} \log \frac{nW}{\epsilon p_{\min}}\right)$$

iterations.

PROOF. From Lemma 8, in every iteration at least one score increases by a factor of

$$\frac{1 - \delta}{1 - \epsilon} \geq 1 + \frac{\epsilon}{2}.$$

Since each score is bounded by M_n , the number of times each score can increase is bounded by:

$$\log_{1+\epsilon/2} M_n = O\left(n \log_{1+\epsilon/2} \left(\frac{n^2 W}{\epsilon p_{\min}}\right)\right) = O\left(\frac{n}{\epsilon} \log \frac{nW}{\epsilon p_{\min}}\right).$$

Summing over all nodes completes the proof. \square

4.5 Uniqueness

We show that the solution to the steady-state inversion problem is unique up to normalization. More formally,

THEOREM 11. *For any two solutions s and s' to the steady-state inversion problem, there exists a constant $c > 0$ such that $s = c \cdot s'$ element-wise.*

PROOF. Consider the solution s' and normalize it so that s and s' agree on some node v . Let $Z = \{i \mid s'_i > s_i\}$. Without loss of generality $Z \neq \emptyset$, since otherwise we can swap the roles of s and s' .

Consider the nodes in $A = \Gamma_{\text{in}}(Z) \setminus Z$. Note that $A \neq \emptyset$ since the graph is strongly connected and $v \notin Z$. Then,

$$\begin{aligned} \sum_{i \in Z} p_i &= \sum_{j \in A} p_j \frac{s_i w_{ji}}{\sum_{i' \in \Gamma_{\text{out}}(j)} s_{i'} w_{ji'}} \\ &> \sum_{i \in A} p_j \frac{s'_i w_{ji}}{\sum_{i' \in \Gamma_{\text{out}}(j)} s'_{i'} w_{ji'}} \\ &= \sum_{i \in Z} p_i. \end{aligned}$$

Here the inequality follows from the fact that $\frac{x}{x+y}$ is increasing in x and decreasing in y , and the last equality follows from s' being a solution. \square

4.6 Generalizations

We now state a generalization of our result on the steady-state inversion problem. As before, let $G = (V, E)$ be an instance of the steady-state inversion problem and let $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ be a function. We can define the transition matrix $T_{G,s}^f$ with respect to f and a score function s as

$$T_{G,s}^f(i, j) = \frac{f(w_{ij}, s_j)}{\sum_{j' \in \Gamma_{\text{out}}(i)} f(w_{ij'}, s_{j'})},$$

i.e., f is applied to the edge weight and the score function. Let $\pi_{G,s}^f$ be the stationary distribution of the Markov chain. Note that the problem considered in Section 4.1 corresponds to $f(x, y) = x \cdot y$.

We now state the sufficient conditions on f for the feasibility of the corresponding steady-state inversion problem, i.e., given p , does there exist a score function s such that $\pi_{G,s}^f = p$. We state the result for the bipartite case as in Section 4.2.

THEOREM 12. *Given a graph G and a probability distribution p , and a function f , the steady-state inversion problem with f is feasible if p is consistent with G and f satisfies the following two properties: (i) $f(x, y)$ is continuous and increasing in y and (ii) $\lim_{y \rightarrow \infty} f(\cdot, y) = \infty$.*

We omit the details of this proof in this version.

4.7 Connections to choice theory

The steady-state inversion problem is somewhat related to the classical discrete choice problem. We recall the *Choice Axiom* formulated by Luce in his seminal work [23, 24]. The axiom states that when faced with a choice between two objects, the relative probability of selecting one object over another is independent of the presence of other alternatives. An equivalent formulation states that each object has a latent score s_i and objects are selected with probability proportional to their score.

Let U be the universe of objects. An instance of the choice problem is of a collection of m observations, O_1, \dots, O_m where each observation O_i is a tuple containing a set $S_i \subseteq U$ of objects a user was choosing between as well as the object $\text{sel}(S_i) \in S_i$ that was selected. The goal is then to find scores that maximize the likelihood of the data, where the likelihood of the data is:

$$\mathcal{L} = \prod_{i \in m} \frac{s_{\text{sel}(S_i)}}{\sum_{u_k \in S_i} s_k}.$$

We can model this as an instance of the bipartite steady-state inversion problem. Create one node v_i for each observation O_i , and one node u_j for each item in the universe U . Let $V = \cup\{v_i\}$. We create a bipartite graph $G = (V, U, E)$ with the following edges. For each observation O_i , set $\Gamma_{\text{out}}(v_i) = S_i$. Finally, for each node $u_j \in U$, set

$$p_j = \frac{1}{m} \cdot |\{i \mid \text{sel}(S_i) = u_j\}|,$$

i.e., the number of times item u_j was selected, scaled appropriately.

THEOREM 13. *The solution to the steady-state inversion problem defined above maximizes the likelihood in the choice problem.*

PROOF. First observe that the instance of the bipartite steady-state inversion problem is consistent. For each observation O_i , let $\alpha_{ij} = 1/m$ if $j = \text{sel}(S_i)$ and 0 otherwise. By construction this setting of α 's satisfies the consistency property.

Let s^* be the solution to the steady-state inversion problem. For every item $u_j \in U$ we have:

$$\frac{1}{m} \cdot |\{i \mid \text{sel}(S_i) = u_j\}| = \sum_{i: u_j \in S_i} \frac{1}{m} \cdot \frac{s_j}{\sum_{u_k \in S_i} s_k}. \quad (4)$$

On the other hand, the log-likelihood of the data in the choice problem is:

$$\log \mathcal{L} = \sum_{i=1}^m \left(\log s_{\text{sel}(S_i)} - \log \sum_{u_k \in S_i} s_k \right).$$

Optimality conditions imply that for any element u_j :

$$\begin{aligned} 0 &= \frac{\partial \log \mathcal{L}}{\partial s_j} \\ &= \sum_{i \mid \text{sel}(S_i) = u_j} \frac{1}{s_j} - \sum_{i \mid u_j \in S_i} \frac{1}{\sum_{u_k \in S_i} s_k} \\ &= |\{i \mid \text{sel}(S_i) = u_j\}| - \sum_{i \mid u_j \in S_i} \frac{s_j}{\sum_{u_k \in S_i} s_k}, \end{aligned}$$

which is exactly the condition in (4). Since we know the solution to the steady-state inversion problem is unique (Theorem 11), the proof follows. \square

The connection to choice theory also explains the consistency condition (Definition 4) required for the feasibility of the bipartite steady-state inversion problem. Below we show that the consistency condition is satisfied if and only if there exists a set of individual choices that lead to the observed stationary distribution.

THEOREM 14. *Let $G = (L, R, E)$ and a distribution p be an instance of the bipartite steady-state inversion problem.*

Then p is consistent with G if and only if there is an equivalent discrete choice problem. Moreover, such a discrete choice problem can be found in polynomial time.

PROOF. Suppose that p is consistent with G . Let α_{ij} be the rational weights on the edges that satisfy consistency. Let

$$N = \left(\prod_{(i,j) \in E} \alpha_{ij} \right)^{-1}.$$

We will create a discrete choice problem with N observations. For each node $i \in L$ create $1/p_i$ observations, each with the identical choice set $\text{off}_{\text{out}}(i)$. For each $j \in \Gamma_{\text{out}}(i)$, designate u_j as selected $1/\alpha_{ij}$ times. Note that by the choice of N all of these are integer quantities. It is easy to see that this choice problem is equivalent to the steady-state inversion problem.

If G and p are consistent, then one can find the α_{ij} with a single round of maximum flow. Given the bipartite graph G create a source node s , connected to each node $v_i \in L$ with an edge of capacity p_i . In addition, create a sink node t , connected to each node $v_j \in R$ with an edge of capacity p_j . Compute the maximum flow on the resulting graph. If the value of the max flow is below $1/2$ then p is not consistent for G . If the value of the flow is $1/2$, then setting α_{ij} to the flow on each edge (i, j) will satisfy the consistency constraints. \square

5. EXPERIMENTS

The solution to the steady-state inversion problem together with the modeling assumptions made by choice theory allow us to predict the number of times each particular choice was made by observing only the final outcomes of the decision process. For example, given the relative popularity of web pages, we can infer the number of times users followed each particular link. Our main experiment will show that by only considering the *structure* of the graph, that is the underlying choice sets, we can reduce the error by almost a factor of 2 compared to raw popularity based predictions. In all of the experiments we set $\epsilon = 10^{-5}$; we will refer to this setting of the scores as **Steady-State Inversion**. We also study the performance of this algorithm, especially its convergence and running time properties.

5.1 Data

We consider the following four datasets for our evaluation. The datasets span different genres and cover different online user experiences. Three of the datasets are publicly available for repeatability purposes.

Wikipedia paths.

WIKI is a set of navigation paths through Wikipedia, collected as part of the Wikipedia human computation game [32, 33]. In the game subjects were asked to navigate from a starting Wikipedia page to a particular target only by following links in the documents. The dataset consists of 51,318 (source, destination) paths which we then decompose into 199,945 transition pairs. The dataset is publicly available from snap.stanford.edu/data/wikispeedia.html.

We let the subgraph induced by the transition pairs denote the choice sets available at each web page, and set the desired stationary distribution probability to be proportional to the

number of times the node appears in the dataset. Our goal is then to predict the number of transitions along every link in the graph.

Restaurant choice.

REST captures the decisions made by a user as a result of a broad restaurant search query issued to Google. For each query we look at the restaurants offered in the search results and note which of the choices was first clicked. Each result displays some metadata about the restaurant such as the star rating, type of cuisine etc. It is important to note that unlike traditional search results, the UI treatment displays all of the options in a horizontal panel, thereby reducing position effects.

We take a random sample of 100,000 queries, which had 66,464 unique selection choices. Our goal is to predict the restaurant that will be clicked by the user in each of the queries.

Entree recommendations.

ENTREE is a public log of the interactions with the Entree Chicago restaurant recommendation system (available at kdd.ics.uci.edu/databases/entree/entree.html as part of the UCI repository [2]). The data consists of 50,672 user sessions, with each session consisting of restaurants explored and the final restaurant chosen by the user (the End point). We treat the explored restaurants as a choice set and the final restaurant as the action. There are 27,668 unique choice sets, and 1,066 total restaurants selected.

Comedy.

COMEDY is the result of YouTube Comedy Slam, a video discovery experiment (available at archive.ics.uci.edu/ml/datasets/YouTube+Comedy+Slam+Preference+Data, also from [2]) that ran on YouTube in 2011 and 2012. In the experiment users were shown a pair of videos in random order and were asked to vote for the video that was funnier. We use this as a binary choice dataset, with 225,593 total choices over 75,447 unique choice sets.

5.2 Baselines and metrics

We compare our algorithm to the following four baselines. Each of these baselines will compute a score for each node in the graph.

- **Uniform:** The score of each node is identical, simulating a uniform random walk on the graph.
- **Popularity:** The score of each node is set to the popularity of the node. This captures the notion that popular items should have higher scores, but is agnostic to the structure of the graph, and thus cannot differentiate between an item being popular because it is present in many choice sets (and thus has many chances of being selected), or popular because it has very high quality (and thus is selected almost every time it is present).
- **PageRank:** The score of each node is set to the PageRank [7] of the node.
- **Temperature Rank:** The score of each node is set to the Temperature Rank [31] of the node.

We will use two metrics to explore our results. The first is the RMSE of predicting the number of transitions along each link. The second is the log-likelihood of the data using the particular set of scores under the choice model. To be able to interpret the numbers across the four datasets, we scale them so that the numbers for **Popularity** are always unit.

5.3 Predicting the transitions

We show the RMSE of the transition prediction tasks for both datasets in Table 1, where we normalize the results to have the **Popularity** RMSE = 1.0. Note that lower values mean better prediction.

For all datasets our algorithm outperforms the four baselines, reducing the error by 10–30% over the best available baseline. Note that on the WIKI dataset all of the other baselines outperform the **Popularity** baseline. Nonetheless, the **Steady-State Inversion** solution is better still. On the other hand, for restaurant choice, **Popularity** is the strongest baseline, but **Steady-State Inversion** beats it by an additional 10%. Note again that in all of the prediction tasks we do not take *any* metadata into account; these predictions are based solely on the structure of the graph.

A trend nearly identical to that in Table 1 is obtained for the log-likelihood metric as well. For sake of brevity, we do not repeat the numbers.

5.4 Rate of convergence

The algorithm we presented is guaranteed to converge to the optimal set of scores. Below we show that the upper bound on the convergence derived in Theorem 10 is incredibly pessimistic and in practice, the algorithm reaches a good solution after fewer than ten iterations, and converges soon after.

In Figures 2 and 3 we plot the log-likelihood as well as the RMSE as the function of the number of iterations on the WIKI and REST datasets. Again we normalize the likelihood so that it is -1 for the **Popularity** baseline. By design, the algorithm begins with the popularity baseline. It is evident that the biggest gains in RMSE and likelihood occur during the first few iterations, after that the algorithm quickly hits diminishing returns and terminates after 32 iterations for WIKI and 9 iterations for REST. Similar trends are observed for the other two datasets and we do not repeat the corresponding plots.

6. CONCLUSIONS

In this paper, we studied the following setting. Users navigate between states according to a Markov process. We are able to observe the steady state of this process, and we wish to learn the transition matrix. Even though this problem is underconstrained in general (as the steady state has linear number of variables, while the matrix can have quadratically many variables), we used a specific form of the transition matrix: each state of the Markov process has an unknown score and the probability of transition from a source state to a destination state is dependent on the score of the destination state. We showed a natural sufficient condition on the feasibility of the problem. For the instances that are feasible we gave a simple iterative algorithm to solve this inverse problem; we also showed the polynomial rate of convergence of this algorithm, and proved the solution unique. We then applied this algorithm to a few natural settings and showed

	Popularity	Uniform	PageRank	Temperature Rank	Steady-State Inversion
WIKI	1.0	0.65	0.83	0.65	0.57
REST	1.0	1.17	1.39	1.21	0.89
ENTREE	1.0	0.69	1.01	0.56	0.42
COMEDY	1.0	0.65	0.90	0.78	0.36

Table 1: The RMSE of the transition prediction task normalized by the RMSE obtained by the Popularity baseline.

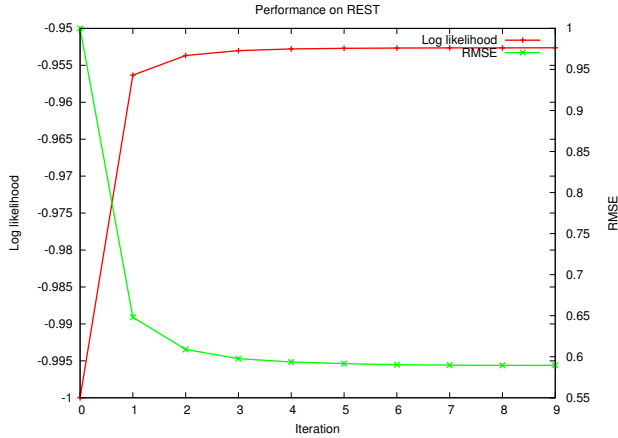


Figure 3: The log-likelihood and RMSE as a function of the number of iterations on the REST dataset.

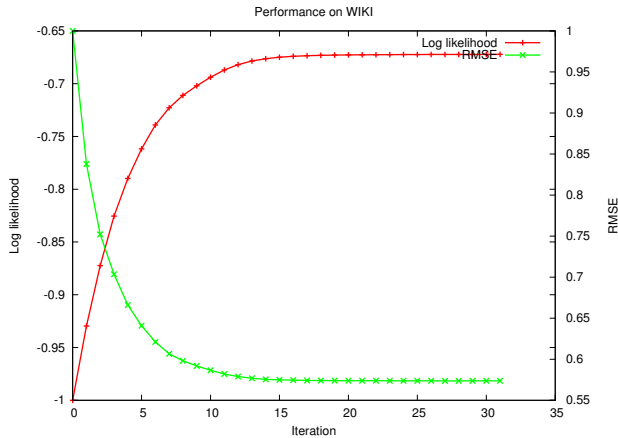


Figure 2: The log-likelihood and RMSE as a function of the number of iterations on the WIKI dataset.

that this algorithm has a better likelihood of predicting transitions in user behavior than other strong baselines.

Our work throws open a number of research questions. Even though our algorithm admits an obvious parallelization and is reasonably efficient in practice, making it scale to hundreds of millions of nodes would be very useful. It will also be interesting to study our model with a random teleportation component, as in PageRank. Finally, using our framework for other application domains including opinion ratings seems fruitful.

Acknowledgments

We thank Bill Heavlin and the reviewers for many useful pointers and suggestions.

7. REFERENCES

- [1] K. Avrachenkov, D. Nemirowsky, and K. S. Pham. A survey on distributed approaches to graph based reputation measures. In *Performance Evaluation Methodologies and Tools*, pages 1–9, 2007.
- [2] K. Bache and M. Lichman. UCI Machine Learning Repository, 2013.
- [3] M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, 1985.
- [4] P. Berkhin. A survey on PageRank computing. *Internet Mathematics*, 2(1):73–120, 2005.
- [5] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: Algorithms, theory, and experiments. *ACM Trans. Internet Technol.*, 5(1):231–297, 2005.
- [6] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46:667–689, 2004.
- [7] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 56(18):3825–3833, 2012.
- [8] Y.-Y. Chen, Q. Gan, and T. Suel. Local methods for estimating PageRank values. In *CIKM*, pages 381–389, 2004.
- [9] S. Chib and E. Greenberg. Understanding the Metropolis–Hastings algorithm. *American Statistician*, 49(4):327–335, 1995.
- [10] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM Review*, 50:37–66, 2008.
- [11] P. Hall. On representatives of subsets. *J. London Math. Soc.*, 10(1):26–30, 1935.
- [12] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [13] T. H. Haveliwala. Topic-sensitive PageRank. In *WWW*, pages 517–526, 2002.
- [14] D. Hensher, J. Rose, and W. Greene. *Applied Choice Analysis: A Primer*. Cambridge University Press, 2005.
- [15] S. Kamvar, T. Haveliwala, and G. Golub. Adaptive methods for the computation of PageRank. *Linear Algebra and its Applications*, pages 51–65, 2004.
- [16] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Extrapolation methods for accelerating PageRank computations. In *WWW*, pages 261–270, 2003.

- [17] J. G. Kemeny and J. L. Snell. *Finite Markov Chains (with a New Appendix "Generalization of a Fundamental Matrix")*. Springer, 1983.
- [18] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
- [19] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004.
- [20] A. N. Langville and C. D. Meyer. Updating PageRank with iterative aggregation. In *WWW (Posters)*, pages 392–393, 2004.
- [21] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.
- [22] N. Litvak, W. R. W. Scheinhardt, and Y. Volkovich. In-degree and PageRank: Why do they follow similar power laws? *Internet Math.*, 4(2-3):175–198, 2007.
- [23] R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.
- [24] R. D. Luce. The choice axiom after twenty years. *Journal of Mathematical Psychology*, 15(3):215–233, 1977.
- [25] F. McSherry. A uniform approach to accelerated PageRank computation. In *WWW*, pages 575–582, 2005.
- [26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [27] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *SIGIR*, pages 258–266, 2001.
- [28] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, InfoLab, Stanford University, 1999.
- [29] R. B. Potts and R. M. Oliver. *Flows in Transportation Networks*. Academic Press, 1972.
- [30] D. Rafiei and A. O. Mendelzon. What is this page known for? Computing Web page reputations. In *WWW*, pages 823–835, 2000.
- [31] J. A. Tomlin. A new paradigm for ranking pages on the world wide Web. In *WWW*, pages 350–355, 2003.
- [32] R. West and J. Leskovec. Human wayfinding in information networks. In *WWW*, pages 619–628, 2012.
- [33] R. West, J. Pineau, and D. Precup. Wikispeedia: An online game for inferring semantic distances between concepts. In *IJCAI*, pages 1598–1603, 2009.