MANIPULATION OF PAGERANK
AND
COLLECTIVE HIDDEN MARKOV MODELS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Daniel Sheldon

Februrary 2010

© 2010 Daniel Sheldon

ALL RIGHTS RESERVED

MANIPULATION OF PAGERANK

AND

COLLECTIVE HIDDEN MARKOV MODELS

Daniel Sheldon, Ph.D.

Cornell University 2010

The first part of this thesis explores issues surrounding the manipulation of PageRank, a popular link-analysis based reputation system for the web. PageRank is an essential part of web search, but it is also subject to manipulation by selfish web authors. We develop an alternative to PageRank, based on expected hitting time in a random walk, that is provably robust to manipulation by outlinks. We then study the effects of manipulation on the network itself by analyzing the stable outcomes of the *PageRank Game*, a network-formation model where web pages place outlinks strategically to maximize PageRank.

The second part of the thesis explores probabilistic inference algorithms for a family of models called *collective* hidden Markov models. These generalize hidden Markov models (HMMs) to the situation in which one views partial information about many indistinguishable objects that all behave according to the same Markovian dynamics. Collective HMMs are motivated by an important problem in ecology: inferring bird migration paths from a large database of observations.

BIOGRAPHICAL SKETCH

Daniel Sheldon received a B.A. in Mathematics from Dartmouth College in 1999. Before enrolling at Cornell University in 2004, he worked at Akamai Technologies followed by DataPower Technology, two internet startup companies based in Cambridge, MA.

Dedicated to my wife, Heather, with love.

ACKNOWLEDGEMENTS

I gratefully acknowledge the help I have received along the way from the students, staff and faculty of the Cornell University Department of Computer Science, and, in particular, my advisor John Hopcroft.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Algorithmic Wide-Angle Lens

This is a thesis in two parts. Chapters 2 and 3 explore issues surrounding manipulation of PageRank, and Chapter 4 explores inference algorithms for *collective hidden Markov models*, a class of probabilistic models whose definition was motivated by a data-centric study of bird migration. The most obvious common thread connecting the two investigations is the suite of algorithmic tools employed: probability, statistics, networks and discrete Markov chains. However, the two parts also enjoy a more subtle, and perhaps stronger, thematic connection, one that is emblematic of computer science in the information age. We call it the "algorithmic wide-angle lens".

In 2007, in a presentation to the Federated Computing Research Conference, Christos Papadimitriou described a transformative export from computational research to the sciences he called the "algorithmic lens" [52]. He was referring not only to the export of analytical tools, but, more importantly, to a new algorithmic world view being adopted by researchers in diverse fields such as mathematics, natural sciences, economics and sociology. The algorithmic *wide-angle* lens extends this metaphor to describe the role of algorithms in understanding certain phenomena, from either the natural or digital world, that are so broad in scope that they can *only* be viewed algorithmically. The link structure of the web and the migration of wild birds are two such phenomena.

Humans interact with the web daily, yet no one can answer simple

questions about global web structure — "How can one navigate from `www.cs.cornell.edu` to `google.com` in the fewest number of clicks?", "How many pages are on the web?", "What is the average number of hyperlinks per web page?" — without the assistance of algorithms. Similarly, almost every person has observed bird migration: perhaps they noted the seasonal appearance or disappearance of birds in their neighborhood, or glanced to the sky in the fall to see a flock of geese migrating south for the winter. However, the continental phenomenon of bird migration occurs on too broad a scale and is far too complex for any individual to grasp by his or her own observations. It is only by assembling observations from many individuals over a broad geographic range that a large scale picture of migration can emerge. With the help of digital information and the algorithms to interpret it, a *quantitative* picture can emerge.

The remainder of this chapter introduces and motivates the work on manipulation of PageRank that appears in Chapters 2 and 3. Chapter 4 is a self-contained presentation of collective hidden Markov models for modeling bird migration.

1.2 Manipulation of PageRank

Reputation and ranking systems are an essential part of web search and e-commerce. The general idea is that the reputation of one participant is determined by the endorsements of others; for example, one web page endorses another by linking to it. However, not all participants are honorable, so manipulation-resistance is an important consideration.

Link analysis algorithms for web search first appeared in the late 1990s. Notable examples are HITS [44] and PageRank [7]; these algorithms use the global hyperlink structure of the web to determine the importance of individual pages. Each uses notions of endorsements among pages, so we also refer to them as link-based reputation systems. Our work focuses on PageRank, an influential link-analysis algorithm developed by Sergey Brin and Lawrence Page to rank web pages in the original Google search engine [7]. PageRank is a success story: it is still today an important ingredient in search technology, and it has been exported to many other domains.

However, because PageRank is so successful, it is also a popular target for manipulation in the form of *link spam*, where web authors manipulate hyperlinks to boost their own reputation. There is a strong economic incentive for pages to rank highly in search results, and PageRank is a well-known and simple-to-understand part of a search engine. Moreover, despite the global nature of the PageRank algorithm, it is relatively easy for a page to boost its PageRank by changing *only outlinks*. This is an undesirable property in a repuation system: one would like the reputation of a page to determined by the actions of *other* participants.

1.2.1 A Short Description of PageRank

It is instructive to take a closer look at the PageRank to see where things go wrong. Let $G = (V, E)$ be a directed graph (e.g, the web). PageRank assigns the score $\pi(v)$ to node v, where π is defined to be the stationary distribution of a random walk on G, giving the pleasing interpretation that the score of page v

is the fraction of time a web surfer spends there if she randomly follows links forever. For technical reasons, the random walk is modified to restart in each step with probability α, jumping to a page chosen at random, either uniformly or from a pre-specified distribution. Informally, this allows the random walk to escape "dangling" portions of the web graph that do not connect back to main portion. Mathematically, it ensures that π exists, is unique, and efficient to compute. A well-known fact about Markov chains is that the expected *return time* of v — the number of steps it takes a random walk starting at v to return to v — is equal to $1/\pi(v)$ [1]. A heuristic argument for this equivalence is that a walk that returns to v every r steps on average should spend $1/r$ of all time steps there.

1.2.2 Manipulation by Outlinks

The intuition of the manipulation strategy follows easily from the interpretation of PageRank as the inverse of expected return time. If node v wishes to minimize the expected return time, it should link only to nodes from which a random walk will return to v very quickly, in expectation. By partnering with just one other node to form a 2-cycle with no other outlinks, v ensures a return in two steps — the minimum possible assuming self-loops are ignored — unless the walk jumps first. In this fashion, v can often boost its PageRank by a factor of 3 to 4 for typical settings of α [17]. We will refer to this as the *short-cycle* manipulation strategy.

1.2.3 Contributions

This thesis addresses two issues related to outlink manipulation in PageRank. In Chapter 2, we develop a reputation system that is nearly identical to PageRank, but provably robust to the type of manipulation described above: no page can increase its reputation by changing outlinks alone. The result is based on the following observation: to succeed in the short-cycle manipulation strategy, a node must increase the probability that the walk will return before the first random jump. Indeed, the jump destination is independent of v's outlinks, and return time is determined once the walk reaches v again, so v's outlinks have no further impact once a jump has occurred. This suggests an alternative experiment to determine the reputation of v: measure the expected time to hit v *following* the first random jump. This quantity is called the expected *hitting time* of v from the restart distribution. We will prove a precise relationship between expected hitting time and PageRank, and show that a reputation system based on hitting time is provably resistant to manipulation by placement of outlinks.

In Chapter 3, we address the following question:

How does selfish behavior induced by PageRank affect the structure of the underlying network?

To answer this question, we introduce a network formation game called the PageRank Game. The players in the game are the nodes of a directed graph, and they place outlinks selfishly, attempting to maximize PageRank. We analyze the outcomes of the game to understand the influence that PageRank has on network structure. Our theoretical analysis of stable outcomes (Nash equilibria) reveals a rich set of equilibria with surprising connections to several classes

of graphs that possess strong symmetry or regularity properties. Thus, many complex and highly connected structures are *possible* outcomes of the game. However, our study of best response dynamics, a realistic model of sequential play, reveals that *typical* outcomes are highly fragmented, consisting of many singletons, two-cycles and other small components.

Our main qualitative finding in Chapter 3 is that strategic behavior in the PageRank game is destructive — PageRank incentivizes the removal of edges and leads toward fragmentation of the network. Indeed, in the short-cycle manipulation strategy, node v places outlinks only in its immediate neighborhood, attempting to ensure a quick return of the random walk. If many nodes behave this way, small tightly connected neighborhoods emerge from which it is difficult to escape. In fact, the situation is more extreme: a typical best strategy for v is to delete all of its links and place a single link to one of its in-neighbors. This, in turn, gives the neighbor very strong incentive to link only to v, resulting in "mutual admiration" dynamics that lead to many disconnected two-cycles.

1.2.4 Related Work

Since the introduction of PageRank by Brin and Page [7], it has been adapted to a variety of applications, including personalized web search [51], web spam detection [35], and trust systems in peer-to-peer networks [42]. Each of these uses the same general formulation and our work applies to all of them.

Much work has focused on the PageRank system itself, studying computation methods, convergence properties, stability and sensitivity, and, of course, implementation techniques. The survey by Langville and Meyer is an excellent

summary of this wide body of work [46]. Computationally, the Monte Carlo methods of Fogaras and Rácz [25] and Avrachenkov et al. [3] are similar to our algorithms for hitting time in Chapter 2. They use a probabilistic formulation of PageRank in terms of a *short* random walk that permits efficient sampling. In particular, we will use the same idea as [25] to efficiently perform many random walks simultaneously in a massive graph, with only sequential access to the list of edges.

Link spam and manipulability of PageRank have been studied in several contexts [4, 17, 33, 34]. Several methods have been proposed to detect link spam [32, 48], and to develop reputation systems, often based on principles similar to PageRank, that are robust to manipulation [26, 38, 66].

For a more general treatment of reputation systems in the presence of strategic agents, Friedman et al. provide an excellent overview with specific results from the literature [26]. Cheng and Friedman [16] prove an impossibility result that relates to our work in Chapter 2 — a wide class of reputation systems (including expected hitting time) cannot be resistant to a particular attack called the *sybil attack* [22]. We will discuss sybils and our work in the context of this result in Section 2.3.3 of Chapter 2.

Hitting time is a classical quantity of interest in Markov chains. See Chapter 2 of the monograph by Aldous and Fill for an overview [1]. The exact terminology and definitions vary slightly: we will define hitting time as a random variable, but sometimes it is defined as the expectation of the same random variable. Also, the term *first passage time* is sometimes used synonymously. In a context similar to our research, hitting time was used as a measure of proximity between nodes to predict link formation in a social network [47]; also, the node

similarity measure in [40] can be formulated in terms of hitting time.

In Section 2.2 of Chapter 2, we prove a quantitative relationship between expected hitting time and PageRank; the result relies on the fact that the PageRank random walk periodically restarts by jumping to a random node. This analysis is related to a class of models called *regenerative* stochastic processes. In fact, Theorem 1 can be derived as a special case of a general result about regenerative stochastic processes. See Equation (15) in [36] and the references therein for details.

After our initial conference publication on this topic [38], we discovered the paper by Avrachenkov et al. that independently proved one of the results in our paper [2]. In their work on the effect of new links on PageRank, they show that the PageRank of page v can be written as a product of two terms, where only the first term depends on the outlinks of v. The second term in their formulation — which is independent of v's outlinks — is exactly our measure of the reputation of v.

In Chapter 3, we use the PageRank game to analyze strategic link-placement on the web. This is an example of a *network formation game*, in which the nodes of a network are strategic agents that buy edges attempting to optimize a specific objective. Network formation games have been used to model various types of strategic network formation, ranging from communication networks to social networks. Tardos and Wexler [62] and Jackson [39] provide two recent surveys. In some models, the players at the two endpoints of the edge must cooperate to purchase an edge; in others, the decision to place an edge is unilateral. Fabrikant et al. introduced a unilateral model that received significant attention [24]: in their model, either u or v may elect to purchase the undirected edge connect-

ing the two nodes. Our model is also unilateral, but with directed edges: only node u may buy the directed edge (u, v), corresponding to a page u creating a hyperlink to page v.

The network formation game most similar to the PageRank game is that of Rogers in his work on an economic model of social status [57]. Players place *weighted* links to other players, and they earn social status from incoming links, weighted by both the status of the linking player and the weight of the link. A version of the Rogers model is, in fact, identical to PageRank, and some of the analysis is similar to that of the PageRank Game. However, the fact that players may divide their link budget arbitrarily into weighted links is fundamentally different from the PageRank game where probability is divided equally among all outgoing links. In the Rogers model, each player solves a *continuous* optimization problem, while in the PageRank game, that problem is discrete: to whom should I link?

Finally, our characterization of Nash equilibria in the PageRank game touches on some topics in algebraic graph theory. An excellent reference is the book by Godsil [28]. Other useful references are [6, 13, 30]. We define a graph property called *edgewise walk-regularity* to characterize Nash equilibria. This is analogous to two existing concepts: *walk-regularity* [29] and *distance-regularity* [9]. Each of these concepts establishes a combinatorial equivalence among certain pairs of vertices requiring that for each qualifying pair, the number of walks of length t connecting the two is the same, for all t. We discuss these connections in more detail in Chapter 3.

CHAPTER 2

MANIPULATION-RESISTANT REPUTATIONS USING HITTING TIME

2.1 Introduction

PageRank is susceptible to the short-cycle manipulation strategy in which nodes place outlinks to trap the random walk in their local neighborhood and force it to return quickly. In this chapter, we develop a reputation system based on hitting time that is robust to this type of manipulation. The main contributions are:

- In Theorem 1, we develop a precise relationship between PageRank and expected hitting time in a random walk with restart, and show that the expected hitting time of v is equal to $(1-p)/\alpha p$, where p is the probability that v is reached before the first restart. We will adopt p as our measure of the reputation of v.
- We prove that the resulting reputation system resists manipulation, using a natural definition of influence. For example, node v has a limited amount of influence that depends on her reputation, and she may spread that influence using outlinks to increase others' reputations. However, node v cannot alter her own reputation with outlinks, nor can she damage w's reputation by more than her original influence on w. Furthermore, the advantage that v gains by purchasing new nodes, often called *sybils* of v, is limited by the restart probability of the sybils.
- We present an efficient algorithm to simultaneously compute hitting time for all nodes in a large graph. In addition to one PageRank calculation,

our algorithm uses Monte Carlo sampling with running time that is linear in $|V|$ for given accuracy and confidence parameters. This is a significant improvement over traditional algorithms, which require a large-scale computation for each node.[1]

The chapter is structured as follows. In Section 2.2 we present Theorem 1, a characterization of hitting time that is the foundation for the following sections. In Section 2.3 we develop a reputation system using hitting time and show that it is resistant to manipulation. In Section 2.4 we present algorithms for computing hitting time.

2.2 Characterizing Hitting Time

Recall that the PageRank is equal to the inverse of expected return time. This section paves the way toward a reputation system based on hitting time by stating and proving Theorem 1, which characterizes hitting time and relates it to return time, and hence PageRank. Part (i) of the theorem shows that expected hitting time and expected return time are essentially the same *except* for nodes where the random walk is likely to return before jumping, the sign of a known manipulation strategy. Part (ii) proves that the expected hitting time of v is completely determined by the probability that v is reached before the first jump; this will lead to precise notions of manipulation-resistance in section 4.

[1]Standard techniques can simultaneously compute hitting time from all possible sources to a single target node using a system of linear equations. However, what is desired for reputation systems is the hitting time from one source, or in this case a distribution, to all possible targets.

2.2.1 Preliminaries

Let $G = (V, E)$ be a directed graph. Consider the *standard random walk* on G, where the first node is chosen from starting distribution q, then at each step the walk follows an outgoing link from the current node chosen uniformly at random. Let $\{X_t\}_{t \geq 0}$ be the sequence of nodes visited by the walk. Then $\Pr[X_0 = v] = q(v)$, and $\Pr[X_t = v \mid X_{t-1} = u] = 1/\text{outdegree}(u)$ if $(u, v) \in E$, and zero otherwise. Here, we require $\text{outdegree}(u) > 0$.[2] Now, suppose the walk is modified to restart with probability α at each step, meaning the next node is chosen from the starting distribution (henceforth, *restart distribution*) instead of following a link. The new transition probabilities are:

$$\Pr[X_t = v \mid X_{t-1} = u] = \begin{cases} \alpha q(v) + \frac{1-\alpha}{\text{outdegree(u)}} & \text{if } (u, v) \in E \\ \alpha q(v) & \text{otherwise} \end{cases}.$$

We call this the α-*random walk* on G, and we parameterize quantities of interest by the restart probability α. A typical setting is $\alpha = 0.15$, so a jump occurs every $1/.15 \approx 7$ steps in expectation. The *hitting time* of v is $H_\alpha(v) = \min\{t : X_t = v\}$. The *return time* of v is $R_\alpha(v) = \min\{t \geq 1 : X_t = v \mid X_0 = v\}$. When v is understood, we simply write H_α and R_α. We write H and R for the hitting time and return time in a standard random walk.

2.2.2 Theorem 1

Before stating Theorem 1, we make the useful observation that we can split the α-random walk into two independent parts: (1) the portion preceding the first

[2] This is a technical condition that can be resolved in a variety of ways, for example, by adding self-loops to nodes with no outlinks.

jump is the beginning of a standard random walk, and (2) the portion following the first jump is an α-random walk independent of the first portion. The probability that the first jump occurs at time t is $(1-\alpha)^{t-1}\alpha$, i.e., the first jump time J is a geometric random variable with parameter α, independent of the nodes visited by the walk. Then we can model the α-random walk as follows: (1) start a standard random walk, (2) independently choose the first jump time J from a geometric distribution, and (3) at time J begin a new α-random walk. Hence we can express the return time and hitting time of v recursively:

$$R_\alpha = \begin{cases} R & \text{if } R < J \\ J + H'_\alpha & \text{otherwise} \end{cases}, \qquad H_\alpha = \begin{cases} H & \text{if } H < J \\ J + H'_\alpha & \text{otherwise} \end{cases}. \tag{2.1}$$

Here H'_α is an independent copy of H_α. It is convenient to abstract from our specific setting and state Theorem 1 about general random variables of this form.

Theorem 1. *Let R and H be independent, nonnegative, integer-valued random variables, and let J be a geometric random variable with parameter α. Define R_α and H_α as in (2.1). Then,*

(i) $E[R_\alpha] = \Pr[R \geq J]\left(\frac{1}{\alpha} + E[H_\alpha]\right)$,

(ii) $E[H_\alpha] = \frac{1}{\alpha} \cdot \frac{\Pr[H \geq J]}{\Pr[H < J]}$,

(iii) $E[R_\alpha] = \frac{1}{\alpha} \cdot \frac{\Pr[R \geq J]}{\Pr[H < J]}$.

Part (i) relates expected return time to expected hitting time. In the expression describing the relationship, $\Pr[R \geq J]$ is the probability that the walk does not return before jumping. On the web, for example, we expect $\Pr[R \geq J]$ to be close to 1 for most pages, so the two measures are roughly equivalent. However, pages attempting to optimize PageRank can drive $\Pr[R \geq J]$ much lower,

achieving an expected return time that is much lower than expected hitting time.

For parts (ii) and (iii), we adopt the convention that $\Pr[H < J] = 0$ implies $E[H_\alpha] = E[R_\alpha] = \infty$, corresponding to the case when v is not reachable from any node with positive restart probability. To gain some intuition for part (ii) (part (iii) is similar), we can think of the random walk as a sequence of independent explorations from the restart distribution "looking" for node v. Each exploration succeeds in finding v with probability $\Pr[H < J]$, so the expected number of explorations until success is $1/\Pr[H < J]$. The expected number of steps until an exploration is terminated by a jump is $1/\alpha$, so a rough estimate of hitting time is $\frac{1}{\alpha} \cdot \frac{1}{\Pr[H<J]}$. Of course, this is an overestimate because the final exploration is cut short when v is reached, and the expected length of an exploration conditioned on not reaching v is slightly shorter than $1/\alpha$. It turns out that $\Pr[H \geq J]$ is exactly the factor needed to correct the estimate, due to the useful fact about geometric random variables[3] stated in Lemma 1 below. We stress that the expected hitting time of v in the α-random walk is completely determined by $\Pr[H < J]$, the probability that a given exploration succeeds; this will serve as our numeric measure of reputation.

Lemma 1. *Let X and J be independent random variables such that X is nonnegative and integer-valued, and J is a geometric random variable with parameter α. Then $E[\min(X, J)] = \frac{1}{\alpha}\Pr[X \geq J]$.*

Proof of Lemma 1. Recall that J is the time of the first success in a sequence of

[3]We mentioned that Theorem 1 can be derived from a result about regenerative stochastic processes [36]. In fact, Theorem 1 captures most of the generality; to write recurrences as in (2.1), the process need not be Markovian, it is only necessary that the process following a restart is a replica of the original. The only non-general assumption made is that the reset time J is a geometric random variable; this simplifies the conclusions of the theorem.

independent trials that succeed with probability α, so $\Pr[J > t] = (1-\alpha)^t$, and $\Pr[J = t+1] = \alpha \cdot \Pr[J > t] = \alpha(1-\alpha)^t$. The support of J is the set of positive integers. We use these facts to show the result:

$$\begin{aligned}
E[\min(X, J)] &= \sum_{t=0}^{\infty} \Pr[\min(X, J) > t] \\
&= \sum_{t=0}^{\infty} \Pr[X > t] \Pr[J > t] && \text{by independence of } X, J \\
&= \frac{1}{\alpha} \sum_{t=0}^{\infty} \Pr[X > t] \Pr[J = t+1] \\
&= \frac{1}{\alpha} \sum_{t=1}^{\infty} \Pr[J = t] \Pr[X \geq J \mid J = t] \\
&= \frac{1}{\alpha} \Pr[X \geq J]
\end{aligned}$$

□

Proof of Theorem 1. We rewrite $R_\alpha = \min(R, J) + I\{R \geq J\} H'_\alpha$, where $I\{R \geq J\}$ is the indicator variable for the event $[R \geq J]$. Note that $I\{R \geq J\}$ and H'_α are independent. Then, using linearity of expectation and Lemma 1,

$$\begin{aligned}
E[R_\alpha] &= E[\min(R, J)] + \Pr[R \geq J] E[H'_\alpha] \\
&= \frac{1}{\alpha} \Pr[R \geq J] + \Pr[R \geq J] E[H_\alpha] \\
&= \Pr[R \geq J] \left(\frac{1}{\alpha} + E[H_\alpha] \right).
\end{aligned}$$

This proves part (i). The proof of (ii) uses part (i), taking advantage of the fact that the theorem is stated about generic random variables of the form (2.1). Note that the definition of H_α in (2.1) is a special case of the more general form used to define R_α, so we can apply the result from part (i) with H substituted for R.

We get

$$E[H_\alpha] = \Pr[H \geq J]\left(\frac{1}{\alpha} + E[H_\alpha]\right).$$

Solving this expression for $E[H_\alpha]$ gives (ii). Part (iii) is obtained by substituting (ii) into (i). □

2.3 Manipulation-Resistance

In this section we develop a reputation system based on hitting time, and quantify the extent to which an individual can tamper with reputations. It is intuitively clear that node u cannot improve its own hitting time by placing outlinks, but we would also like to limit the damage that u can cause to v's reputation. Specifically, u should only be able to damage v's reputation if u was responsible for v's reputation in the first place. Furthermore, u should not have a great influence on the reputation of too many others. To make these ideas precise, we define reputation using the quantity $\Pr[H < J]$ instead of $E[H_\alpha]$. By Theorem 1, either quantity determines the other — they are roughly inversely proportional — and $\Pr[H < J]$ is convenient for reasoning about manipulation.

Definition 1. *Let rep*$(v) = \Pr[H(v) < J]$ *be the* reputation *of* v.

In words, rep(v) is the probability that a random walk hits v before jumping. Of all walks that reach v before jumping, an attacker u can only manipulate those that hit u first. This leads to our notion of influence.

Definition 2. *Let infl*$(u, v) = \Pr[H(u) < H(v) < J]$ *be the* influence of u on v.

Definition 3. *Let infl*$(u) = \sum_v$ *infl*(u, v) *be the* total influence of u.

When the graph G is not clear from context, we write these quantities as $\Pr_G[\cdot]$, $\text{rep}_G(\cdot)$ and $\text{infl}_G(\cdot,\cdot)$ to be clear. To quantify what can change when u manipulates outlinks, let $\mathcal{N}_u(G)$ be the set of all graphs obtained from G by the addition or deletion of edges originating at u. It is convenient to formalize the intuition that u has no control over the random walk until it hits u for the first time.

Definition 4. *Fix a graph G and node u. We say that an event A is u-*invariant *if $\Pr_G[A] = \Pr_{G'}[A]$ for all $G' \in \mathcal{N}_u(G)$. If A is u-invariant, we also say that the quantity $\Pr[A]$ is u-invariant.*

Lemma 2. *An event A is u-invariant if the occurrence or non-occurrence of A is determined by time $H(u)$.*

Proof of Lemma 2. Let $G' \in \mathcal{N}_u(G)$. It is enough to show that $\Pr_G[A \cap (H(u) = t)] = \Pr_{G'}[A \cap [H(u) = t]]$ for all $t \geq 0$. Let $W_{u,t}$ be the set of all walks that first hit u at step t, that is, let $W_{u,t} = \{w_0 \dots w_t : w_t = u,\ w_i \neq u \text{ for } i < t\}$. For any walk $w = w_0 \dots w_t$, let $\Pr[w]$ be shorthand for the probability of the walk w:

$$\Pr[w] = \Pr[X_0 = w_0] \Pr[X_1 = w_1 \mid X_0 = w_0] \dots \Pr[X_t = w_t \mid X_{t-1} = w_{t-1}].$$

Then, for $w \in W_{u,t}$, the transition probabilities in the expression above are independent of u's outlinks, so $\Pr_G[w] = \Pr_{G'}[w]$. Finally, since A is determined by time $H(u)$, there is a function $I_A : W_{u,t} \to \{0,1\}$ that indicates the occurrence or non-occurrence of A for each $w \in W_{u,t}$. Putting it all together,

$$\begin{aligned}
\Pr_G[A \cap [H(u) = t]] &= \Pr_G[H(u) = t] \Pr_G[A \mid H(u) = t] \\
&= \sum_{w \in W_{u,t}} \Pr_G[w]\, I_A(w) \\
&= \sum_{w \in W_{u,t}} \Pr_{G'}[w]\, I_A(w) \\
&= \Pr_{G'}[A \cap [H(u) = t]]
\end{aligned}$$

□

With the definitions in place, we can quantify how much u can manipulate reputations.

Theorem 2. *For any graph $G = (V, E)$ and $u, v \in V$,*

(i) $\mathit{infl}(u, u) = 0$,

(ii) $\mathit{infl}(u, v) \geq 0$,

(iii) $\mathit{infl}(u, v) \leq \mathit{rep}(u)$,

(iv) $\mathit{infl}(u) \leq \frac{1}{\alpha}\mathit{rep}(u)$.

Let $G' \in \mathcal{N}_u(G)$. Then

(v) $\mathit{rep}_{G'}(v) = \mathit{rep}_G(v) + \mathit{infl}_{G'}(u, v) - \mathit{infl}_G(u, v)$.

Parts (i)-(iv) bound the influence of u in terms of its reputation. Part (v) states that when u modifies outlinks, the change in v's reputation is equal to the change in u's influence on v. Substituting parts (i-iii) into part (v) yields some simple but useful corollaries.

Corollary 1. *Let $G' \in \mathcal{N}_u(G)$. Then*

(i) $\mathit{rep}_{G'}(u) = \mathit{rep}_G(u)$,

(ii) $\mathit{rep}_{G'}(v) \geq \mathit{rep}_G(v) - \mathit{infl}_G(u, v)$,

(iii) $\mathit{rep}_{G'}(v) \leq \mathit{rep}_G(v) - \mathit{infl}_G(u, v) + \mathit{rep}_G(u)$.

No matter what actions u takes, it cannot alter its own reputation (part (i)). Nor can u damage the portion of v's reputation *not* due to u's influence (part (ii)). On the other hand, u may boost its influence on v, but its final influence cannot exceed its reputation (part (iii)).

Proof of Theorem 2. For the most part, these are simple consequences of the definitions. Parts (i) and (ii) are trivial:

$$\text{infl}(u,u) = \Pr\left[H(u) < H(u) < J\right] = 0,$$

$$\text{infl}(u,v) = \Pr\left[H(u) < H(v) < J\right] \geq 0.$$

For part (iii), a walk that hits u then v before jumping contributes equally to u's reputation and u's influence on v:

$$\text{infl}(u,v) = \Pr\left[H(u) < H(v) < J\right] \leq \Pr\left[H(u) < J\right] = \text{rep}(u).$$

Part (iv) uses the observation that not too many nodes can be hit after u but before the first jump. Let $L = |\{v : H(u) < H(v) < J\}|$ be the number of all such nodes. Then,

$$E\left[L\right] = E\left[\sum_v I\{H(u) < H(v) < J\}\right] = \sum_v \Pr\left[H(u) < H(v) < J\right] = \text{infl}(u).$$

But L cannot exceed $J - \min(H(u), J)$, so

$$\begin{aligned}
\text{infl}(u) = E\left[L\right] &\leq E\left[J\right] - E\left[\min(H(u), J)\right] \\
&= E\left[J\right]\left(1 - \Pr\left[H(u) \geq J\right]\right) \qquad \text{(by Lemma 1)} \\
&= E\left[J\right]\Pr\left[H(u) < J\right] \\
&= \frac{1}{\alpha}\text{rep}(u).
\end{aligned}$$

For part (v), we split walks that hit v before jumping into those that hit u first and those that don't:

$$\begin{aligned}\text{rep}_G(v) &= \Pr_G[H(v) < J] \\ &= \Pr_G[H(u) < H(v),\ H(v) < J] + \Pr_G[H(u) \geq H(v),\ H(v) < J] \\ &= \text{infl}_G(u,v) + \Pr_G[H(u) \geq H(v),\ H(v) < J]\end{aligned}$$

The event $[H(u) \geq H(v),\ H(v) < J]$ is determined by time $H(u)$, and hence it is u-invariant. By the preceding calculation, $\Pr[H(u) \geq H(v),\ H(v) < J]$ is equal to $\text{rep}_G(v) - \text{infl}_G(u,v)$, and repeating the calculation for G' gives $\text{rep}_{G'}(v) = \text{infl}_{G'}(u,v) + \text{rep}_G(v) - \text{infl}_G(u,v)$. □

2.3.1 Manipulating the Rankings

The previous results quantify how much node u can manipulate reputation *values*, but often we are more concerned with how much u can manipulate the ranking, specifically, how far u can advance by manipulating outlinks only. The following two corollaries follow easily from Theorem 2 and Corollary 1. Suppose $\text{rep}_G(u) < \text{rep}_G(v)$ and u manipulates outlinks to produce $G' \in \mathcal{N}_u(G)$. We say that u *meets* v if $\text{rep}_{G'}(u) = \text{rep}_{G'}(v)$, and u *surpasses* v if $\text{rep}_{G'}(u) > \text{rep}_{G'}(v)$.

Corollary 2. *Node u cannot surpass a node that is at least twice as reputable.*

Corollary 3. *Node u can meet or surpass at most $\frac{1}{\alpha\gamma}$ nodes that are more reputable than u by a factor of at least $(1+\gamma)$.*

Proof of Corollary 2. Suppose $\text{rep}_G(v) \geq 2 \cdot \text{rep}_G(u)$, then

$$\begin{aligned}\text{rep}_{G'}(v) &\geq \text{rep}_G(v) - \text{infl}_G(u,v) \geq \text{rep}_G(v) - \text{rep}_G(u) \\ &\geq 2 \cdot \text{rep}_G(u) - \text{rep}_G(u) = \text{rep}_G(u) = \text{rep}_{G'}(u).\end{aligned}$$

□

Proof of Corollary 3. Let $A = \{v : \text{rep}_G(v) \geq (1+\gamma)\text{rep}_G(u),\ \text{rep}_{G'}(v) \leq \text{rep}_{G'}(u)\}$ be the set of all nodes with reputation at least $(1+\gamma)$ times the reputation of u that are met or surpassed by u. Then

$$\sum_{v \in A} \text{rep}_G(v) \geq |A|(1+\gamma)\text{rep}_G(u),$$
$$\sum_{v \in A} \text{rep}_{G'}(v) \leq |A|\text{rep}_{G'}(u) = |A|\text{rep}_G(u),$$

so $\sum_{v \in A}(\text{rep}_G(v) - \text{rep}_{G'}(v)) \geq \gamma|A|\text{rep}_G(u)$. But by Corollary 1, $\text{rep}_G(v) - \text{rep}_{G'}(v) \leq \text{infl}_G(u,v)$, so

$$\gamma|A|\text{rep}_G(u) \leq \sum_{v \in A}(\text{rep}_G(v) - \text{rep}_{G'}(v)) \leq \sum_{v \in A} \text{infl}_G(u,v) \leq \text{infl}_G(u) \leq \frac{1}{\alpha}\text{rep}_G(u),$$

hence $|A| \leq \frac{1}{\alpha\gamma}$. □

2.3.2 Reputation and Influence of Sets

We have discussed reputation and influence in terms of individual nodes for ease of exposition, but all of the definitions and results generalize when we consider the reputation and influence of sets of nodes. Let $U, W \subseteq V$, and recall that $H(W) = \min_{w \in W} H(w)$ is the hitting time of the set W. Then we define $\text{rep}(W) = \Pr[H(W) < J]$ to be the reputation of W, we define $\text{infl}(U,W) = \Pr[H(U) < H(W) < J]$ to be the influence of U on W, and we

define $\text{infl}(U) = \sum_{v \in V} \text{infl}(U, \{v\})$ to be the total influence of U. With these definitions, exact analogues of Theorem 2 and its corollaries hold for any $U, W \subseteq V$, with essentially the same proofs. Note that U and W need not be disjoint, in which case it is possible that $H(U) = H(W)$. Further details are omitted.

2.3.3 Sybils

In online environments, it is often easy for a user to create new identities, called *sybils*, and use them to increase her own reputation, even without obtaining any new inlinks from non-sybils. On the web, a spammer might control a large number of sites, arranging them to boost the PageRank of a given target page; such a configuration is called a *spam farm* [34]. In general, a wide class of reputation systems is vulnerable to sybil attacks [16], and, in the extreme, hitting time can be heavily swayed as well. For example, if u places enough sybils so the random walk almost surely starts at a sybil, then adding links from each sybil to u ensures the walk hits u by the second step unless it jumps. In this fashion, u can achieve reputation almost $1 - \alpha$, the probability that the walk does not jump in the first step, and drive the reputation of all non-sybils to zero. We'll see that this is actually the *only* way that sybils can aid u, by gathering restart probability and funneling it towards u. So an application can limit the effect of sybils by limiting the restart probability granted to new nodes. In fact, applications of hitting time analogous to Personalized PageRank [51] and TrustRank [35] are already immune, since they place all of the restart probability on a fixed set of known or trusted nodes. Applications like web search that give equal restart probability to each node are more vulnerable, but in cases like the web the sheer number of nodes requires an attacker to place many sybils to have a substantial

effect. This stands in stark contrast with PageRank, where one sybil is enough to employ the 2-cycle manipulation strategy and increase PageRank by several times [17].

To model the sybil attack, suppose $G' = (V \cup S, E')$ is obtained from G by a sybil attack launched by u. That is, the sybil nodes S are added, and links originating at u or inside S can be set arbitrarily. All other links must not change, with the exception that those originally pointing to u can be directed anywhere within $S \cup \{u\}$. Let q' be the new restart distribution, assuming that q' diverts probability to S but does not redistribute probability within V. Specifically, if $\rho = \sum_{s \in S} q'(s)$ is the restart probability alloted to sybils, we require that $q'(v) = (1-\rho)q(v)$ for all $v \in V$.

Theorem 3. *Let $U = \{u\} \cup S$ be the nodes controlled by the attacker u, and let v be any other node in V. Then*

(i) $\mathit{rep}_{G'}(u) \leq \mathit{rep}_{G'}(U) = (1-\rho)\mathit{rep}_G(u) + \rho$,

(ii) $\mathit{rep}_{G'}(v) \geq (1-\rho)(\mathit{rep}_G(v) - \mathit{infl}_G(u, v))$,

(iii) $\mathit{rep}_{G'}(v) \leq (1-\rho)(\mathit{rep}_G(v) - \mathit{infl}_G(u, v) + \mathit{rep}_G(u)) + \rho$.

Compared with Corollary 1, the only additional effect of sybils is to diminish all reputations by a factor of $(1-\rho)$, and increase the reputation of certain target nodes by up to ρ.

Proof of Theorem 3. We split the attack into two steps, first observing how reputations change when the sybils are added but no links are changed, then applying Theorem 2 for the step when only links change. Let G^+ be the intermediate graph where we add the sybils but do not change links. Assume the sybils

have self-loops so the transition probabilities are well-defined. We can compute $\text{rep}_{G^+}(U)$ by conditioning on whether $X_0 \in V$ or $X_0 \in S$, recalling that $\Pr[X_0 \in S] = \rho$.

$$\begin{aligned}\text{rep}_{G^+}(U) &= (1-\rho) \cdot \Pr_{G^+}[H(U) < J \mid X_0 \in V] + \rho \cdot \Pr_{G^+}[H(U) < J \mid X_0 \in S] \\ &= (1-\rho) \cdot \Pr_G[H(u) < J] + \rho \\ &= (1-\rho)\text{rep}_G(u) + \rho.\end{aligned}$$

In the second step, $\Pr_{G^+}[H(U) < J \mid X_0 \in V] = \Pr_G[H(u) < J]$ because hitting U in G^+ is equivalent to hitting u in G: all edges outside U are unchanged, and all edges to U originally went to u; also, the conditional distribution of X_0 given $[X_0 \in V]$ is equal to q, by our assumption on q'. The term $\Pr_{G^+}[H(U) < J \mid X_0 \in S]$ is equal to one, since $X_0 \in S$ implies $H(U) = 0 < J$. A similar calculation gives

$$\text{rep}_{G^+}(v) = (1-\rho)\text{rep}_G(v) + \rho \cdot \Pr_{G^+}[H(v) < J \mid X_0 \in S] = (1-\rho)\text{rep}_G(v).$$

The term $\Pr_{G^+}[H(v) < J \mid X_0 \in S]$ vanishes because S is disconnected, so a walk that starts in S cannot leave. Another similar calculation gives $\text{infl}_{G^+}(U, v) = (1-\rho)\text{infl}_G(u, v)$. Finally, we complete the sybil attack, obtaining G' from G^+ by making arbitrary changes to edges originating in U, and apply Corollary 1 (the version generalized to deal with sets) to G^+. Parts (i-iii) of this theorem are obtained by direct substitution into their counterparts from Corollary 1. □

Theorem 3 can also be generalized to deal with sets.

2.4 Computing Hitting Time

To realize a reputation system based on hitting time, we require an algorithm to efficiently compute the reputation of all nodes. Theorem 1 suggests several possibilities. Recall that $\pi(v)$ is the PageRank of v. Then $E[R_\alpha(v)] = 1/\pi(v)$ can be computed efficiently for all nodes using a standard PageRank algorithm, and the quantity $\Pr[R(v) \geq J]$ can be estimated efficiently by Monte Carlo sampling. Combining these two quantities using Theorem 1 yields $E[H_\alpha(v)]$.

It is tempting to estimate the reputation $\Pr[H(v) < J]$ directly using Monte Carlo sampling. However, there is an important distinction between the quantities $\Pr[R(v) \geq J]$ and $\Pr[H(v) < J]$. We can get one sample of either by running a random walk until it first jumps, which takes about $1/\alpha$ steps. However $\Pr[H(v) < J]$ may be infinitesimal, requiring a huge number of independent samples to obtain a good estimate. On the other hand, $\Pr[R(v) \geq J]$ is at least α since the walk has probability α of jumping in the very first step. If self-loops are disallowed, we obtain a better lower bound of $1 - (1 - \alpha)^2$, the probability the walk jumps in the first two steps. For this reason we focus on $\Pr[R(v) \geq J]$.

2.4.1 A Monte Carlo Algorithm

In this section we describe an efficient Monte Carlo algorithm to simultaneously compute hitting time for all nodes. To obtain accuracy ϵ with probability at least $1 - \delta$, the time required will be $O(\frac{\log(1/\delta)}{\epsilon^2\alpha^2}|V|)$ in addition to the time of one PageRank calculation. The algorithm is:

1. Compute π using a standard PageRank algorithm.[4] Then $E[R_\alpha(v)] = 1/\pi(v)$.
2. For each node v, run k random walks starting from v until the walk either returns to v or jumps. Let $y_v = \frac{1}{k} \cdot$ (# of walks that jump before returning to v).
3. Use y_v as an estimate for $\Pr[R(v) \geq J]$ in part (i) or (iii) of Theorem 1 to compute $E[H_\alpha(v)]$ or $\Pr[H(v) < J]$.

How many samples are needed to achieve high accuracy? Let $\mu = \Pr[R(v) \geq J]$ be the quantity estimated by y_v. We call y_v an (ϵ, δ)-approximation for μ if $\Pr[|y_v - \mu| \geq \epsilon\mu] \leq \delta$. A standard application of the Chernoff bound (e.g., see [49]) shows that y_v is an (ϵ, δ)-approximation if $k \geq (3\ln(2/\delta))/\epsilon^2\mu$. Using the fact that $\mu \geq \alpha$, it is sufficient that $k \geq (3\ln(2/\delta))/\epsilon^2\alpha$. Since each walk terminates in $\frac{1}{\alpha}$ steps in expectation, the total expected number of steps is no more than $\frac{3\ln(2/\delta)}{\epsilon^2\alpha^2}|V|$.

For massive graphs like the web that do not easily fit into main memory of a computer, it is not feasible to collect the samples in step 2 of the algorithm sequentially, because each walk requires random access to the edges, which is prohibitively expensive for data structures stored on disk. We describe a method from [25] to collect all samples simultaneously making efficient use of disk I/O.

Conceptually, the idea is to run all walks simultaneously and incrementally by placing tokens on the nodes recording the location of each random walk. Then we can advance all tokens by a single step in one pass through the entire graph. Assuming the adjacency list is stored on disk sorted by node, we store

[4] PageRank algorithms are typically iterative and incur some error. Our analysis bounds the additional error incurred by our algorithm.

the tokens in a separate list sorted in the same order. Each token records the node where it originated to determine if it returns before jumping. Then in one pass through both lists, we load the neighbors of each node into memory and process each of its tokens, terminating the walk and updating y_v if appropriate, else choosing a random outgoing edge to follow and updating the token. Updated tokens are written to the end of a new unsorted token list, and after all tokens are processed, the new list is sorted on disk to be used in the next pass.

The number of passes is bounded by the walk that takes the longest to jump, which is not completely satisfactory, so in practice we can stop after a fixed number of steps t, knowing that the contribution of walks longer than t is nominal for large enough t, since $\Pr[R \geq J,\ J > t] \leq \Pr[J > t] = (1-\alpha)^t$, which decays exponentially.

2.4.2 Finding Highly Reputable Nodes Quickly

We noted that estimating $\text{rep}(v) = \Pr[H(v) < J]$ directly by Monte Carlo sampling is troublesome in the case when this probability is very small. However, a benefit of this approach is that a single random walk gives a sample of $\text{rep}(v)$ for *all nodes*, and in some situations we may not care about nodes of low reputation. For example, suppose we want to find all nodes with reputation exceeding some fixed threshold c. A simple approach is to run many random walks and return all nodes for which the empirical estimate of $\text{rep}(v)$ exceeds c. We will show that the requisite number of walks depends very modestly on the size of the graph, and in some cases is independent of the size of the graph.

Specifically, we'll treat this as a classification problem, to label v as high rep-

utation if rep$(v) \geq c$, and low reputation otherwise. It will be very difficult to classify nodes with reputation almost exactly c, so we relax the problem slightly and allow either classification for some small interval $[a, b]$ containing c. Let $\epsilon = \frac{b-a}{b}$. Then we have the following result.

Theorem 4. *Using $O(\log(1/\delta)/a\epsilon^2)$ Monte Carlo samples, we can label all nodes as high or low reputation, such that the expected number of mislabeled nodes is at most $\delta|V|$. With $O(\log(|V|/\delta)/a\epsilon^2)$ samples, we can classify all nodes correctly with probability at least $1 - \delta$.*

The first result *does not depend on the size of the graph*, only on the threshold parameters a and ϵ. For graphs with highly skewed reputation distributions, a can be set to a high value to find the most reputable nodes very quickly. It is likely that real-world graphs will have skewed reputation distributions: for example, PageRank on the web graph has been observed to follow a power-law distribution [17].

Proof of Theorem 4. Let $\mu =$ rep(v), and suppose we perform k walks, letting

$$z_v = \frac{1}{k} \cdot (\text{\# of walks that hit } v \text{ before jumping})$$

be the empirical estimate for μ. The symmetric Chernoff bounds (see, e.g., [49] p. 64) give:

$$\Pr\left[z_v \geq (1+\epsilon)\mu\right] \leq \exp(-k\mu\epsilon^2/3)$$

$$\Pr\left[z_v \leq (1-\epsilon)\mu\right] \leq \exp(-k\mu\epsilon^2/3)$$

Recall that $\epsilon = \frac{b-a}{b}$, so $a = (1-\epsilon)b$, and $b > (1+\epsilon)a$. The probability that a

low-reputation node is misclassified is

$$\begin{aligned}\Pr\left[z_v \geq b \mid \mu \leq a\right] &\leq \Pr\left[z_v \geq b \mid \mu = a\right] \\ &\leq \Pr\left[z_v \geq (1+\epsilon)\mu \mid \mu = a\right] \\ &\leq \exp(-ka\epsilon^2/3).\end{aligned}$$

The probability that a high-reputation node is misclassified is

$$\begin{aligned}\Pr\left[z_v \leq a \mid \mu \geq b\right] &= \Pr\left[z_v \leq (1-\epsilon)b \mid \mu \geq b\right] \\ &\leq \Pr\left[z_v \leq (1-\epsilon)\mu \mid \mu \geq b\right] \\ &\leq \exp(-k\mu\epsilon^2/3) \\ &\leq \exp(-ka\epsilon^2/3).\end{aligned}$$

Choosing $k \geq \frac{3\ln(1/\delta)}{a\epsilon^2}$ ensures that each node is misclassified with probability at most δ, so the expected number of misclassified nodes is at most $\delta|V|$. Furthermore, by the union bound, the probability that any node is misclassified is at most $|V|\exp(-ka\epsilon^2/3)$, so choosing $k \geq \frac{3\ln(|V|/\delta)}{a\epsilon^2}$ ensures that all nodes are classified correctly with probability at least $1-\delta$. □

CHAPTER 3

THE PAGERANK GAME

3.1 Introduction

What is the effect of link spam on the network? In this chapter we analyze a game-theoretic model of strategic outlink placement called the *PageRank game*. The main contributions are:

- *Characterization of equilibria*. We characterize best response strategies and Nash equilibria in the PageRank game, and present a complete combinatorial characterization of α-*insensitive equilibria*, those graphs that are Nash equilibria for any setting of the PageRank jump parameter α. Equilibria may have a rich and varied structure — for example, every edge transitive graph is a Nash Equilibrium.
- *Best response dynamics*. We analyze best response dynamics in the PageRank game to determine which equilibria occur in realistic scenarios of play. We prove that best response dynamics converge as long as a minimal fairness condition is imposed on order of play. Contrary to the theoretical possibility of complex and highly-connected equilibria, best response dynamics lead only to highly fragmented equilibria that consist of many simple components.

3.2 Preliminaries

First we recall some basic facts about PageRank and related quantities.

3.2.1 PageRank and the α-Random Walk.

Let G be a directed graph. The *α-random walk* on G is a random walk that is modified to restart (also called a jump) with probability α in each step, by choosing the next node from distribution q instead of following a link. We assume that G has no self-loops and that $0 < \alpha < 1$. Some nodes of G may be "dangling", meaning they have no outlinks. We eliminate dangling nodes by preprocessing the graph as follows: create an artificial node called `wait` with a self-loop but no other outlinks, and add a link from each dangling node to `wait`. Hence, each time the walk reaches a dangling node, it immediately moves to `wait` and remains there until the next restart.

Let π be the stationary distribution (it is guaranteed to be unique by virtue of the random jump); then the PageRank of node i is defined as $\pi(i)$. Let $\{X_t\}_{t\geq 0}$ be the sequence of nodes visited by the α-random walk, where $X_0 \sim q$. Let $J \geq 1$ be the time of the first random jump — i.e., the minimum of all $t \geq 1$ such that X_t is chosen from the restart distribution instead of following a link. Let

$$\phi_{ij} = \Pr\left[\bigcup_{t=0}^{J-1}(X_t = j) \mid X_0 = i\right].$$

In words, ϕ_{ij} is the probability that j is reached before the first jump, given that the walk starts at i. Let $I\{\cdot\}$ be an indicator variable for the event in braces, and let N be the matrix with entries

$$N_{ij} = E\left[\sum_{t=0}^{J-1} I\{X_t = j\} \mid X_0 = i\right].$$

This is the expected number of visits to j before the first jump, given that the walk starts at i. Finally, let P be the transition matrix of the *standard* random walk on G, i.e., the simple random walk that does not make any random jumps.

Proposition 1. *Let N, P and Φ be defined as above. Then (i) the matrix N is equal to $\sum_{t=0}^{\infty} P^t(1-\alpha)^t = (I-(1-\alpha)P)^{-1}$, (ii) the PageRank vector π^T is proportional to $q^T N$, (iii) for all i and j, we have $N_{ij} = \phi_{ij} N_{jj}$, and (iv) for all i and j, the quantity ϕ_{ij} does not depend on j's outlinks.*

Proof. Since the decision to jump is made by coin flips that are independent of those for following links, J is independent of X_t for all t. Then

$$N_{ij} = E\left[\sum_{t=0}^{\infty} I\{X_t = j\} I\{J > t\} \mid X_0 = i\right]$$
$$= \sum_{t=0}^{\infty} \Pr[X_t = j \mid X_0 = i] \Pr[J > t] = \sum_{t=0}^{\infty} (P^t)_{ij} (1-\alpha)^t.$$

Hence $N = \sum_{t=0}^{\infty} P^t(1-\alpha)^t$. Since P is stochastic and $\alpha > 0$, this sum converges and is equal to $(I-(1-\alpha)P)^{-1}$. Part (ii) is well-known, and can be verified algebraically. We prefer a probabilistic argument. The value $(q^T N)_j$ is the expected number of visits to j between two restarts. Since the walk is a sequence of probabilistically identical segments delimited by restarts, the number of visits between restarts is proportional to stationary probability. See Proposition 3 in Chapter 2 of Aldous and Fill [1] for details. Part (iii) is a simple probabilistic statement. To count the number of visits to j starting from i, we first decide if the walk hits j, then count the number of visits starting from j. Part (iv) was proved in Chapter 2. Recall the intuition: to measure the probability that v is hit before the first jump, say, by Monte Carlo simulation, one need never follow a link leaving v. □

Remark. The matrix N is equal to the "personalized" PageRank matrix: Proposition 1 shows that $q^T N$ is proportional to the PageRank vector for any

restart distribution q. Hence the ith row of N yields the PageRank vector for a restart distribution concentrated on node i.

3.2.2 The PageRank Game

In the PageRank game, nodes place outlinks strategically to maximize PageRank. Let V be a set of n players, the nodes in a directed graph. A *strategy* for node v is a set of outlinks. An outcome is a directed graph G consisting of the outlinks chosen by each player. A *best response* for player v with respect to G is a set of outlinks E_v, such that, if v deletes its outlinks from G and adds outlinks E_v, then v maximizes PageRank over all possible choices of E_v. A directed graph G is a *Nash equilibrium* (or *Nash*) if the set of outlinks for each node is a best response: no player can increase its reputation by choosing different outlinks.

3.3 Nash Equilibria

In this section, we characterize Nash equilibria in the PageRank game. We begin by characterizing best response strategies and then we prove a decomposition theorem that allows us to focus on strongly connected graphs. For strongly connected graphs, we will show the following (summarized in Figure 3.1):

- All edges in a Nash equilibrium are bidirectional.
- The class of α-insensitive equilibria, consisting of those graphs that are equilibria for all values of α, has an exact combinatorial characterization.

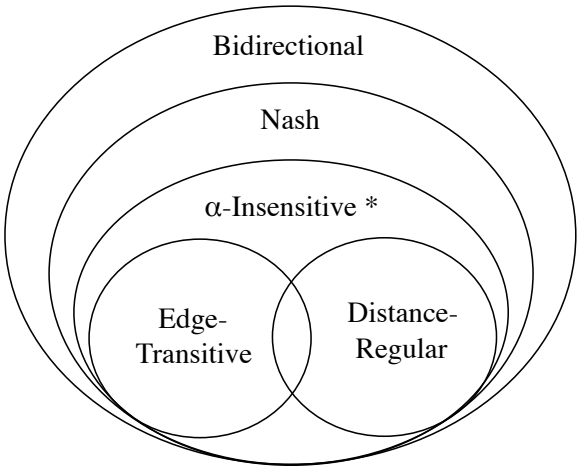


Figure 3.1: Venn diagram illustrating characterization of strongly connected Nash equilibria. Inclusion marked by an asterisk is not known to be proper.

The key property is an equivalence among all (bidirectional) edges: the number of walks of length t connecting the two endpoints of any edge is the same, for all t.

- All edge-transitive and distance-regular graphs are α-insensitive equilibria. These provide most known examples of Nash equilibria, including many with highly-connected and complex structure.

3.3.1 Best Responses

The first step toward understanding equilibria is to characterize best response strategies. Recall the short-cycle manipulation strategy in which node j places its outlinks to form short cycles so that the random walk from j will return to j quickly. This idea is the basis of a best response: one should link to nodes from which the walk will return quickly. We make a precise statement in terms of the

quantity ϕ_{ij} — recall that this is the probability that a walk starting at i reaches j before the first jump. Henceforth, we call ϕ_{ij} the *potential*[1] *of i with respect to j*.

Lemma 3. *A best response for node j consists of links to any subset of nodes that maximize $\phi_{\cdot j}$. If j has no inlinks, the empty set is also a best response.*

Proof. From Proposition 1, the PageRank vector is proportional to $q^T N$, and $N_{ij} = \phi_{ij} N_{jj}$, so node j wishes to maximize

$$(q^T N)_j = \sum_{i=1}^{n} q_i N_{ij} = N_{jj} \sum_{i=1}^{n} q_i \phi_{ij}.$$

Since ϕ_{ij} is independent of j's outlinks for all i, it suffices to maximize N_{jj}, the expected number of returns to j. Let ϕ_{jj}^+ be the probability that a walk starting at j returns to j before the first jump. Then the number of returns is a geometric random variable with parameter ϕ_{jj}^+: it counts the number of successes (returns) before the first failure (non-return) in a sequence of independent experiments that each succeed with probability ϕ_{jj}^+. Hence the expected number of returns is

$$N_{jj} = \frac{1}{1 - \phi_{jj}^+},$$

so it suffices for j to maximize ϕ_{jj}^+. By conditioning on the first step of the walk, this can be written as

$$\phi_{jj}^+ = \frac{1 - \alpha}{|\Gamma(j)|} \sum_{k \in \Gamma(j)} \phi_{kj},$$

where $\Gamma(j)$ is the set of neighbors of j. We see that ϕ_{jj}^+ is equal to a constant times the average potential of j's out-neighbors, and, as before, these potentials are independent of j's outlinks. Hence, node j can only control which terms appear in the average. To maximize the average, node j should link only to

[1]For an undirected graph G, take the electrical network corresponding to G having unit conductance on each edge, and additional links of conductance α from each node to a sink. Connect a one-volt battery causing potential of 1 volt at j and 0 volts at the sink. Then ϕ_{ij} is the electrical potential, in volts, at node i.

nodes of maximum potential. If j places no links, our preprocessing step creates a link to `wait`, which has zero potential. This can only be a best response if all nodes have zero potential, meaning that j has no inlinks. □

3.3.2 Bidirectionality

Lemma 3 has a very interesting consequence. It is straightforward to show that only in-neighbors can maximize potential, leading to a bidirectional structure in all Nash equilibria.

Lemma 4. *If j has any inlinks, then j links only to in-neighbors in best response strategies.*

Proof. Suppose, for contradiction, that j has an in-neighbor i but elects to link to some other node k which is *not* an in-neighbor. If $\phi_{kj} = 0$, then k cannot be a best response because $\phi_{ij} > 0$. If $\phi_{kj} > 0$, then we again condition on the first step of the walk to write

$$\phi_{kj} = \frac{1-\alpha}{|\Gamma(k)|} \sum_{\ell \in \Gamma(k)} \phi_{\ell j}.$$

Since the entire expression is nonzero and $1 - \alpha < 1$, we have $\phi_{kj} < |\Gamma(k)|^{-1} \sum_{\ell \in \Gamma(k)} \phi_{\ell j}$. Hence, at least one term in this average must exceed ϕ_{kj}, so there is a node ℓ such that $\phi_{\ell j} > \phi_{kj}$, and k cannot be a best response. □

Lemma 4 indicates that all edges are bidirectional except for those that originate at a node with no inlinks. It also allows us to restate the Nash conditions in a useful way.

Corollary 4 (Restatement of Nash conditions)**.** *A graph G is a Nash equilibrium if and only if for all $(i, j), (i, k) \in E$, we have $\phi_{ji} = \phi_{ki}$ (or, equivalently, $N_{ji} = N_{ki}$).*

Proof. If G is a Nash, then for any node i, all of its neighbors must maximize potential. Hence any two neighbors j and k have the same potential. Conversely, if all neighbors have the same potential, they necessarily maximize potential because a non-neighbor cannot maximize potential. Finally we note, since $N_{\ell i} = \phi_{\ell i} N_{ii}$ for all ℓ, that

$$\phi_{ji} = \phi_{ki} \iff \phi_{ji} N_{ii} = \phi_{ki} N_{ii} \iff N_{ji} = N_{ki}.$$

□

These facts allow us to characterize the strongly connected components of Nash equilibria. We say that a graph is bidirectional if all of its edges are bidirectional.

Theorem 5 (Decomposition)**.** *In a Nash equilibrium, every strongly connected component is either: (i) a component of size two or more with no outlinks whose induced subgraph is a bidirectional Nash equilibrium, or (ii) a single node with no inlinks, and with outlinks only to non-singleton strongly connected components.*

Proof. Suppose C is a strongly connected component of size two or more. Each node $i \in C$ must have at least one in-neighbor, so Lemma 4 ensures that i links only to in-neighbors. Hence, all links originating in C are reciprocated, so there are no links leaving C and the induced subgraph is bidirectional.

To see that the induced subgraph is a Nash, we must consider how potentials in the subgraph differ from those in G. Let $i, j \in C$, and let ϕ_{ij}^C be the potential of i with respect to j in the induced subgraph. Because C has no outlinks, a random walk in G that starts at node i is probabilistically identical to a random walk that starts at i in the subgraph, *until the time of the first restart*, at which point

the walk in G may escape to some other component. However ϕ_{ij} measures the probability of hitting j *before* the first restart, so $\phi_{ij}^C = \phi_{ij}^G$. Since potentials within C do not change, the restated Nash conditions from Corollary 4 remain true in the induced subgraph.

Let $\{j\}$ be a strongly connected component of size one. Node j cannot have an inlink (i, j), otherwise j would link to i to form a non-singleton component by Lemma 4. Similarly, j cannot link to another singleton because that link would also be reciprocated. Node j may link to a node k in a non-singleton component, as long as j does not become a unique best response for k. □

Remark. The decomposition theorem allows us to focus on strongly connected Nash equilibria, which must be bidirectional, as the building blocks for any other equilibrium. Henceforth, we restrict attention to strongly connected graphs and dispense with directionality, speaking only of degrees, links, and neighbors, instead of their directed equivalents.

3.3.3 Examples

It is instructive to look at some examples using the tools developed thus far. In Figure 3.2, graph (a) is not a Nash equilibrium. It is easy to see that $\phi_{ji} = 1 - \alpha$, because the only way the walk starting at j can fail to hit i before jumping is if it jumps in the very first step, an event that happens with probability α. But $\phi_{ki} < 1 - \alpha$, because the walk starting at k may fail to hit i *either* by jumping in the first step (which happens with probability α), *or* by stepping away from i (which happens with probability $(1 - \alpha)/2$). Hence $\phi_{ki} < \phi_{ji}$, so i will drop its link to k in a best response.

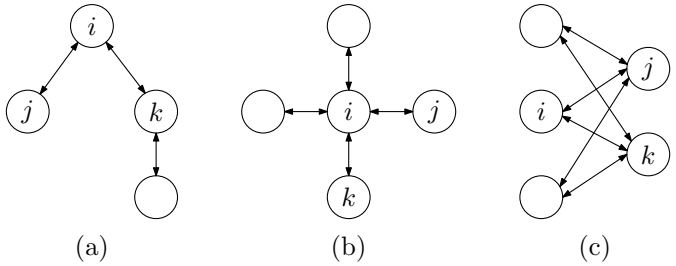


Figure 3.2: Nash equilibrium examples. Graph (a) is not a Nash equilibrium; graphs (b) and (c) are Nash equilibria.

However, it is easy to see that graphs (b) and (c) are equilibria by a simple symmetry argument. Both have the property that, for any node i, all of its neighbors "look identical" in the following sense: if j and k are two neighbors of i, then there is a graph automorphism mapping j to k that fixes i. Such an automorphism ensures that $\phi_{ki} = \phi_{ji}$, and hence the restated Nash conditions of Corollary 4 are satisfied. This symmetry property is related to another property called edge-transitivity. We discuss both in more detail in Section 3.3.5.

3.3.4 α-Insensitive Equilibria

Examples (b) and (c) in Figure 3.2 have an additional property. Because we argued that they are equilibria by symmetry, and without appealing to the particular value of the jump parameter α, they are Nash equilbria *all* settings of α.

Definition 5 (α-insensitive Nash equilibrium)**.** *G is an* α-insensitive Nash equilibrium *is G is a Nash equilibrium for all* $0 < \alpha < 1$.

In this section we present a combinatorial characterization of α-insensitive equilibria. The result is a means to compare Nash equilibria with other classes

of graphs from the literature, and to construct nearly all known examples of equilibria.

At first glance, α-insensitivity seems to be an unnatural condition. The PageRank random walk behaves very differently for extreme values of α: when α is close to zero, the walk rarely jumps and behaves similarly to the standard random walk on G, while for values of α that are close to one, almost every step is made by sampling from the restart distribution. In any real application, there is only one choice of α, and one expects strategic nodes to optimize link-placement for that particular parameter setting. However, we justify the study of α-insensitivity by showing that it is equivalent to a (seemingly) much weaker condition that is very natural. Suppose that α_0 is the stated parameter setting, but we restrict to equilibria that have a minimal amount of stability with respect this value.

Definition 6 (α_0-stable Nash equilibrium)**.** *G is an α_0-stable Nash equilibrium if there exists some $\epsilon > 0$ such that G is a Nash equilibrium for all $\alpha \in [\alpha_0 - \epsilon, \alpha_0 + \epsilon]$.*

If G is a Nash equilibrium that is *not* α_0-stable, then some node is playing a strategy that is a best response for parameter α_0, but is no longer a best response after aribtrary perturbation with *any* interval containing α_0. The condition of α_0-stability is very natural to consider, because the exact parameter setting of a reputation system is rarely known to participants. For example, many research papers cite the parameter setting of $\alpha = .15$, but the true value used by Google is a trade secret. The following Lemma implies that the two conditions are actually equivalent.

Lemma 5. *If G is a Nash for at least n different values of α, then G is α-insensitive.*

Proof. We use the restated Nash conditions from Corollary 4. To indicate dependence on α, we now write $\phi_{ij}(\alpha)$ and $N_{ij}(\alpha)$ instead of ϕ_{ij} and N_{ij}. We will show that the equality of $N_{ji}(\alpha)$ and $N_{ki}(\alpha)$ can be expressed as the equality between two polynomials of degree $n-1$ in the variable α. Hence, if they are equal for at least n values of α, they are equal for *all* $\alpha \in (0,1)$. Let $M(\alpha) = I - (1-\alpha)P$, and recall from Proposition 1 that $N(\alpha) = M^{-1}(\alpha)$. Let $M_{[j,i]}(\alpha)$ denote the matrix of size $(n-1)\times(n-1)$ obtained from $M(\alpha)$ by deleting row j and column i. Using the formula for the matrix inverse in terms of cofactors,

$$N_{ji}(\alpha) = M^{-1}_{ji}(\alpha) = \frac{(-1)^{j+i}\det M_{[j,i]}(\alpha)}{\det M(\alpha)}.$$

Hence, $N_{ji}(\alpha) = N_{ki}(\alpha)$ if and only if

$$(-1)^{j+i}\det M_{[j,i]}(\alpha) = (-1)^{k+i}\det M_{[k,i]}(\alpha). \tag{3.1}$$

Each determinant in (3.1) is a sum of products of $n-1$ matrix elements, each of which is a monomial in α. Hence both sides of (3.1) are polynomials of degree $n-1$ in α; if they are equal for at least n distinct values of α, they are equal as polynomials. □

Corollary 5. *G is an α-insensitive Nash equilibrium if and only if G is an α_0-sensitive Nash equilibrium for some $0 < \alpha_0 < 1$.*

Combinatorial Characterization

Before we proceed to characterize α-insensitive Nash equilibria, we recall two basic definitions from graph theory. A graph is *regular* if all of its vertices have the same degree. A bipartite graph is *semiregular* if, within each of the two partitions, every vertex has the same degree.

Theorem 6 (Characterization of α-insensitive Nash equilibria)**.** *Let $G = (V, E)$ be a bidirectional connected graph with adjacency matrix A. Then G is an α-insensitive Nash equilibrium if and only if both of the following properties hold: (i) G is either regular or bipartite and semiregular, and (ii) for all $(i, j), (k, l) \in E$, and for all $t \geq 0$, $(A^t)_{ij} = (A^t)_{kl}$.*

Proof. G is an α-insensitive Nash equilibrium if and only if

$$\forall\ 0 < \alpha < 1 \text{ and } \forall (i, j), (i, k) \in E,\ N_{ji}(\alpha) = N_{ki}(\alpha). \tag{3.2}$$

Recall that $N_{ji}(\alpha) = \sum_{t=0}^{\infty} (P^t)_{ji}(1 - \alpha)^t$, so $N_{ji}(\alpha)$ is the generating function for the sequence $\{(P^t)_{ji}\}_{t\geq 0}$, evaluated at $1 - \alpha$. Similarly, $N_{ki}(\alpha)$ is the generating function for $\{(P^t)_{ki}\}_{t\geq 0}$. The two generating functions are equal if and only if the sequences are identical, so an equivalent statement of the Nash conditions is:

$$\forall t \geq 0 \text{ and } \forall (i, j), (i, k) \in E,\ (P^t)_{ji} = (P^t)_{ki}. \tag{3.3}$$

Now, we will show that (3.3) implies the two conditions in the theorem. Taking $t = 1$ in (3.3), we have that for all $(i, j), (i, k) \in E$,

$$\deg(j)^{-1} = P_{ji} = P_{ki} = \deg(k)^{-1}.$$

Hence, any two nodes that share a neighbor have the same degree. Then, it is easy to see that any walk in G is a sequence of nodes in which either: (a) all nodes have the same degree, or (b) the sequence alternates between nodes with degree d_1 and nodes with degree d_2. Because G is connected, this implies that G is either regular or bipartite and semiregular.

To prove property (ii) from the statement of the theorem, let W_{ji}^t be the set of all walks of length t from j to i. Note that $|W_{ji}^t| = (A^t)_{ji}$. We already argued that

all walks in G are sequences of nodes that alternate in degree between d_1 and d_2 (where $d_1 = d_2$ in the regular case). Let d_1 be the degree of j. Then all walks of length t starting at j have probability $\left(d_1^{\lceil t/2 \rceil} d_2^{\lfloor t/2 \rfloor}\right)^{-1}$. Hence,

$$(P^t)_{ji} = \sum_{w \in W_{ji}^t} \Pr[w] = (A^t)_{ji} \left(d_1^{\lceil t/2 \rceil} d_2^{\lfloor t/2 \rfloor}\right)^{-1} \tag{3.4}$$

A similar argument holds for $(P^t)_{ki}$, allowing us to conclude for any t and for any $(i,j),(i,k) \in E$, that $(P^t)_{ji} = (P^t)_{ki}$ if and only if $(A^t)_{ji} = (A^t)_{ki}$. Hence, the Nash conditions can be restated as

$$\forall t \geq 0 \text{ and } \forall (i,j),(i,k) \in E,\ (A^t)_{ji} = (A^t)_{ki}. \tag{3.5}$$

In other words, any two edges that share an endpoint have the same number of walks of length t connecting their endpoints, for all t. Because G is bidirectional, $(A^t)_{ji} = (A^t)_{ij}$. Using this fact and the fact that G is connected, we can extend the equality in (3.5) along paths to conclude that *any* two edges have the same number of walks of length t connecting their endpoints. Hence (3.3) implies condition (ii) from the statement of the theorem:

$$\forall t \geq 0 \text{ and } \forall (i,j),(k,\ell) \in E,\ (A^t)_{ij} = (A^t)_{k\ell}. \tag{3.6}$$

We have proven that properties (i) and (ii) in the theorem are necessary. To see that they are sufficient, note that property (i) is all that is needed to derive (3.4) and (3.5). Then, using these two equations, it is easy to see that (3.6) implies (3.3). □

3.3.5 Related Classes of Graphs

The characterization in Theorem 6 is important because it allows us to compare α-insensitive Nash equilibria to several other classes of graphs, and, in particular, to construct many examples of Nash equilibria.

We start by comparing property (ii) from Theorem 6 to two graph-theoretic concepts from the literature. In each case, we define a property by first defining a relation $\sim$ among pairs of vertices, and then imposing the condition that a combinatorial equivalence exist among all vertex pairs that are related by $\sim$:

$$(A^t)_{ij} = (A^t)_{kl} \text{ for all } t \geq 0, \text{ and for all } i, j, k, \ell \in V \text{ such that } (i,j) \sim (k,\ell). \tag{3.7}$$

Three different properties are obtained from the three relations:

$$(i,j) \sim_A (k,\ell) \iff i = j \text{ and } k = \ell$$

$$(i,j) \sim_B (k,\ell) \iff (i,j), (k,\ell) \in E$$

$$(i,j) \sim_C (k,\ell) \iff d(i,j) = d(k,\ell)$$

The property obtained by using relation $\sim_A$ in (3.7) defines the class of *walk-regular* graphs, due to Godsil and MacKay [29]. The relation $\sim_B$ results in condition (ii) from Theorem 6 — we call a graph that satisfies this condition *edgewise walk-regular*. The relation $\sim_C$ results in a property held by all *distance-regular* graphs. Distance-regular graphs are defined by a slightly different combinatorial equivalence among all pairs of vertices at distance d. See [9] and Chapter 11 of [28] for more details; the fact the property above is held by distance-regular graphs is a consequence of Lemmas 1.1 and 1.2 in Chapter 13 of Godsil [28].

Lemma 6. *Every distance-regular graph is an α-insensitive Nash.*

Proof. Every distance-regular graph is regular [28], hence property (i) of Theorem 6 is satisfied. It is easy to see that $(i,j) \sim_B (k,\ell)$ implies $(i,j) \sim_C (k,\ell)$: if $(i,j),(k,\ell) \in E$, then $d(i,j) = d(k,\ell) = 1$. Hence, (3.7) holds for all $(i,j) \sim_B (k,\ell)$, so property (ii) is also satisfied. □

Another relevant concept from graph theory is *edge-transitivity* (see, e.g., [6]). Recall that an *automorphism* on G is a permutation π of the vertices such that $(\pi(i),\pi(j)) \in E$ if and only if $(i,j) \in E$. Let Aut(G) be the set of all automorphisms on G. An undirected graph is edge-transitive if, for any pair of edges, there is an automorphism mapping one edge to the other.

Definition 7. *G is* edge-transitive *if for all $(i,j),(k,\ell) \in E$, there exists $\pi \in Aut(G)$ such that either $\pi(i) = k$ and $\pi(j) = \ell$, or $\pi(i) = \ell$ and $\pi(j) = k$.*

Recall also the related symmetry property from Section 3.3.3; we argued informally that a graph is a Nash if for each node, all of its neighbors "look identical".

Condition 1. *For all $(i,j),(i,k) \in E$, there exists $\pi \in Aut(G)$ such that $\pi(i) = i$ and $\pi(j) = k$.*

It is not hard to show that any graph that satisfies Condition 1 is edge-transitive, but the converse may not be true. However, from the characterization in Theorem 6 we can see that the weaker condition of edge-transitivity is also sufficient to be an α-insensitive Nash.

Lemma 7. *Every edge-transitive graph is an α-insensitive Nash.*

Proof. Every edge-transitive graph is either regular or bipartite and semiregular. If G is edge-transitive, then for all $(i,j),(k,\ell) \in E$, there is an automorphism that

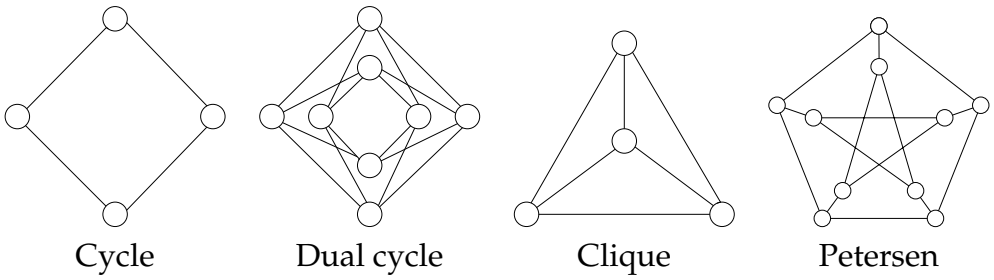


Figure 3.3: Edge-transitive Nash equilibria.

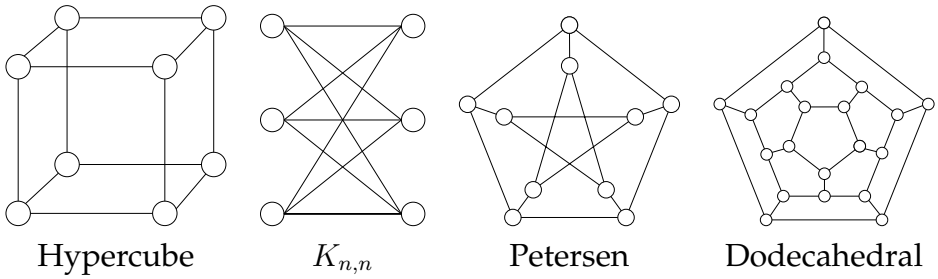


Figure 3.4: Distance-regular Nash equilibria.

maps (i, j) to (k, ℓ). This gives a bijection between the set of walks connecting (i, j) and those connecting (k, ℓ), so $(A^t)_{ij} = (A^t)_{k\ell}$ for all t. □

In fact, edgewise walk-regularity (property (ii) from Theorem 6) can be seen as a combinatorial relaxation of edge-transitivity: rather than requiring an automorphism mapping any edge to another, it requires the weaker combinatorial equivalence that all edges share the same number of walks of each length connecting their endpoints.

Examples. We have already seen two examples of edge-transitive graphs: the star and bipartite clique from Figure 3.2, parts (b) and (c). Figure 3.3 shows several more examples. In addition to stars and bipartite cliques, all cycles, dual cycles, cliques, hypercubes and many more families of graphs (some of

them substantially more complex, and often based on algebraic constructions) are edge-transitive [6,64]. Figure 3.4 shows several examples of distance-regular graphs. All of the following graphs are distance-regular: cliques, hypercubes, bipartite cliques with equally-sized partitions ($K_{n,n}$), and odd cycles [9, 63]. All of these examples happen to also be edge-transitive, a fact that is true of most small distance-regular graphs but not in general — e.g., Phelps constructed distance-regular graphs based on Latin squares that have no nontrivial automorphisms [53].[2] On the other hand, it is easy to point out examples of edge-transitive graphs that are not distance-regular: any edge-transitive graph that is not regular, such as the star or asymmetric bipartite clique, cannot be distance-regular.

3.3.6 α-Sensitive Equilibria

At the time of our initial work on this subject, we left open the question of existence of Nash equilibria that are *not* α-insensitive. Since that time, Chen et al. [15] have resolved this question. They constructed a family of graphs that are α-*sensitive* Nash equilibria, i.e., that are Nash equilibria only for a particular value of α. In this section, we review their construction.

The example is called $G_{n,m}$, and it is parameterized by $n \geq m \geq 0$. The construction is illustrated in Figure 3.5. It consists of vertices $U = \{u_1, \ldots, u_n\}$ and $V = \{v_{ij} \mid i = 1, \ldots, m,\ j = 1, \ldots, n\}$ such that

- The sets $V_i = \{v_{ij} \mid j = 1, \ldots, n\}$ are fully connected,
- Each vertex $u_j \in U$ is connected to the node $v_{ij} \in V_i$ for all $i = 1, \ldots, m$,

[2] Thanks to Chris Godsil for pointing out this example.

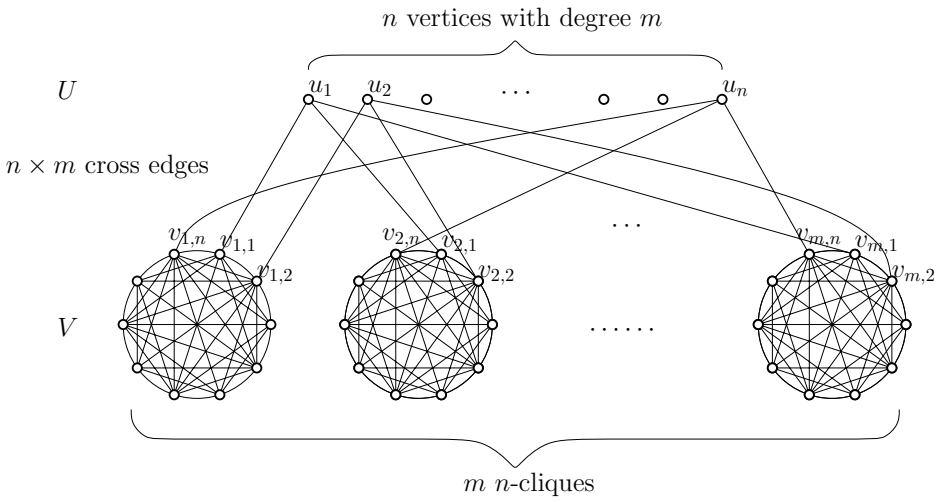


Figure 3.5: Construction of $G_{n,m}$ from [15]. Figure courtesy of the authors.

- All edges are bidirectional.

It is easy to see by symmetry that the Nash conditions are satisfied for every node in U. Similarly, for any node v_{ij}, all of its neighbors *within the clique* V_i have the same potential. But, node v_{ij} has one additional neighbor: the node $u_j \in U$. For $G_{n,m}$ to be a Nash, it must be the case that u_j has the same potential with respect to v_{ij} as do the neighbors of v_{ij} within the clique; by symmetry, this condition is sufficient. The result of Chen et al. is that there exists a parameter setting for which these two potentials are equal.

Theorem 7 (Chen et al. [15])**.** *If* $4 \leq m \leq n-2$*, then there exists* α *such that* $G_{n,m}$ *is an* α*-sensitive Nash equilibrium.*

The idea of their proof is as follows. For clarity of notation, we now write potentials as $\phi_\alpha(u, v)$ instead of $\phi_{uv}(\alpha)$. For α near 1, the walk jumps frequently, so the short-term behavior of the walk is the dominant consideration in determining potential. For example, consider the first order approximation given by

the probability of hitting v_{ij} in the first step:

$$\phi_\alpha(u_j, v_{ij}) \approx \frac{1-\alpha}{m}, \quad \phi_\alpha(v_{ik}, v_{ij}) \approx \frac{1-\alpha}{n}.$$

By this heuristic, v_{ij} should link to u_j, the node with smaller degree. Indeed they prove that there is some $\alpha_1 < 1$ such that $\phi_\alpha(u_j, v_{ij}) > \phi_\alpha(v_{ik}, v_{ij})$ for all $\alpha \in [\alpha_1, 1)$.

Next, they argue that for α very close to zero, the long-term behavior of the random walk is dominant. If v_{ij} links to v_{ik}, then the walk is likely to stay inside V_i for some time and has many chances to hit v_{ij} directly before jumping. However, if v_{ij} links to u_j, the walk will move to another clique with probability $1-\frac{1}{m}$, and may take a long time to return to V_i. Hence v_{ij} should link to v_{ik}. Again, they show formally that there exists $\alpha_0 > 0$ such that $\phi_\alpha(v_{ik}, v_{ij}) > \phi_\alpha(u_j, v_{ij})$ for all $\alpha \in (0, \alpha_0]$.

Finally, they argue that each potential is a continuous function of α, and then invoke the mean value theorem to conclude that there is some $\alpha^* \in [\alpha_0, \alpha_1]$ such that $\phi_{\alpha^*}(v_{ik}, v_{ij}) = \phi_{\alpha^*}(u_j, v_{ij})$. This indicates that $G_{n,m}$ is an α^*-sensitive Nash equilibrium.

3.4 Best Response Dynamics

We have seen that the PageRank game has a rich set of equilibria, including edge-transitive graphs, distance-regular graphs, and the family $G_{n,m}$ of α-sensitive equilibria. However, these examples are somewhat surprising: those that are large and highly connected require a great deal of symmetry or regularity in their link structure, a property that we do not expect in real-world

networks, and one that is difficult to imagine as the result of uncoordinated competitive play.

In this section, we will refine the analysis of Nash equilibria by analyzing *best response dynamics*, a realistic model of sequential play for the PageRank game in which nodes take turns playing best response strategies. This helps overcome the fact that the Nash equilibrium can be a rather coarse solution concept: it describes *only* the stability of an outcome, but does not consider how players might arrive at such an outcome in any realistic scenario of play. Best response play, if it converges, always leads to a Nash equilibrium, but in this case we have also modeled an explicit sequence of strategic moves that leads there from an initial state of our choosing. With reasonable choices for the initial network and order of play, best response dynamics differentiate between common versus uncommon Nash equilibria. For the PageRank game, we find that, in contrast to the theoretical possibility of large and highly connected equilibria, best response dynamics always lead to highly fragmented equilibria, consisting of many small disjoint components.

3.4.1 Definition

To define best response dynamics, we choose an initial network and a method to determine the order of play. Then, during each turn, a single node rewires its links to play a best response in the current network. In general, we allow a node to play any best response, i.e., to link to any subset of nodes that maximize potential. However, *we always assume that a singleton does not place any outlinks.* We will focus primarily on *random round robin* play: the game proceeds in rounds

during which each node plays in a random order selected at the beginning of the round. However, several other variants are possible and we will show that best response dynamics converge as long as the order of play satisfies a minimal fairness condition.

Condition 2 (Fair order of play)**.** *For all $i, j \in V$, after any turn of player i, there is probability at least $1/2$ that player j will be the next of the two nodes to play.*

This condition is obviously satisfied by random round robin player-order. It is also satisfied by other common ordering methods, such as: (1) *deterministic round robin*, in which play proceeds in rounds, but, during each round, players always play in the same pre-determined order, and (2) *random*, in which the next player is always chosen uniformly at random.

3.4.2 Convergence

If, after some finite number of turns, no player changes its strategy, we say that best response play has converged. In this case, the outcome is a Nash equilibrium, because every node is playing a best response in the current network. We will see that, under Condition 2, best response dynamics always converge.

Lemma 8. *If Condition 2 is satisfied, then any time an edge (i, j) is deleted, there is probability at least $1/2$ that it will never be added again.*

Proof. Suppose (i, j) is deleted during turn t. If edge (j, i) does not exist, then neither edge can be re-created after time t, because neither node links to the other, and hence neither is a best response for the other. Assume (j, i) does exist. If j is the next of the two nodes to play, then j will remove the edge (j, i)

because i is not an in-neighbor and hence not a best response. Again, neither edge can be re-created. The event that j plays next happens with probability at least $1/2$. □

Theorem 8. *Best response dynamics in the PageRank game converge with probability 1.*

Proof. If play does not converge, then some edge (i, j) must be deleted and then re-created infinitely often. However, the probability that (i, j) is deleted and re-created k times is no more than 2^{-k}, which goes to zero as $k \to \infty$, so the event that (i, j) is deleted and added infinitely often has probability zero. □

Remark. Best response dynamics can fail to converge if player order is chosen adversarially. For example, consider a Nash equilibrium where player i has two neighbors j and k. Player i may delete its link to j and then re-create that link as long as j does not play in the interim. The order can be arranged so this happens infinitely often, but any such ordering has probability zero under Condition 2.

3.4.3 Tie-Breaking

Consider a simple variant of best-response dynamics where a player must always break ties when selecting a best response strategy, so it links to a *single* node that maximizes potential.

Lemma 9. *Suppose players always link to a single node of maximum potential. Then best response dynamics converge to a union of singletons and two-cycles.*

Proof. Convergence is guaranteed by Theorem 8. Tie-breaking ensures that the resulting graph has out-degree at most one. Furthermore, all links are bidirectional (recall our assumption that singletons place no outlinks). The only bidirectional components with outdegree at most one are the singleton and two-cycle. □

We can prove a similar result if players choose among best response strategies at random, as long as there is some chance during each turn that a player does not link to *all* nodes that maximize potential.

Lemma 10. *Suppose each player chooses among best response strategies randomly in a way that satisfies the following condition: if S is the set of nodes that maximize potential and $|S| > 1$, then, with probability at least c, the player links to a proper subset of S. Then, with probability one, best response dynamics converge to a union of singletons and two-cycles.*

Proof. The proof is similar to the proof of Theorem 8. Let S_i^t be the set of nodes that maximize potential for player i during its tth turn. If $|S_i^t| > 1$, then there is constant probability that $|S_i^{t+1}| < |S_i^t|$ because i fails to link to some node $j \in S_i^t$, and that node plays again before i. Hence, with probability one, all nodes have degree one upon convergence, so the outcome must be a union of singletons and two-cycles. □

3.4.4 Experiments

With no assumptions about tie-breaking, we cannot guarantee that best response play leads to very simple equilibria. In this section, we conduct ex-

periments that simulate best response dynamics in a real web hostgraph. Our starting point is the hostgraph G from the WEBSPAM-UK2006 dataset [12]. The nodes of G are the 11,402 web hosts (e.g., `www.bbc.co.uk`) taken from a crawl of the `.uk` domain conducted in 2006; a link is present from u to v if any page from host u links to any page from host v. There are a total of 730,774 links.

We simulate best response dynamics in random round robin order. During each turn, the current player rewires to link to *all* neighbors that maximize potential. This choice is made to maximize the number of links retained during play; we have already shown that that tie-breaking strategies that remove links always lead to singletons and two-cycles. We ran ten runs of best response dynamics until convergence, each time with different random seeds to determine the order of play in each round. Convergence happened quickly: seven of ten trials converged in just 3 rounds, while the longest took 9 rounds.

Outcomes. Once again, contrary to the possibility of complicated equilibria, all equilibria reached by simulation consisted of many small and simple components. Table 3.4.4 shows the classification of all components. Since all edges are bidirectional, it is simpler to classify the graphs as if they were undirected, e.g., we classify the directed two-cycle as an undirected path P_2 on two nodes. This was by far the most commonly occurring component. Other graphs that occurred were singletons, stars (S_n), cycles (C_n), cliques (K_n) and bipartite cliques ($K_{m,n}$), most occurring only in relatively small sizes. The biggest component in each trial was a star; the largest contained 109 vertices. The closest runner-up was a bipartite clique $K_{2,58}$, essentially a star with dual hubs.

Table 3.1: Classification of strongly connected components in Nash equilibria reached by best response dynamics. (Note: components are counted toward the most specific classification, as ordered from left to right. For example a P_2 is not counted as a S_n or K_n, and a C_3 is not counted as a K_n.)

Trial	Single-	Paths		Stars	Cycles		Cliques	
	tons	P_2	P_3	S_n	C_3	C_4	K_n	$K_{2,n}$
1	1274	3642	340	113	191	34	19	13
2	1192	3630	341	134	182	24	20	17
3	1178	3612	345	127	183	33	21	15
4	1278	3609	325	120	204	29	20	21
5	1158	3641	304	125	212	31	22	16
6	1244	3566	394	147	157	25	19	11
7	1284	3578	361	116	179	27	20	22
8	1205	3639	342	115	196	25	20	25
9	1174	3710	342	129	191	25	19	12
10	1286	3559	367	134	171	23	21	16

Dismantling connectivity. In addition to the final outcome, we are interested in what happens along the way. How quickly does best response play converge? What happens to global connectivity? We observed that best response play converges very quickly and is incredibly efficient at dismantling global connectivity in the graph. During most turns, a player deletes all but a single link. Moreover, because the PageRank game favors "mutual admiration" dynamics in which pairs of nodes break off into two-cycles, it is much more damaging to global connectivity than other schemes that remove a similar number of edges.

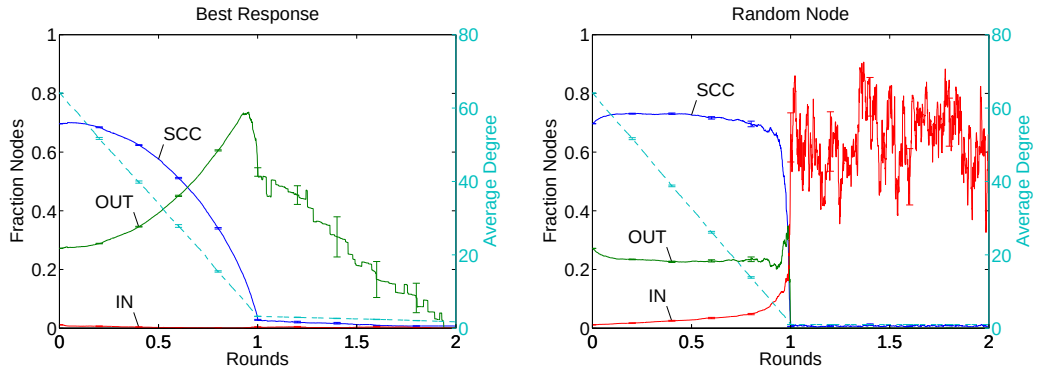


Figure 3.6: Evolution of connectivity during (a) best-response dynamics and (b) random-rewiring. Each plot shows the fraction of nodes in SCC, IN, and OUT, and the average outdegree (dashed) during the first two rounds of play. Results are averaged over 10 trials, with error bars indicating one standard error.

Figure 3.6, part (a) shows the evolution of several connectivity measurements during the first two rounds of play. We measured the average degree and the sizes of the "bow-tie" components SCC, OUT and IN [8]. SCC is the largest strongly connected component, OUT is the set of remaining nodes that are reachable from the SCC, and IN is the set of remaining nodes that can reach the SCC. The initial graph (round 0) is typical of web graphs, with a large fraction of nodes in SCC, and the remaining nodes split between IN and OUT. In this case, IN is very small, a property we expect in a web crawl done from a small initial set because, by definition, IN is not reachable from most pages on the web.

During the first round, SCC is completely dismantled. At the end of the round, the average degree is 3.3 and OUT contains about half the nodes, but then proceeds to break apart during the second round. For comparison, part (b) of Figure 3.6 shows the results of a random rewiring strategy, where, instead of

playing a best response, each player selects a single outlink uniformly at random (the result after one round will be a random graph with outdegree exactly one). This is guaranteed to remove *more* edges than best response dynamics, yet we see that the global bow-tie structure remains relatively stable until the average degree has is reduced to be very close to one, while best response play has already systematically dismantled SCC at a much higher average degree.[3]

[3]The explosion of IN during round two in Figure 3.6 part (b) is an artifact of the fact that after one round of play this is a random graph with outdegree one. In such a graph, paths can converge but never split or end, implying that each component contains a single directed cycle with directed trees pointing into the cycle. Hence OUT is necessarily empty, and IN contains the directed trees leading into the biggest cycle.

CHAPTER 4

COLLECTIVE HIDDEN MARKOV MODELS FOR MODELING BIRD MIGRATION

4.1 Introduction

In this chapter, we explore a family of inference problems on *collective* hidden Markov models, a generalization of hidden Markov models (HMMs) designed to model population-wide behavior that results from a collection of individuals that each behave according to a HMM. We present algorithms and hardness results for the hidden-data reconstruction problem in several different variants of the model obtained by making different assumptions about how the population is observed, and what data about individuals should be reconstructed. The reconstruction problems we consider are analogous to that solved by the Viterbi algorithm in HMMs, which can be interpreted as finding a shortest path in the HMM *trellis graph*; our algorithms use convex optimization based on network flows in the trellis graph.

4.1.1 Motivating Application

This work was motivated by a novel and important application in ecology: inferring bird migration paths from a large collection of spatial observations. The eBird database hosted by the Cornell Lab of Ornithology contains millions of bird observations from throughout North America, reported by the general public at the eBird website[1] [50, 61]. Observations report location, date, species

[1] `http://www.ebird.org`

and number of birds observed. The eBird data set is very rich; the human eye can easily discern migration patterns from animations showing the observations as they unfold over time on a map of North America. However, the eBird data are *static*, and they do not explicitly record movement, only the locations of birds (or relative number of birds at different locations) at different points in time. Conclusions about migration patterns are made by the human observer. Our goal is to build a mathematical framework to infer dynamic migration models from the static eBird data. Quantitative migration models are of great scientific and practical import: for example, this problem originally arose out of an interdisciplinary project at Cornell University to model the possible spread of avian influenza in North America through wild bird migration.

The migratory behavior for a species of birds can be modeled as a generative process that independently governs how individual birds fly between locations. At any given time step, the eBird data reveal partial information about the status of the population. In particular, we assume that, from eBird, one can derive estimated counts $N_t(\ell)$ of the number of birds at location ℓ at time t, for all locations ℓ and times t (note that we deal with a finite set of locations and times, e.g., by quantizing space and time). Hence, we observe a sequence of snapshots, each showing the approximate spatial distribution of birds at a given time. However, a number of aspects of the migration process as a whole are unobserved:

1. *Transitions*. Because individual birds are indistinguishable, we cannot match observations at time t to those at time $t+1$. Hence we do not know the flights that occurred to shift the population between its spatial distribution at time t and its distribution at time $t+1$.
2. *Hidden state of individuals*. The transition model for individual birds de-

scribes their flights between locations. The model may include, in addition to location, a number of other aspects of a bird's state that affect its migratory behavior but cannot be observed, such as life-history status or energy reserves.

3. *True counts*. The problem of estimating the location counts $N_t(\ell)$ from eBird is itself a very challenging problem, so we must assume that these counts are inexact, and that they provide only approximate information about the true underlying counts.

Our primary goal will be to make inferences about these unobserved quantities given the eBird data and appropriate assumptions about models for generating the data. In what follows, we present the collective HMM framework generically. We will discuss its applicability to migration inference and other problems, including an example application in multiple target tracking.

4.1.2 Hidden Markov Models

A hidden Markov model (HMM) is a generative model for sequential data in which a sequence of states $X = (X_1, \ldots, X_T)$ is drawn from a Markov chain. The states are hidden from the observer, who sees instead a sequence of output symbols $Y = (Y_1, \ldots, Y_T)$, each depending probabilistically on the current state. We often describe this as an object transitioning from state to state, and emitting an output symbol in each state. For an observed output sequence $\mathbf{y} = (y_1, \ldots, y_T)$, the observer would like to reconstruct the most probable state sequence, i.e., to find the state sequence $\mathbf{x}$ maximizing $p(\mathbf{x} \mid \mathbf{y})$.

The Viterbi algorithm is a dynamic programming algorithm to solve this

reconstruction problem [55]; it is well suited to labeling or tagging a single sequence of data. For example, in part of speech tagging, one assumes that the words $Y_1, \ldots, Y_T$ in a sentence are generated in two steps: first, parts of speech $X_1, \ldots, X_T$ (e.g., *noun, verb, noun*), are generated from a language-based transition model. Then, for all t, the word Y_t is chosen at random from all words with part of speech X_t. Although this is a crude model for generating actual sentences, it is a powerful inference framework. The Viterbi algorithm determines the part of speech for each word in a sentence by reconstructing the most probable part-of-speech sequence given the word sequence. HMMs have been successfully applied in many areas including speech recognition [55], natural language processing [14], and biological sequencing [23].

4.1.3 Collective Hidden Markov Models

Collective HMMs generalize this framework to consider the case when many objects independently evolve and emit output symbols according to a hidden Markov model. However, the objects are identical in appearance, so the observer cannot distinguish between identical symbols emitted by different objects. The information available to the observer is then the *multiset* of symbols emitted at each time step, represented by counts that indicate the number of times each symbol appears. Suppose there are M objects in total. Let $X_1^{(m)}, \ldots, X_T^{(m)}$ be the state sequence of object m, and let $Y_1^{(m)}, \ldots, Y_T^{(m)}$ be the corresponding sequence of output symbols. Then the tth observation is the multiset $A_t = \{Y_t^{(1)}, \ldots, Y_t^{(M)}\}$, and the inference problem analogous to the Viterbi reconstruction problem is to find the collection of state-symbol sequences $\{X_t^{(m)}, Y_t^{(m)}\}_{m=1}^M$ to maximize $p(\{X_t^{(m)}, Y_t^{(m)}\}_{m=1}^M | A_1, \ldots, A_T)$. We call

this the *multiple path reconstruction problem* — the observer attempts to reconstruct a specific set of M sample paths from the HMM, given collective observations of the entire population. In what follows, it is usually more convenient to think directly in terms of symbol-counts instead of multisets: for symbol α, let $N_t(\alpha)$ be the number of times that α appears in A_t, i.e., the number of objects that emit α at time t.

Example: Multiple Target Tracking. Suppose M identical targets are tracked by a collection of sensors. The state of a target consists of the pair (ℓ, d), where ℓ is one of a finite number of locations, and d is one of a finite number of directions. A pre-defined Markov model gives the probability of transitioning between states. For example, the probability $p((\ell, d), (\ell', d))$ that the target moves from location ℓ to location ℓ' while continuing to traveling in direction d is nonzero only if ℓ' can be reached by moving in direction d from ℓ. Sensors placed at each location can detect the presence of a target but not its direction of travel: a target in state (ℓ, d) emits the output symbol ℓ. Moreover, sensors can accurately count the number of targets present, so the observed data are the counts $N_t(\ell)$ of the number of targets at location ℓ during time t, for all ℓ and t. Then the problem of reconstructing the most probable tracks for each target given these counts is an instance of the multiple path problem.

4.1.4 Problem Variants and Summary of Results

There is a rather direct analogy between the preceding target-tracking example and migration inference, if one considers birds to be "targets". However, the assumptions are slightly different. In the bird migration application, the true

population size M is large and unknown. Furthermore, for birds, we generally consider the location-specific counts $N_t(\ell)$ to be relative values instead of absolute counts, and subject to estimation error. Finally, we do not wish to make inferences about individual birds, but about portions of the population: e.g., "10% of the population flies from A to B during March."

These examples illustrate some of the different modeling assumptions one can make in applications of collective HMMs, and the choices impact the difficulty of the corresponding inference problem. We will consider several specific variants of the multiple path reconstruction problem that arise from particular assumptions about states and observations.

- *States: hidden vs. observed.* In hidden Markov models, the true state is never known to the observer, otherwise the reconstruction problem is trivial. In our model, it remains interesting to consider the case when states are observed, but objects are indistinguishable. In this case, the observer sees counts of *states* at each time step, instead of symbols. For example, suppose the target-tracking example is simplified so the state consists only of a target's location. Then the problem of reconstructing the tracks of individual targets remains challenging.
- *Counts: exact vs. noisy.* We have so far assumed that the observed quantity $N_t(\alpha)$ is the exact count of the number of objects that emit symbol α at time t, but it will be useful to consider the case when these counts are corrupted by noise to give only approximate counts. For a number of useful noise models, this does not significantly affect the difficulty of the inference problem.
- *Reconstruction: integer vs. fractional.* When the number of objects is large

and unknown, as in the bird example, it is more appropriate to consider counts to be relative values that indicate a fraction of a population, rather than integer values. In this case, we also seek a fractional reconstruction, and the inference problem becomes easier to solve.

Table 4.1: Summary of inference results for collective HMMs.

states	counts	reconstruction	algorithm	comment
hidden	*	integer	IP	NP-hard
hidden	*	fractional	LP	
observed	exact	*	transportation	decomposes by t
observed	noisy	*	flow	

Table 4.1 summarizes the algorithms for solving the reconstruction problems that arise from these different choices. All of the algorithms to be presented later in this chapter are based on an integer program (IP) formulation of a modified network flow problem. The problem as originally presented, with hidden states and an integer reconstruction, is NP-hard (first row), but a fractional reconstruction can be obtained in polynomial time by solving the linear program (LP) relaxation of the IP (second row). When states are observed, the problem becomes tractable for both fractional and integer reconstructions. With exact counts (third row), the reconstruction problem simplifies to a sequence of $T-1$ problems that are instances of the *transportation problem* [19], a variant of bipartite matching. When counts are noisy (fourth row), the reconstruction problem no longer decomposes by time step, but can be solved as a min-cost flow problem [18], where the edge-cost functions depend on the noise model. In Section 4.5.3, we will discuss a number of noise models that lead to convex cost func-

tions, in which case the noisy versions of each problem can be solved at little increase in computational complexity.

Structure of chapter. The remainder of the chapter is structured as follows. In section 4.2, we discuss related work. Section 4.3 introduces preliminary material. Sections 4.4 and 4.5 present hardness results and algorithms, respectively, for difference problem variants. Finally, section 4.6 presents experiments that evaluate collective HMMs on a synthetic target-tracking problem and demonstrate its use for migration inference.

4.2 Related Work

We briefly mention some related work. Caruana et al. [10] and Phillips et al. [54] used machine learning techniques to model bird distributions from observations and environmental features. For problems on sequential data, many variants of HMMs have been proposed [23], and recently, conditional random fields (CRFs) have become a popular alternative [45]. Roth and Yih [58] present an integer programming inference framework for CRFs that is similar to our problem formulations.

While developing the example target tracking application for this chapter, we discovered two closely related papers from the radar tracking community that were omitted from our conference publication [59]. Wolf et al. introduced a dynamic programming algorithm to find the best collection of M disjoint paths through a trellis graph for tracking multiple objects by radar [65]. The algorithm was exponential in M, but Castañon later observed that the same problem could

be solved in polynomial time by min-cost flow [11]. Although the premise of tracking a small number of objects in a continuous state space is substantially different from our premise of tracking the collective behavior of a large population in a discrete space, the model of Wolf et al. is equivalent to a collective HMM with non-hidden states. A continuous state space may be used because only states corresponding to observed data need be instantiated (see the discussion of pruning the trellis graph in Section 4.6.1); all state counts are then either zero or one because multiple objects never collide in exactly the same state. The "false alarm" observations in the Wolf et al. model that do not correspond to a real target can be modeled by noisy counts in our framework. The algorithm of Castañon is then equivalent to the min-cost flow algorithm that would be applied in our framework (fourth row of Table 4.1). The target tracking example that appears later in this chapter has an additional twist: part of a target's state (the direction) is unobserved, so that a collective HMM with hidden states is used.

Similar ideas have also appeared in the computer vision community. After our initial publication, Zhang et al. independently introduced a model very similar to that of Wolf et al. for tracking multiple objects in video sequences, again solved by min-cost flow [67]. Prior to that, Jiang et al. presented a linear programming approach for multiple object tracking that simultaneously found paths through M trellis graphs [41]. The work of Jiang et al. differs from collective HMMs and the Wolf et al. model in that objects are distinguishable, so observations for object m are matched against a template for that object, and that M copies of the trellis graph are instantiated, each consisting of the observations for a different (distinguishable) object.

4.3 Preliminaries

4.3.1 Models and Notation

A hidden Markov model $(\mathcal{I}, \Sigma, \pi, \sigma)$ is a Markov chain with state set $\mathcal{I}$ and transition probabilities $p(i,j) = \pi_{ij}$ for all $i, j \in \mathcal{I}$. We assume that $0 \in \mathcal{I}$ is a distinguished start state, so the initial distribution over states is given by π_{0i}. In each state, an output symbol from alphabet Σ is emitted. The probability that symbol α is emitted given that the current state is i is equal to the *emission probability* $p(\alpha|i) = \sigma_{i\alpha}$. Let $X = (X_1, \ldots, X_T)$ be a random sequence of states drawn from the HMM, and let $Y = (Y_1, \ldots, Y_T)$ be the corresponding observation sequence. Let $\mathbf{x} = (x_1, \ldots, x_T)$ and $\mathbf{y} = (y_1, \ldots, y_T)$ be a particular state sequence and observation sequence, respectively. We write $p(\mathbf{x}, \mathbf{y})$ for the joint probability $\Pr[X = \mathbf{x}, Y = \mathbf{y}]$. In general, we will adopt a similar convention: for a random variable A that takes value a, we write $p(a)$ to mean $\Pr[A = a]$ whenever this notation is unambiguous. Henceforth, assume that every state sequence $\mathbf{x}$ is augmented to begin with $x_0 = 0$, the distinguished start state. Then, by the Markov assumptions,

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^{T} p(x_{t-1}, x_t) p(y_t | x_t).$$

A collective HMM $(M, \mathcal{I}, \Sigma, \pi, \sigma)$ is an HMM with an additional parameter M indicating the number of objects in the population. Let $X^{(m)}$ denote the mth state sequence and let $Y^{(m)}$ denote the mth observation sequence. We represent a particular collection of samples by the pair of $M \times T$ matrices $\mathbf{X}$ and $\mathbf{Y}$ with rows $\mathbf{x}^{(m)}$ and $\mathbf{y}^{(m)}$ that represent the mth state sequence and observation

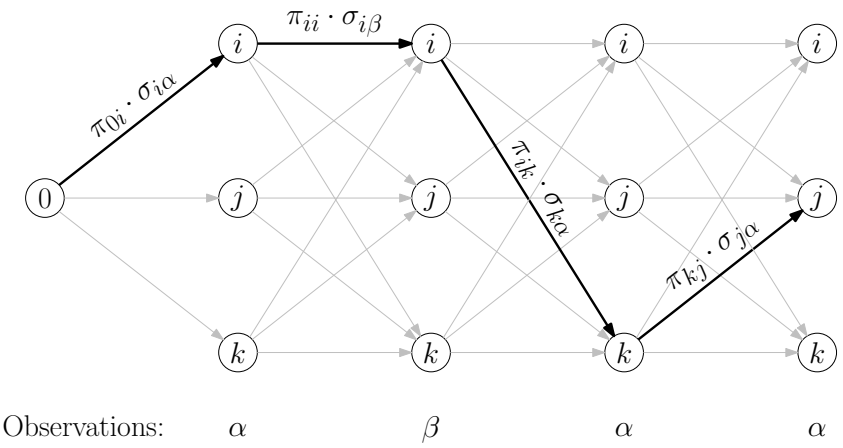


Figure 4.1: Trellis graph for HMM with states $\{0, i, j, k\}$ and alphabet $\{\alpha, \beta\}$, and the observed output sequence $\mathbf{y} = (\alpha, \beta, \alpha, \alpha)$. The path indicated in bold corresponds to sample path $\mathbf{x} = (0, i, i, k, j)$, and the joint probability is $p(\mathbf{x}, \mathbf{y}) = \pi_{0i}\sigma_{i\alpha} \times \pi_{ii}\sigma_{i\beta} \times \pi_{ik}\sigma_{k\alpha} \times \pi_{kj}\sigma_{j\alpha}$.

sequence, respectively. By independence,

$$p(\mathbf{X}, \mathbf{Y}) = \prod_{m=1}^{M} p(\mathbf{x}^{(m)}, \mathbf{y}^{(m)}) = \prod_{m=1}^{M} \prod_{t=1}^{T} p(x_{t-1}^{(m)}, x_t^{(m)}) p(y_t^{(m)} \mid x_t^{(m)}).$$

We will also consider arbitrary distributions λ over state-symbol sequences. When (X, Y) is distributed according to λ, we write $\Pr_\lambda[\cdot]$ and $E_\lambda[\cdot]$ to represent probabilities and expectations under this distribution.

4.3.2 The Trellis Graph and Viterbi as Shortest Path

To develop our flow-based algorithms, it is instructive to build upon a shortest-path interpretation of the Viterbi algorithm (e.g., see [58]). In the reconstruction problem for standard HMMs, we are given a model $(\mathcal{I}, \Sigma, \pi, \sigma)$ and observations $\mathbf{y} = (y_1, \ldots, y_T)$, and seek the state sequence $\mathbf{x}$ that maximizes $p(\mathbf{x} \mid \mathbf{y})$. The problem is conveniently illustrated using the *trellis graph* of the Markov model (see Figure 4.1). Here, nodes represent the states at each time step, and edges

connect a state at time $t-1$ to its possible successors at time t. A path through the trellis corresponds to a particular state sequence. For a known output sequence $\mathbf{y}$, the edge from state i to state j between times $t-1$ and t is labeled with probability $p(i,j)p(y_t \mid j)$, equaling the probability that an object makes the specified transition *and* emits the observed symbol. For any state sequence $\mathbf{x}$, the joint probability $p(\mathbf{x}, \mathbf{y})$ is given by the product of the edge probabilities for the corresponding path through the trellis.

The Viterbi algorithm solves the problem $\arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y}) = \arg\max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y})$, i.e., it finds a path of maximum probability through the trellis. We can equivalently work with costs instead of probabilities by making the transformation $c(\cdot) = -\log p(\cdot)$ for all relevant quantities. Then

$$c(\mathbf{x}, \mathbf{y}) = -\log p(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{T} (c(x_{t-1}, x_t) + c(y_t \mid x_t)).$$

Under this transformation, the Viterbi algorithm solves the problem $\arg\min_{\mathbf{x}} c(\mathbf{x}, \mathbf{y})$, where the cost $c(\mathbf{x}, \mathbf{y})$ is equal to the sum of the edge costs of the trellis path corresponding to $\mathbf{x}$. Thus the Viterbi algorithm finds the shortest path in the trellis graph.

4.4 The Multiple Path Reconstruction Problem

In collective HMMs, M independent state-symbol sequences $\{X^{(m)}, Y^{(m)}, m = 1, \ldots, M\}$ are drawn from a HMM, and the observations reveal the number of objects that emit the symbol α at time t, for all α and t. Let $\mathbf{N}$ be the $|\Sigma| \times T$ matrix containing the counts $N_t(\alpha) = \sum_m I\{Y_t^{(m)} = \alpha\}$. The multiple path reconstruction problem is to find $\mathbf{X}$ and $\mathbf{Y}$ to maximize $p(\mathbf{X}, \mathbf{Y} \mid \mathbf{N})$. Written

more explicitly, the problem is

$$\max_{\mathbf{X},\mathbf{Y}} \prod_{m=1}^{M} \prod_{t=1}^{T} p(x_{t-1}^{(m)}, x_t^{(m)})\, p(y_t^{(m)} \mid x_t^{(m)}) \qquad \text{subject to } N_t(\alpha) = \sum_{m=1}^{M} I\{y_t^{(m)} = \alpha\} \text{ for all } \alpha, t. \tag{4.1}$$

4.4.1 Reduction to Viterbi

A naive approach to the multiple path reconstruction problem is to reduce the collective HMM to a standard HMM on state set $\mathcal{I}^M \times \Sigma^M$, where state $\langle i^{(1)}, \ldots, i^{(M)}, \alpha^{(1)}, \ldots, \alpha^{(M)} \rangle$ encodes an entire tuple of states and output symbols from the original model, and the transition probabilities are given by the product of the element-wise transition probabilities, with emissions probabilities also rolled in:

$$p(\langle i^{(1)}, \ldots, i^{(M)}, \alpha^{(1)}, \ldots, \alpha^{(m)} \rangle, \langle j^{(1)}, \ldots, j^{(M)}, \beta^{(1)}, \ldots, \beta^{(m)} \rangle) = \prod_{m=1}^{M} p(i^{(m)}, j^{(m)}) p(\beta^{(m)} \mid j^{(m)})$$

A state from $\mathcal{I}^M \times \Sigma^M$ represents a pair of columns from the matrices $\mathbf{X}$ and $\mathbf{Y}$; hence a sequence of T states corresponds to complete matrices $\mathbf{X}$ and $\mathbf{Y}$ representing the entire collection of state-symbol sequences. Furthermore, it is easy to see that the probability of a state sequence in the new model is equal to the joint probability $p(\mathbf{X}, \mathbf{Y})$ of the entire collection of state-symbol sequences in the original model. To complete the reduction, we form a new alphabet $\hat{\Sigma}$ whose symbols represent multisets of size M on Σ. The emission probability in the new model, $p(A \mid \langle i^{(1)}, \ldots, i^{(M)}, \alpha^{(1)}, \ldots, \alpha^{(m)} \rangle)$, is equal to 1 if A is equal to the multiset $\{\alpha^{(1)}, \ldots, \alpha^{(m)}\}$, otherwise it is equal to zero. Then the solution to (4.1) can be found by running the Viterbi algorithm to find the most likely sequence

of states from $\mathcal{I}^M \times \Sigma^M$ that produce output symbols (multisets) $A_1, \ldots, A_T$. The running time is polynomial in $|\mathcal{I}^M \times \Sigma^M|$ and $|\hat{\Sigma}|$, but exponential in M.

4.4.2 Network Flow Formulation

Can we do better than the naive approach? Just as a single sample path corresponds to a path through the trellis, a collection of M paths corresponds to a flow of M units through the trellis — given M paths, we can route one unit along each to get a flow, and we can decompose any M-unit flow into M paths each carrying a single unit of flow. Thus we can write the optimization problem in (4.1) as the flow-based integer program that appears below. Let $c_{ij} = -\log \pi_{ij}$, and let $d_{i\alpha} = -\log \sigma_{i\alpha}$. The variable f_{ij}^t indicates the amount of flow traveling from i to j at time t; or, the number of objects that transition from state i to state j at time t. The variable $e_{i\alpha}^t$ represents the number of objects in state i that emit symbol α at time t.

Problem IP:

$$\min \sum_{i,j,t} c_{ij} f_{ij}^t + \sum_{i,\alpha,t} d_{i\alpha} e_{i\alpha}^t \tag{4.2}$$

$$\text{s.t.} \quad \sum_i f_{ij}^{t-1} = \sum_k f_{jk}^t \qquad \text{for all } j, t, \tag{4.3}$$

$$\sum_i f_{ij}^{t-1} = \sum_\alpha e_{j\alpha}^t \qquad \text{for all } j, t, \tag{4.4}$$

$$\sum_j e_{j\alpha}^t = N_t(\alpha) \qquad \text{for all } \alpha, t, \tag{4.5}$$

$$f_{ij}^t, e_{i\alpha}^t \geq 0, \text{ integer} \qquad \text{for all } i, j, \alpha, t.$$

The objective (4.2) arises by transforming the probabilities in (4.1) to costs

and rearranging the resulting sum to collect terms for identical transitions and emissions made in each time step. The flow conservation constraints (4.3) are standard: the flow entering j at time t must equal the flow leaving j at time t. The flow-output conservation constraints (4.4) ensure that the total number of symbols emitted by objects in state j at time t is correct — it must equal the incoming flow. The observation constraints (4.5) specify that the total number of times α is emitted at time t (by objects in any state) is equal to the observed count $N_t(\alpha)$. Without the observation constraints (4.5), IP could be massaged into an instance of the minimum-cost flow problem[2] [18], which is solvable in polynomial time by a variety of algorithms [31]. However, we cannot hope to encode the observation constraints into the flow framework, due to the following result.

Lemma 11. *The multiple path reconstruction problem is NP-hard.*

Proof. The proof is by reduction from SET COVER. An instance of SET COVER consists of a ground set $A = \{a_1, \ldots, a_n\}$ and a collection of subsets $A_1, \ldots, A_L \subseteq A$. The problem is to determine, for a given integer k, whether there exist k subsets $A_{\ell_1}, \ldots, A_{\ell_k}$ whose union is equal to A. If so, the subsets are said to *cover* A. For the reduction, let the states $\mathcal{I} = \{1, \ldots, L\}$ be indices of sets, and let the output symbols $\Sigma = A \cup \{0\}$ be the elements of the ground set plus an additional null symbol. Let the initial distribution over states be uniform, after which self-transitions have probability one and all other transitions have probability zero. Hence, every object selects a single state i, corresponding to set A_i, and remains there forever; the collection of sequences belonging to any k objects then corresponds to a selection of k sets. State i can output any symbol

[2]To do so, replace each trellis node by a gadget such that paths in the trellis encode state transitions *and* symbol emissions.

from the set A_i, or the symbol 0, each with equal probability. Finally, let the counts $\mathbf{N}$ be such that there are k objects in total, and at time t the symbol a_t is emitted once and the symbol 0 is emitted $k-1$ times, for $t = 1, \ldots, n$. These counts ensure that each element of A must be covered by the set corresponding to one of the k objects. Then, for any collection of k state sequences $\mathbf{X}$ and output sequences $\mathbf{Y}$, we have $p(\mathbf{X}, \mathbf{Y}) > 0$ if and only if the paths $\mathbf{X}$ correspond to a selection of subsets that cover A. □

A consequence of the proof is that it remains NP-hard to approximate the problem within any factor, because an algorithm need only find an outcome with nonzero probability to decide SET COVER. One may use a general purpose integer program solver to solve IP directly; our experiments show that this may be very efficient in some cases despite the lack of polynomial time performance guarantees. In the following sections we discuss alternatives with polynomial time guarantees.

4.5 Efficiently Solvable Problem Variants

4.5.1 Fractional Reconstruction

In applications like migration inference, where the true population size is large and unknown, it is more appropriate to consider population measurements to be relative values instead of absolute counts, and to make inferences about the behavioral patterns of *portions* of the population, instead of individuals. These are the assumptions behind the fractional reconstruction problem. Conceptually, the problem arises by letting $M \to \infty$ in the multiple paths problem. Op-

erationally, it is modeled by dropping the integrality constraints in IP. We also normalize the input values, letting $q_t(\alpha) = N_t(\alpha)/M$ specify the *fraction* of paths that output α at time t. The resulting LP is shown below; all variables now represent fractions of the total population, instead of absolute counts.

Problem RELAX:

$$\begin{aligned}
\min \sum_{i,j,t} c_{ij} f_{ij}^t + \sum_{i,\alpha,t} d_{i\alpha} e_{i\alpha}^t & \\
\text{s.t.} \quad \sum_i f_{ij}^{t-1} = \sum_k f_{jk}^t & \qquad \text{for all } j,t, \\
\sum_i f_{ij}^{t-1} = \sum_\alpha e_{j\alpha}^t & \qquad \text{for all } j,t, \\
\sum_j e_{j\alpha}^t = q_t(\alpha) & \qquad \text{for all } \alpha,t, \\
f_{ij}^t, e_{i\alpha}^t \geq 0 & \qquad \text{for all } i,j,\alpha,t.
\end{aligned}$$

The relaxed problem has the following interpretation. A feasible solution to RELAX is a single-unit flow, which can be decomposed over unique state-symbol sequences such that $\pi(\mathbf{x}, \mathbf{y})$ fractional units of flow follow the trellis path corresponding to state sequence $\mathbf{x}$ *and* emit output symbols $\mathbf{y}$. Hence, the observer chooses $\pi(\mathbf{x}, \mathbf{y})$ fractional units of each possible state-symbol sequence $(\mathbf{x}, \mathbf{y})$, totaling one unit, such that $q_t(\alpha)$ units emit symbol α at time t. Put another way, π is a distribution over state-symbol sequences (X, Y), and π satisfies $\Pr_\pi [Y_t = \alpha] = q_t(\alpha)$, that is, q_t specifies the marginal distribution over symbols at time t. The objective is to minimize $\sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{I}^T \times \Sigma^T} \pi(\mathbf{x}, \mathbf{y}) c(\mathbf{x}, \mathbf{y})$, or, equivalently, to maximize $E_\pi [\log p(X, Y)]$. Hence, we seek the distribution π with the specified marginal distributions that maximizes the expected log-probability, under the HMM probability measure $p(\cdot)$, of a random state-symbol sequence (X, Y)

drawn from π.

This formulation is similar in concept to maximum entropy or minimum cross entropy modeling, but the details are slightly different: such a model would typically find the distribution π with the correct marginals that minimizes the cross entropy or Kullback-Leibler divergence [60] between p and π, which, after removing a constant term, translates to maximizing $E_p[\log \pi(X, Y)]$.

4.5.2 Non-Hidden States

A useful simplification to collective HMMs is to dispense with output symbols and allow states to be observed directly. In this case, the counts $N_t(i)$ specify the number of objects in state i at time t, for all i and t. For example, we can use such a model for the bird migration problem when we do not wish to model the transitions as depending on any hidden aspects of the state of a bird, such as its migratory direction or energy reserves. The reconstruction problem in this case is $\arg\max_{\mathbf{X}} p(\mathbf{X} \mid \mathbf{N})$, and the integer program simplifies by eliminating the e variables.

Problem OBS:

$$\min \sum_{i,j,t} c_{ij} f_{ij}^t \tag{4.6}$$

$$\text{s.t.} \quad \sum_i f_{ij}^{t-1} = \sum_k f_{jk}^t \qquad \text{for all } j,t, \tag{4.7}$$

$$\sum_i f_{ij}^{t-1} = N_t(j) \qquad \text{for all } j,t, \tag{4.8}$$

$$f_{ij}^t \geq 0, \text{ integer} \qquad \text{for all } i,j,t.$$

OBS is an instance of the min-cost flow problem, for which a variety of efficient algorithms exist [18,31], or, one may use a general purpose LP solver; any basic solution to the LP relaxation is guaranteed to be integral [18].

Exact Counts. When the counts are exact as we have assumed thus far, the problem OBS simplifies even further, because the observation constraints (4.8) specify the exact flow through each node in the trellis. Hence, the flow conservation constraints (4.7) can be replaced by the following equivalent constraints:

$$N_t(j) = \sum_k f_{jk}^t \quad \text{for all } j,t. \tag{4.7$'$}$$

In the new constraint set for OBS, each constraint refers only to variables f_{ij}^t for a single time step t. Hence, the problem can be decomposed into $T-1$ disjoint subproblems for $t = 1, \ldots, T-1$. The tth subproblem is:

Problem SP$_t$:

$$\begin{aligned}
&\min \sum_{i,j} c_{ij} f_{ij}^t \\
&\text{s.t.} \quad \sum_i f_{ij}^t = N_{t+1}(j) && \forall j, && (4.7') \\
&\qquad \sum_j f_{ij}^t = N_t(i) && \forall i, && (4.8) \\
&\qquad\quad f_{ij}^t \geq 0, \text{ integer} && \forall i, j.
\end{aligned}$$

Problem SP$_t$ is illustrated on the trellis in Figure 4.2. State i at time t has a supply of $N_t(i)$ units of flow coming from the previous step, and we must route $N_{t+1}(j)$ units of flow to state j at time $t+1$, so we place a demand of $N_{t+1}(j)$ at the corresponding node. Then the problem reduces to finding a minimum cost matching of the supply from time t to meet the demand at time $t+1$, solved separately for all $t = 1, \ldots, T-1$. This is a variant of bipartite matching called the transportation problem [19], and a special case of min-cost flow.

4.5.3 Noisy Counts

In cases when the counts $N_t(\cdot)$ are themselves estimates, it is useful to allow the solution to deviate somewhat from these counts to find a better overall reconstruction. To model this, we assume that the true counts $N_t(\cdot)$ are *unobserved*, and instead, a corrupted count $Z_t(\cdot)$ is available. We furthermore assume that the corruption process operates independently on each count, so the variables $Z_t(\cdot)$ are conditionally independent given $\mathbf{N}$, and that the conditional distribution of $Z_t(\alpha)$ given $N_t(\alpha)$ is specified by the density $p(z \mid n)$. The reconstruction problem then becomes $\arg\max_{\mathbf{X},\mathbf{Y}} p(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$ (in the case of hidden states), or

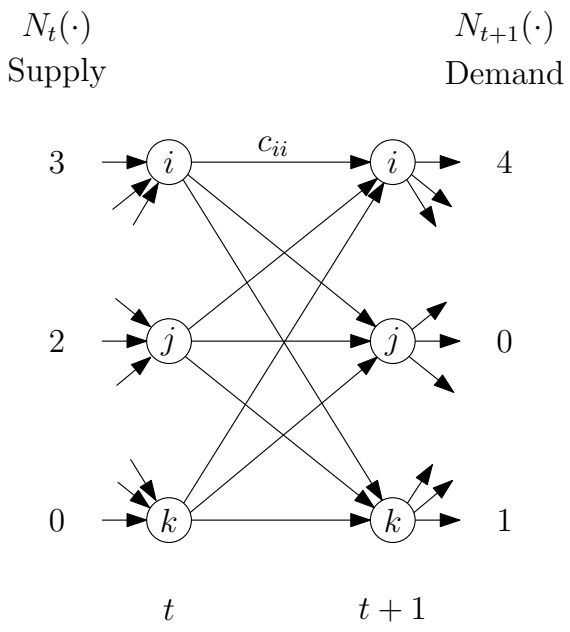


Figure 4.2: Illustration of subproblem on the trellis, with example supply and demand values for $M = 5$.

$\arg\max_{\mathbf{X}} p(\mathbf{X} \mid \mathbf{Z})$ (in the case of observed states).

The modification of either problem is similar; to be concrete, we focus on the case of observed states. Because $\mathbf{Z}$ is now observed in place of $\mathbf{N}$, the problem no longer has hard constraints on state counts; instead, an additional likelihood term appears in the joint probability:

$$p(\mathbf{X}, \mathbf{Z}) = p(\mathbf{X})p(\mathbf{Z} \mid \mathbf{X}) = p(\mathbf{X}) \prod_{t,i} p\big(Z_t(i) \mid N_t(i)\big).$$

Here, we recall that $N_t(i) = \sum_m I\{x_t^{(m)} = i\}$ is a deterministic function of $\mathbf{X}$. The problem OBS from the previous section is modified by dropping the observation constraints (4.8) and adding the negative log-likelihood term $-\log p(\mathbf{Z} \mid \mathbf{X})$ in the objective. Let $\ell(n; z) = -\log p(z \mid n)$. Then the modified problem is

Problem OBS+NOISE:

$$\begin{aligned}
\min \sum_{i,j,t} c_{ij} f_{ij}^t + \sum_{t,i} \ell(n_i^t; Z_t(i)) & \\
\text{s.t.} \quad \sum_i f_{ij}^{t-1} = n_j^t & \qquad \text{for all } j, t, \\
\sum_k f_{jk}^t = n_j^t & \qquad \text{for all } j, t, \\
\sum_i n_i^T = M & \qquad (4.9) \\
f_{ij}^t, n_i^t \in \mathbb{N} & \qquad \text{for all } i, j, t.
\end{aligned}$$

We have split the flow conservation constraints and introduced the variable n_j^t to represent the true unobserved count for state j at time t. For linear problems, these variables can be eliminated to obtain a more compact formulation. However, when ℓ is nonlinear, this formulation is preferable because the objective function is *separable*: it is a sum of terms where each nonlinear term is a function of only one variable [20]. We have added the constraint (4.9) to ensure that a total of M units of flow travel through the trellis; this constraint was implicit in previous formulations, due to the fact that the observations constraints required M output symbols at each time step.

The OBS+NOISE formulation corresponds to a min-cost flow problem in a modified trellis where each node is split into two nodes connected by a single edge, as illustrated in Figure 4.3. The amount of flow on the new edge is n_i^t, and the cost function of the edge is $\ell(\cdot; Z_t(i))$.

Convexity. The convexity of the negative log-likelihood function $\ell(n; z)$ is an important consideration. This is the cost function applied to the n_i^t variable, and it determines the convexity of the modified optimization problem. Each of

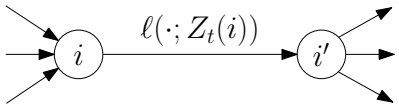


Figure 4.3: Illustration of the gadget that replaces each trellis node in the flow problem represented by OBS+NOISE.

the problems we have discussed incurs only a slight increase in computational complexity when noisy counts are incorporated in the case that $\ell(n; z)$ is convex. We comment briefly on each in turn:

(i) *Hidden states, integer reconstruction*. When states are hidden, the integer problem remains NP-hard. To solve the problem as an integer program, the function ℓ may be replaced by a piecewise-linear function $\hat{\ell}$ that matches ℓ on integer values.

(ii) *Hidden states, fractional reconstruction*. In this case, the modified problem is a convex optimization problem with linear constraints and a separable convex objective. One approach to solve such a problem is to approximate the convex objective with a piecewise-linear objective and solve the resulting linear program (e.g., see Section 24-3 of [20]). An optimal solution of arbitrary precision can be obtained by solving a sequence of polynomially many piecewise-linear approximations [37].

(iii) *Observed states, integer reconstruction*. In this case, the problem is a flow problem with convex edge costs, for which there are specialized strongly polynomial time algorithms [43]. To ensure an integer reconstruction, the objective term ℓ should be replaced by the piecewise linear version $\hat{\ell}$ described in case (i).

(iv) *Observed states, fractional reconstruction*. This case is similar to case (iii), except the original objective ℓ is used instead of the piecewise linear version

$\hat{\ell}$ because fractional variables are allowed.

Log-Concave Noise Models. It remains to explore specific noise models that result in a convex objective. A function $f(n)$ is said to be *log-concave* if $\log f(n)$ is concave. In our case, a likelihood function $p(z \mid n)$ that is log-concave in n is the essential property for efficient inference, because it coincides with a convex negative log-likelihood function $\ell(n; z)$. A similar situation arises in maximum likelihood estimation for generalized linear models, and, in that context, many useful log-concave models have been explored (see, for example, [21]). In this section we will discuss three specific noise models with log-concave likelihoods that are relevant to collective HMMs, and then outline a general situation that yields log-concave likelihoods: when n is the natural parameter of a single-parameter exponential family of distributions.

Examples. For clarity of notation, fix i and t and let let $Z = Z_t(i)$ be the noisy version of the true count $N = N_t(i)$.

- *Additive Laplacian Noise.* Suppose that $Z = N + \epsilon$ where $\epsilon \sim \text{Laplace}(0, b)$, so the conditional density and corresponding negative log-likelihood functions are

$$p(z \mid n) = \frac{1}{2b} \exp\left(-\frac{|z - n|}{b}\right), \quad \ell(n; z) = \log 2b + \frac{|z - n|}{b}.$$

  We see that $p(z \mid n)$ is log-concave because $\ell(n; z)$ is convex in n. Laplacian noise is especially practical because $\ell(n; z)$ is piecewise linear in this case, and hence the optimization problems can be written as linear optimization problems for added efficiency.[3]

[3]The support of $p(z \mid n)$ may be restricted to $[0, \infty)$ to reflect the typical case that the observa-

- *Additive Gaussian Noise.* Perhaps the most common noise model is additive Gaussian noise. Suppose $Z \sim \text{Normal}(N, \sigma^2)$. Then

$$p(z \mid n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z-n)^2}{2\sigma^2}\right), \quad \ell(n; z) = \frac{1}{2}\log 2\pi\sigma^2 + \frac{(z-n)^2}{2\sigma^2}.$$

  Wee see that $\ell(n; z)$ is again convex in n.

- *Poisson counts.* Consider the following model of the eBird counts, in which the number of birds Z counted by an observer has distribution Poisson(αn). This corresponds to a model where each bird in the population is observed independently according to a Poisson process with rate α. Then

$$p(z \mid n) = \frac{(\alpha n)^z \exp(-\alpha n)}{z!}, \quad \ell(n; z) = \log z! - z \log \alpha n + \alpha n.$$

  It is easy to see by differentiating $\ell(n; z)$ twice that this model is also log-concave.

Exponential Families. The log-concavity of the Gaussian noise model is a special case of a more general result. We say that a family of distributions is a *single-parameter exponential family* with natural parameter n if the density can be written as $p(z \mid n) = h(z) \exp(n\,T(z) - A(n))$ for some functions $h(\cdot)$, $T(\cdot)$ and $A(\cdot)$. Many of the most common distributions form an exponential family with respect to some parameter (e.g., the unknown mean of a Gaussian distribution), and each has a natural parameterization that allows it to be written in the form above.

Lemma 12. *Suppose $p(z \mid n)$ is the density function of a single-parameter exponential family with natural parameter n. Then $p(z \mid n)$ is log-concave in n.*

tion Z must be non-negative. However, this change affects only the normalization term $\log 2b$ in $\ell(n; z)$. Since that term is constant with respect to n, we may ignore it during optimization.

Proof. In this case, $\ell(n; z) = -\log h(z) - nT(z) + A(n)$, and $\ell''(n; z) = A''(n)$. Let Z be a random variable from this distribution. In a naturally parameterized exponential family, $\text{Var}(T(Z)) = A''(n)$ (see, for example, [5]). Hence $A''(n) \geq 0$, so $\ell(n; z)$ is convex. □

Lemma 12 implies that a multitude of other models are log-concave, including those based on the binomial, Gamma, and negative binomial distributions. It is important to note that Lemma 12 only applies directly to our situation when n is the *natural* parameter of the exponential family. In some cases, this is quite sensible: for example, in the Gaussian example, the natural parameter is the mean. In other cases, the natural parameter may be less appropriate. For example, the Poisson distribution with unknown rate λ forms an exponential family with natural parameter $\log \lambda$, so the fact that the Poisson example we discussed is log-concave with respect to $n = \lambda/\alpha$ does *not* follow directly from Lemma 12. However, the parameterization in the example is a better match for our application, and, in this case, it is also log-concave.

Repeat Counts. Any log-concave noise model can be extended to the case of multiple independent counts. For example, in the eBird problem, it is often the case that many different observers submit counts for a given location and time period; these all depend probabilistically on the same underlying count. Let $Z^{(1)}, \ldots, Z^{(K)}$ be drawn independently from $p(z \mid n)$ with unknown n. By independence, $p(z^{(1)}, \ldots, z^{(K)} \mid n) = \prod_{k=1}^{K} p(z^{(k)} \mid n)$, so we have $\ell(n; z^{(1)}, \ldots, z^{(K)}) = \sum_{k=1}^{K} \ell(n; z^{(k)})$. Hence, for each unknown count N, the likelihood term in the objective is a sum of the individual likelihood terms $\ell(n; z^{(k)})$, and this is convex in n because each member of the sum is convex.

Fractional Counts. Although we have called the unknown value n a count, suggesting that it is integer-valued, the preceding discussion of noisy counts pertains equally well to the case encountered in the fractional reconstruction problem, when the unknown value q is a real number.

4.6 Experiments

In this section, we conduct several experiments with collective HMMs. We evaluate the performance of the integer reconstruction problem on a synthetic target tracking application for which ground truth is known, and then we demonstrate the use of collective HMMs to build visualizations using eBird data for the migration of Ruby-Throated Hummingbird. First, we turn to a practical consideration to reduce the size of the inference problems.

4.6.1 Pruning the Trellis

The optimization problems we have discussed scale with the size of the trellis graph, which scales quadratically with the number of states; this can pose difficulties for large state spaces. However, for smaller values of M, it is often the case that only a tiny fraction of the HMM state space is occupied. When the problem has an appropriate sparsity structure, we can do substantial pruning of the trellis graph based on the observations before solving the optimization problem.

For example, if a symbol α is not emitted at time t, the emission variables $e^t_{i\alpha}$ must all be zero, so they can be eliminated. Similarly, state i can be eliminated

at time t if it cannot produce any symbol observed at time t, or if it is not reachable from an allowed state at time $t-1$. Call a state sequence x *feasible* if each transition has nonzero probability and, for all t, state x_t can emit *some* symbol α that is observed at time t with nonzero probability. Below, we outline a method to prune the trellis so that it consists only of nodes that lie on some feasible path.

First, for $S \subseteq \mathcal{I}$ and $B \subseteq |\Sigma|$, define $\pi(S)$ to be the set of states that can be reached by a transition of nonzero probability from S, and define $\pi^{-1}(S)$ be the set of states with transitions of nonzero probability *to* S. Similarly, let $\sigma(S)$ be the symbols that can be emitted by some state in S and let $\sigma^{-1}(B)$ be the states that can emit some symbol in B. Let B_t be the set of symbols observed at time t, or, in the case of noisy counts, the set of symbols that *may* be emitted at time t (i.e., $B_t = \{\alpha : \Pr[N_t(\alpha) > 0 \mid Z_t(\alpha)] > 0\}$).

To begin, eliminate all emission variables $e^t_{i\alpha}$ where $\alpha \notin B_t$. Next, build a set of active states A_t for each time t as follows. Initialize $A_0 = \{0\}$, and $A_t = \sigma^{-1}(B_t)$ for all $t \geq 1$. This ensures that each active state can emit an observed output symbol. In a forward pass through the trellis, set $A_t = A_t \cap \pi(A_{t-1})$ for $t = 1, \ldots, T$. Next, in a reverse pass, set $A_t = A_t \cap \pi^{-1}(A_{t+1})$ for $t = T-1, \ldots, 1$. These two passes ensure that each retained state lies on a path through active states from the beginning to the end of the trellis. In the final problem, instantiate only flow variables f^t_{ij} such that $i \in A_t$ and $j \in A_{t+1}$.

4.6.2 Target Tracking

Our first example is a synthetic target tracking application. Unlike the example given earlier, we consider a more realistic scenario where the *true* target dynam-

ics are continuous and deterministic based on physical quantities. For inference, a discrete model is used, with stochastic transitions used to accommodate errors in the modeling process that arise from measurement error, quantization error, or uncertainty as to the true initial state or dynamics of targets.

Target Dynamics. Figure 4.4 shows examples of the true target dynamics. We assume that targets are unit velocity and that measurements occur once per second. Targets follow curved paths within an $L \times L$ bounding box and reverse direction at boundaries. The state $(x, y, \theta, d\theta)$ of a target consists of its location (x, y), its direction of travel θ, and the rate of change in direction $d\theta$. The initial target positions and directions are chosen uniformly at random, and $d\theta$ is chosen uniformly in the interval $[-\frac{\pi}{30}, \frac{\pi}{30}]$, to limit maximum curvature.

The continuous track of a target is approximated by a discrete track with F locations per second (so a measurement occurs every F locations on the discrete track). Let $dx = \cos(\theta)$, $dy = \sin(\theta)$. The update rule is given by

$$x = x + \frac{dx}{F}, \quad y = y + \frac{dy}{F}, \quad \theta = \theta + \frac{d\theta}{F}$$

If a target crosses a boundary, its position is reflected across the boundary and the directional component orthogonal to that boundary is reversed.

Observations. For the purposes of observation, the bounding box is divided into an $L \times L$ grid, and a sensor at the center of each cell counts the number of targets inside that cell. We assume that measurements detect all targets within the cell at the instant of measurement.

Transition Model. The transition model approximates the continuous dynamics with a Markov model. The state (ℓ_1, ℓ_2, ρ) consists of the discrete grid location $(\ell_1, \ell_2) \in \{1, \ldots, L\}^2$, and a discrete direction ρ among D evenly spaced directions. The transition probabilities may be estimated using example tracks, but here we specify the model directly. Given the current state, the direction and location of the next state are chosen independently. The target continues in the same direction with probability $1 - \beta_1$, and for each of the two adjacent directions, it adopts that direction with probability $\beta_1/2$. We used $D = 8$ in all our experiments, so the two adjacent directions cover a wide enough angle to accommodate the maximum curvature of the true target dynamics.

The new grid cell is chosen as follows. With probability $1 - \beta_2$, the target moves assuming the true location in continuous space is uniformly distributed within the current grid cell, and that the target moves exactly 1 unit in direction ρ. In this case, it may reach up to four cells depending on the actual starting location within the cell and the direction. Let $dx = \cos(\rho)$ and $dy = \sin(\rho)$. Then, with probability $(1 - |dx|)(1 - |dy|)$ the target stays in the current cell; with probability $(1 - |dx|)|dy|$ it moves in the y direction only; with probability $|dx|(1 - |dy|)$ it moves in the x direction only; and with probability $|dx||dy|$, it moves in both directions. With the remaining probability of β_2, the next cell is chosen uniformly from the 6 grid cells in a "forward" direction: draw a line through the center of the current cell orthogonal to ρ, and take any cell in the positive ρ direction. These transitions account for curved trajectories that reach a cell that is not possible under the straight line assumption. The parameters used in these experiments were $L = 20$, $D = 8$, and $\beta_1 = \beta_2 = .2$.

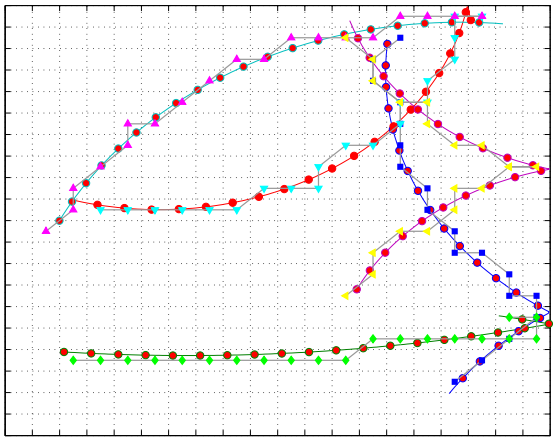
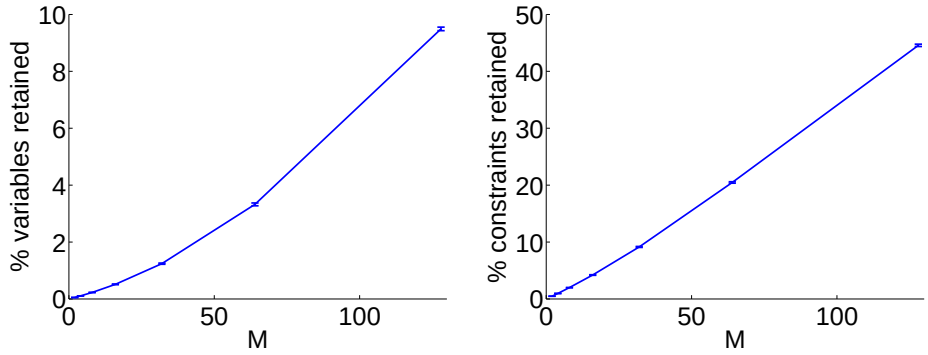


Figure 4.4: Example track reconstruction. True target paths are curved, with red circular markers indicating observation locations. Each reconstructed track is identified by a different marker type, placed at the center of the grid cells on the path.

Results. We ran 10 trials each for increasing numbers of targets ($M = 1, 2, 4, \ldots, 128$) and observation sequences of length $T = 20$. Each time, the initial target states were chosen randomly. We pruned the trellis graph as described in section 4.6.1, and then solved the resulting integer program with the CPLEX mixed integer optimizer. Figure 4.4 shows an example of four reconstructed tracks using this method. Before pruning, the problem has 1.02M variables and 72,000 constraints (these numbers depend only on the transition and emission model, not the targets, and they exclude any transitions or emissions of zero probability). Figure 4.5 shows the reduced problem size for different values of M. For this example, all problems are greatly reduced in size. In each case, the problem contains less than 10% of the possible variables, and the size grows modestly in M, showing only a slight super-linear trend.

Figure 4.6 (a) shows the reconstruction error for different values of M. The error metric is calculated on the flows f_1 and f_2 that are induced on grid cells by the true paths and the reconstructed paths, respectively. If $f_1 = f_2$, the recon-

struction can be decomposed into grid-cell paths that exactly match the grid-cell paths of the true tracks. The error metric is equal to $|f_1 - f_2|/(2M(T-1))$, the fraction of transitions per time step that must be swapped to obtain f_1 from f_2. The reconstruction quality is very good: 95% or better for 50 or fewer targets. Error increases approximately linearly as the number targets grows; as this happens, targets collide in grid cells more frequently so there are more unique reconstructions that explain the data well. Figure 4.6 (b) shows the running time per variable as the number of targets grows. It is approximately constant for $M \leq 64$, suggesting that running time grows linearly in the size of the pruned problem for sparsely occupied state spaces. Running time jumps considerably for $M = 128$. We noted that for smaller problem sizes, CPLEX takes some time to find an integer feasible solution, but proves optimality quickly once it does. It may be possible to speed up the solution process by seeding the optimization routine with an integer solution constructed via a heuristic method.



(a) (b)

Figure 4.5: Problem size growth with number of targets: (a) percent of variables retained (of 1.02×10^6) after pruning, (b) percent constraints retained (of $72,000$).

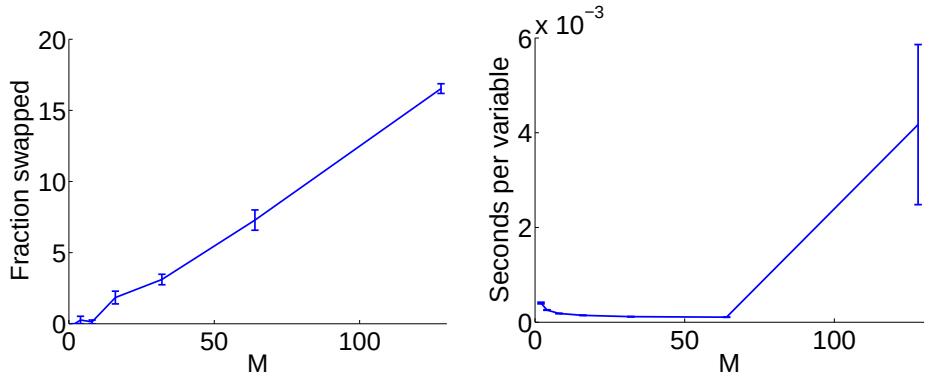


Figure 4.6: Results as number of targets grows: (a) fraction of correctly labeled transitions, (b) running time per variable.

4.6.3 eBird

In this section, we demonstrate the use of fractional reconstruction problem to visualize likely migration routes of the Ruby-throated Hummingbird (*Archilochus colubris*). This species was chosen because it is relatively common and exhibits a consistent migration pattern in the eastern United States, a region that is relatively well covered by eBird observations. As with most songbird species, there is almost no quantitative data about the migration of the Ruby-Throated Hummingbird. Hence the evaluation in this section is necessarily qualitative.

Background. Launched in 2002, eBird is a *citizen science* project run by the Cornell Lab of Ornithology. On the eBird website, bird-watchers submit checklists of birds they observe, indicating a count for each species, along with the location, date, time and additional information. Our data set consists of the 428,648 *complete* checklists from 1995 through 2007, meaning the reporter listed all species observed. This means we can infer a count of zero, or a *negative ob-*

servation, for any species not listed. Using a USGS land cover map, we divide North America into grid cells that are roughly 225 km on a side. All years of data are aggregated into one, and the year is divided into weeks so $t = 1, \ldots, 52$ represents the week of the year.

Observations. For this problem, the observations are estimates $z_t(i)$ of the relative number of birds in each grid cell during each week. Producing these estimates is a difficult problem, but not our primary focus. In these experiments, we use a relatively simple method detailed in our conference publication [59] based on *harmonic energy minimization* [68]. Since the actual eBird observations are highly non-uniform in space and time, this technique performs smoothing on a space-time lattice to infer values for locations with few or no observations by "borrowing" observations from nearby points in space and time. Improved distribution modeling, especially by regression-based methods that incorporate environmental variables (e.g., []), is likely to greatly improve the overall quality of these visualizations.

The smoothing method also yields a confidence heuristic $\gamma_t(i)$ for each estimated relative cell count $z_t(i)$; lower values indicate a more confident estimate that is based either on direct observations, or observations that were made nearby in space or time. The confidence heuristic is used as the parameter of a Laplace noise model:

$$p(z_t(i) \mid q_t(i)) \propto \exp\left(\frac{|z_t(i) - q_t(i)|}{\gamma_t(i)}\right)$$

This typical noise magnitude is thus greater for less confident estimates, and it is less costly for the optimization procedure to deviate from these estimates.

Transition Model. It remains to specify the Markov model. We use a simple Gaussian model favoring short flights, letting $p(i,j) \propto \exp(-d(i,j)^2/\sigma^2)$, where $d(i,j)$ measures the distance between grid cell centers. This corresponds to a squared distance cost function on flights. To reduce problem size, we set $p(i,j)$ to zero when when $d(i,j) > 1350$ km. We also found it useful to impose the upper bound $q_t(i) \leq 2z_t(i)$, so the inferred value cannot the exceed the estimate by more than a factor of 2; this is equivalent to restricting the support of the noise model $p(z \mid q)$. The final flow problem, which was solved using the MOSEK optimization toolbox, had $78{,}521$ constraints and $3{,}031{,}116$ variables.

Results. Figure 4.7 displays the migration paths our model inferred for the four weeks starting on the dates indicated. The left column shows the distribution and paths inferred by the model; presence of birds is indicated by red shaded grid cells, with darker shades reflecting higher density (i.e., higher values for the inferred quantity $q_t(i)$). Arrows indicate flight paths (f_{ij}^t) between the week shown and the following week, with line width in proportion to f_{ij}^t. In the right column, the raw data is given for comparison. Yellow dots indicate negative observations; blue squares indicate positive observations, with size proportional to count. Locations with both positive and negative observations appear a charcoal color. The general pattern of the inferred distributions and migration paths matches qualitative descriptions of Ruby-Throated Hummingbird migration (e.g., see [56]). It is an open problem to evaluate this approach quantitatively.

Poisson Model. We also implemented a Poisson observation model that does not do a separate estimation step for the cell counts. Instead, the eBird counts

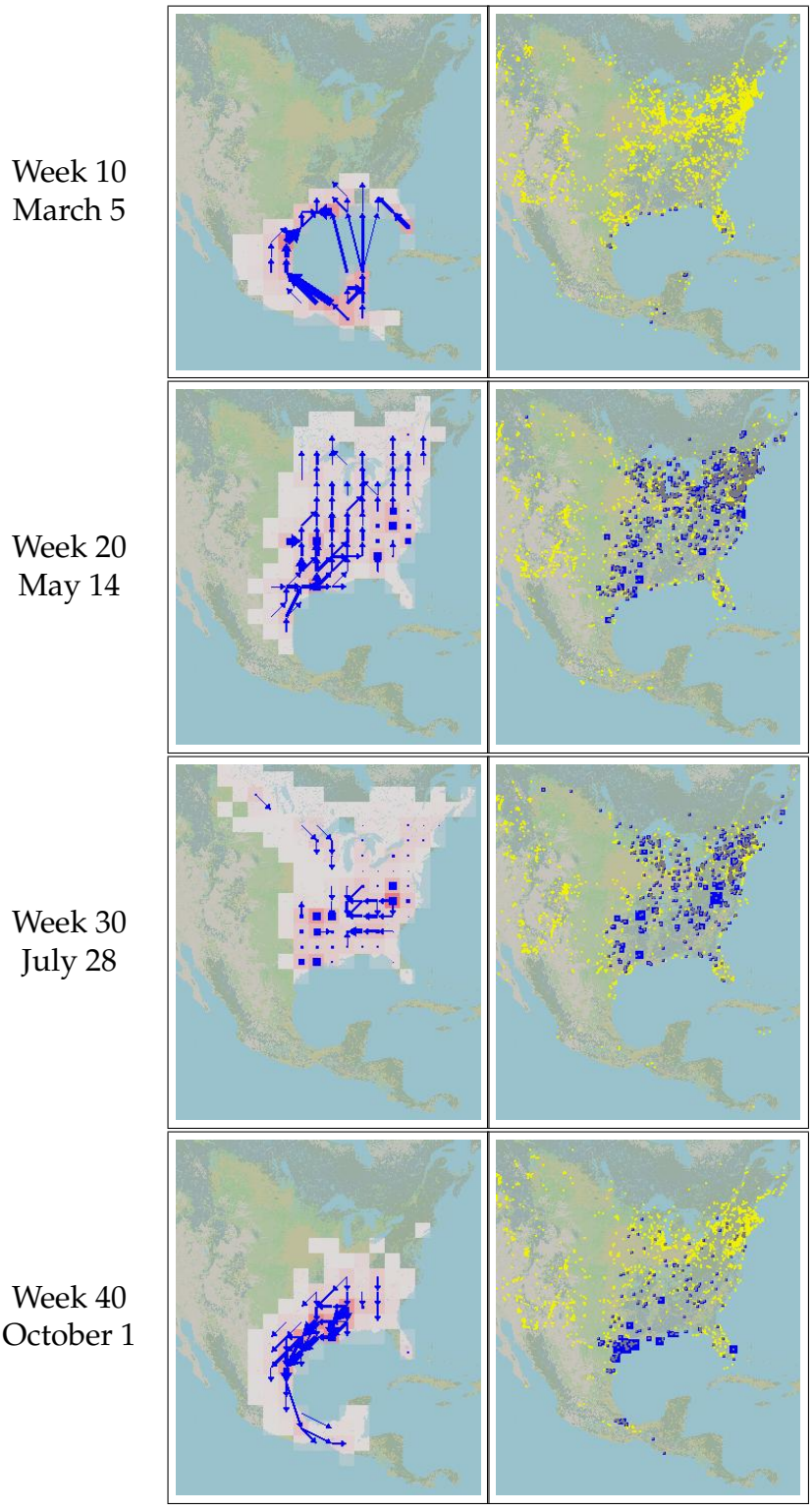


Figure 4.7: Ruby-throated Hummingbird migration. See text for description.

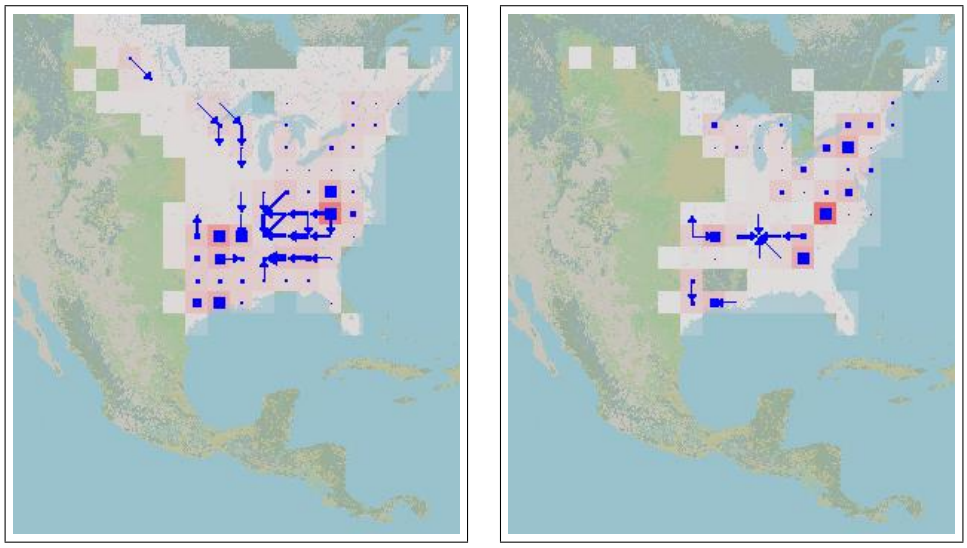


Figure 4.8: Comparison of Poisson model versus standard model. Left: week 30 transitions for standard model. Right: week 30 transitions for Poisson model, $\alpha = 8$.

are incorporated directly into the model as noisy observations that depend on the true underlying density. Specifically, we model each eBird count as an independent Poisson(αq) random variable, where q is the fraction of the population present in the grid cell where the individual went bird-watching.

This model is substantially simpler and more elegant than the previous version because it incorporates cell count estimation directly into the migration inference procedure. The Poisson likelihood function automatically controls for different confidence levels when estimates are based on more or less data: the likelihood is always maximized when αq is equal to the average of the counts conducted in that cell, but the penalty for deviating from the maximum is much greater for cells with many counts. However, the method also suffers somewhat in comparison with the previous approach because no smoothing is performed. Some cells have few or no observations, and hence the likelihood function is indifferent to the inferred count for those cells. It is a promising area of future

research to combine elements of these two models.

Qualitatively, we observed that the Poisson model produces similar patterns to the standard model, but with less apparent "jitter", where birds are seen making significant flights during non-migratory seasons to accommodate fluctuating cell counts. Figure 4.8 compares the transitions of the standard model (left) and the Poisson model (right, $\alpha = 8$) for week 30, during the non-migratory season. Self-transitions are marked by squares instead of arrows; these indicate that the birds stay in their current location. This week is a typical comparison: the Poisson model has fewer transitions that are known to be spurious, but predicts a spatial distribution that is less smooth than the standard model.

BIBLIOGRAPHY

[1] David Aldous and James Fill. Reversible Markov chains and random walks on graphs. `http://www.stat.berkeley.edu/users/aldous/RWG/book.html`, 2009 (last accessed December, 2009).

[2] K. Avrachenkov and N. Litvak. The Effect of New Links on Google Pagerank. *Stochastic Models*, 22(2):319–331, 2006.

[3] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte Carlo methods in PageRank computation: When one iteration is sufficient. Memorandum 1754, University of Twente, The Netherlands, 2005.

[4] M. Bianchini, M. Gori, and F. Scarselli. Inside PageRank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.

[5] Peter J. Bickel and Kjell A. Doksum. *Mathematical statistics: basic ideas and selected topics*. Prentice Hall, San Francisco, 1977.

[6] Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.

[7] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[8] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. *Computer Networks*, 33(1-6):309–320, 2000.

[9] A.E. Brouwer, A.M. Cohen, and A. Neumaier. *Distance-regular graphs*. Ergebnisse der Mathematik und ihrer Grenzgebiete, 1989.

[10] R. Caruana, M. Elhawary, A. Munson, M. Riedewald, D. Sorokina, D. Fink, W.M. Hochachka, and S. Kelling. Mining citizen science data to predict orevalence of wild bird species. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 915. ACM, 2006.

[11] D.A. Castañon. Efficient algorithms for finding the k best paths through a trellis. *Aerospace and Electronic Systems, IEEE Transactions on*, 26(2):405–410, 1990.

[12] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[13] A. Chan and C.D. Godsil. Symmetry and eigenvectors. In G. Hahn and G. Sabidussi, editors, *Graph Symmetry: Algebraic Methods and Applications*. Springer, 1997.

[14] E. Charniak. Statistical techniques for natural language parsing. *AI Magazine*, 18(4):33–44, 1997.

[15] W. Chen, S.H. Teng, Y. Wang, and Y. Zhou. On the α-Sensitivity of Nash Equilibria in PageRank-Based Network Reputation Games. In *Proceedings of the Third International Workshop on Frontiers in Algorithmics*, page 73. Springer, 2009.

[16] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 128–132, New York, NY, USA, 2005. ACM Press.

[17] Alice Cheng and Eric Friedman. Manipulability of PageRank under sybil strategies. In *Proceedings of the First Workshop of Networked Systems (NetEcon06)*, 2006.

[18] V. Chvátal. *Linear Programming*. W.H. Freeman, New York, NY, 1983.

[19] G. B. Dantzig. Application of the simplex method to a transportation problem. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, volume 13 of *Cowles Commission for Research in Economics*, pages 359–373. Wiley, 1951.

[20] George B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1963.

[21] P. Dellaportas and AFM Smith. Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 42(3):443–459, 1993.

[22] J.R. Douceur. The sybil attack. In *Peer-To-Peer Systems: First International Workshop, IPTPS 2002, Cambridge, MA*, page 251, 2002.

[23] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[24] A. Fabrikant, A. Luthra, E. Maneva, C.H. Papadimitriou, and S. Shenker. On a network creation game. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 347–351. ACM New York, NY, USA, 2003.

[25] D. Fogaras and B. Rácz. Towards scaling fully personalized pagerank. *Lecture notes in computer science: Algorithms and Models for the Web Graph*, pages 105–117, 2004.

[26] Eric Friedman, Paul Resnick, and Rahul Sami. Manipulation-resistant reputation systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[27] Krishna Gade and Amit Prakash. Using transient probability distributions of random walk to estimate spam resistant authority scores. Unpublished manuscript, 2007.

[28] C.D. Godsil. *Algebraic Combinatorics*. Chapman & Hall, 1993.

[29] C.D. Godsil and B.D. McKay. Feasibility conditions for the existence of walk-regular graphs. *Linear Algebra and its Applications*, 30:51–61, 1980.

[30] C.D. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

[31] A.V. Goldberg, S.A. Plotkin, and É. Tardos. Combinatorial algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 16(2):351–381, 1991.

[32] Zoltán Gyöngyi, Pavel Berkhin, Hector Garcia-Molina, and Jan Pedersen. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Databases*. ACM, 2006.

[33] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Databases*, pages 517–528. ACM, 2005.

[34] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[35] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587. Morgan Kaufmann, 2004.

[36] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5(1):43–85, 1995.

[37] D.S. Hochbaum and J.G. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *Journal of the ACM*, 37(4):843–862, 1990.

[38] J. Hopcroft and D. Sheldon. Manipulation-Resistant Reputations Using Hitting Time. *Lecture Notes in Computer Science: Algorithms and Models for the Web Graph*, 4863:68–81, 2007.

[39] M.O. Jackson. A Survey of Models of Network Formation: Stability and Efficiency. In *Group Formation in Economics: Networks, Clubs and Coalitions*, pages 11–58. Cambridge University Press, 2005.

[40] Glen Jeh and Jennifer Widom. SimRank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[41] H. Jiang, S. Fels, and JJ Little. A linear programming approach for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007*, pages 1–8, 2007.

[42] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[43] A.V. Karzanov and S.T. McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM Journal on Computing*, 26(4):1245, 1997.

[44] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[45] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling se-

quence data. In *Proceedings of the International Conference on Machine Learning*, pages 282–289, 2001.

[46] Amy N. Langville and Carl D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004.

[47] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

[48] Kahn Mason. *Detecting Colluders in PageRank - Finding Slow Mixing States in a Markov Chain*. PhD thesis, Stanford University, 2005.

[49] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[50] M. Arthur Munson, Kevin Webb, Daniel Sheldon, Daniel Fink, Wesley M. Hochachka, Iliff, Mirek Riedewald, Daria Sorokina, Brian Sullivan, Christopher Wood, and Steve Kelling. The eBird Reference Dataset. Cornell Lab of Ornithology and National Audubon Society, Ithaca, NY, June 2009.

[51] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[52] Christos Papadimitriou. The Algorithmic Lens: How the Computational Perspective is Transforming the Sciences. Computing Community Consortium Presentation at Federated Computing Research Conference, San Diego, CA, 2007.

[53] Kevin T. Phelps. Automorphism free latin square graphs. *Discrete Mathematics*, 31(2):193–200, 1980.

[54] S. J. Phillips, M. Dudík, and R. E. Schapire. A maximum entropy approach to species distribution modeling. In *Proceedings of the International Conference on Machine Learning*, 2004.

[55] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[56] T. R. Robinson, R. R. Sargent, and M. B. Sargent. Ruby-throated Hummingbird (*Archilochus colubris*). In A. Poole and F. Gill, editors, *The Birds of North America*, number 204. The Academy of Natural Sciences, Philadelphia, and The American Ornithologists' Union, Washington, D.C., 1996.

[57] Brian W. Rogers. A Strategic Theory of Network Status. Manuscript. `http://www.its.caltech.edu/~leectr/workshop06/talks/JacksonTalk.pdf`, 2009 (last accessed December, 2009).

[58] D. Roth and W. Yih. Integer linear programming inference for conditional random fields. In *Proceedings of the International Conference on Machine Learning*, page 743. ACM, 2005.

[59] D. Sheldon, M. A. S. Elmohamed, and D. Kozen. Collective inference on Markov models for modeling bird migration. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1321–1328. MIT Press, Cambridge, MA, 2008.

[60] J. Shore and R. Johnson. Properties of cross-entropy minimization. *IEEE Transactions on Information Theory*, 27:472–482, 1981.

[61] B.L. Sullivan, C.L. Wood, M.J. Iliff, R.E. Bonney, D. Fink, and S. Kelling. eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292, 2009.

[62] Éva Tardos and Tom Wexler. Network formation games and the potential function method. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[63] Eric W. Weisstein. Distance-Regular Graph. From MathWorld – A Wolfram Web Resource `http://mathworld.wolfram.com/Distance-RegularGraph.html`, 2009 (last accessed December, 2009).

[64] Eric W. Weisstein. Edge-Transitive Graph. From MathWorld – A Wolfram Web Resource `http://mathworld.wolfram.com/Edge-TransitiveGraph.html`, 2009 (last accessed December, 2009).

[65] JK Wolf, AM Viterbi, and GS Dixon. Finding the best set of K paths through a trellis withapplication to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2):287–296, 1989.

[66] H. Zhang, A. Goel, R. Govindan, K. Mason, and B.V. Roy. Making

eigenvector-based reputation systems robust to collusion. *Lecture notes in computer science: Algorithms and Models for the Web Graph*, 3243:92–104, 2004.

[67] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008*, pages 1–8, 2008.

[68] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning*, 2003.