BEHAVIORS, INTERACTIONS, AND COMMUNITIES IN NETWORKS

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

Isabel Mette Kloumann

May 2016

BEHAVIORS, INTERACTIONS, AND COMMUNITIES IN NETWORKS
Isabel Mette Kloumann, Ph.D.
Cornell University 2016

Exciting and unexpected patterns can emerge when systems are highly connected, even when they are composed of the simplest objects. In this thesis we investigate how networks of people, oscillators, apps, and nodes can be better understood through the behavior of the communities that emerge from their close interactions. Work in this thesis first examines how recent advances in dynamical systems have shed new light on the macrobehavior of networks of coupled oscillators. The dimensionality of the system is greatly reduced by viewing the system not as one of individual coupled oscillators, but as one of a smaller number of interacting groups. We demonstrate that the corresponding governing equations can be solved exactly. This thesis then investigates the seed set expansion problem, or how to uncover the local community structure hidden around nodes, in social and other real-world networks. We explore how topological properties of communities and seed sets correlate with algorithm performance, and explain these empirical observations with theoretical ones. We then turn our focus back to a theoretical setting and develop a principled framework for evaluating ranking methods by studying seed set expansion applied to the stochastic block model. We derive the optimal gradient for separating the two classes of nodes in a stochastic block model, and find, surprisingly, that it is asymptotically equivalent to personalized PageRank. This connection provides a novel formal motivation for the success of personalized PageRank in seed set expansion and node ranking generally. We then leverage this framework to develop several theoretically motivated heuristics that incorporate higher moments of landing probabilities, and show that these techniques yield much stronger performance on seed set expansion for stochastic block models. Work in the second part of this thesis discusses two other highly connected networks, the Facebook social network, and the network of communication between researchers in a series of massive collaborations. In the first case we develop a retention model that accurately models users' tendencies to continue using apps, and at the social level we organize apps along two fundamental axes – popularity and sociality – and show how a user's probability of adopting an app depends on properties of both the local network structure and the match between the user's attributes, their friends' attributes, and the dominant attributes within the app user population. We show how our models give rise to compact sets of features with strong performance in predicting app success. In the second case we study a series of massive online collaborations of professional and amateur mathematicians, who collectively attempt to solve open problems in mathematics research. We identify interesting patterns in the linguistic structure and social reactions that distinguish important research contributions from less important ones. We also observe distinct changes in the language behavior, and the structure and timing of interactions between the same group of contributors depending on whether they are working on a research problem, or simply a very difficult but solved problem taken from the International Math Olympiad.

BIOGRAPHICAL SKETCH

This biographical sketch was provided by the author's oldest and dearest friend, Phoebe, who surely knows her better than she knows herself.

Isabel Mette Kloumann was born in Vermont, the youngest child of a Norwegian and a Venezuelan-American. Known by her friends and family as loyal, loving, and fiercely empathetic, she grew up tall in a family of strong women, thinking for herself, taking risks, and getting lots of scrapes and bruises along the way. Except for a fear of spiders, she very much resembles the protagonist of Ogden Nash's Adventures of Isabel:

Isabel met an enormous bear,
Isabel, Isabel, didn't care;
The bear was hungry, the bear was ravenous,
The bear's big mouth was cruel and cavernous.
The bear said, Isabel, glad to meet you,
How do, Isabel, now I'll eat you!
Isabel, Isabel, didn't worry.
Isabel didn't scream or scurry.
She washed her hands and she straightened her hair up,
Then Isabel quietly ate the bear up.

Isabel met a big data set,
but Isabel, Isabel, didn't fret;
The data was massive, disorganized,
The data wasn't standardized,
The data said, Isabel, glad to meet you,
How do, Isabel, now I'll beat you!
Isabel, Isabel, didn't worry.
Isabel didn't scream or scurry.
She clapped her hands, and that data relented
When elegant algorithms she invented.

Isabel received her bachelors in math and physics from the University of Vermont, graduating summa cum laude in 2011 and with honors in both majors. At the University of Vermont she was a member of Professor Joanna Rankin's pulsar astronomy research group, and a member of the NanoGrav collaboration: an international initiative to use pulsar timing to detect gravidational waves. She was also a member of Chris Danforth and Peter Dodds' Computational Story Lab, and it was here that she first learned of the exciting adventures to be had at the intersection of math and the social sciences. After UVM she moved to Ithaca to do her Ph.D. work in Cornell University's Center for Applied Mathematics, where she studied network science under the guidance of Jon

Kleinberg and Steven Strogatz. During Cornell's summer furloughs of 2014 and 2015 she worked on Facebook's Core Data Science team, with the mentorship of Shaomei Wu and Lada Adamic. After completing her doctoral work at Cornell she will return to Facebook as a more permanent member of the Core Data Science team. Her East coast family and friends are happy for her, (sic) *despite being totally depressed that she'll be so far away.*

In Ithaca, in her spare time, Isabel enjoys relaxing at home with her fiance Jonathan Maddison (who has been her trusty partner in crime throughout all of the above) and their two cats, doing adventurous outdoorsy things, and having breakfast in bed.

For Jonathan. To you, my love, I dedicate everything.

ACKNOWLEDGEMENTS

Oh, how can words
express my gratitude?
A mere acknowledgement
feels lacking, rude.
For my thanks to those
who helped me here,
is, from the bottom of my heart,
sincere.

Firstly I thank
the inimitable Jon Kleinberg,
for you are
every kind of genius,
and could not be nicer.
Though this may be brazen,
you have made me much wiser.
What more could one ask for
than that kind of advisor?
My next great hero
is the real Steve Strogatz,
dropping candid counsel,
teaching the elegance of math.
Your humility is a solace
from the ruling bravado.
The two of you
I thank for everything,
and feel I won the advisor lotto.

Erich Mueller, I thank you
for thought-provoking conversations,
your infectious curiosity
will inspire generations!
For your probing questions
get to the heart of the matter,
and making such connections
is the key of the master.

Incoming!
thanks is huge, Shaomei Wu,
for in fact it is quite crucially
thanks to you,
thanks to your daring, your honesty,
your model, no false humility,
that I follow your footsteps
and leap into industry.
To Facebook Core Data Science,
thanks to the valiant team,
and most dearly to Lada Adamic
for your guidance is keen.
Your band, like no other,
is a squadron of masters,

a hub of the best
scholars, makers, and inventors.

And thank you to
the ever-wise Lillian Lee,
most sincerely for your
guidance and creativity.
Your model is clear,
cogent and brave,
as we seek to understand
how the humans behave.
And to Chenhao Tan, for your
clarity and deep thinking.
Of your great mind,
your questions give us a great inkling!

And to Ian Lizarraga,
thank you both as friend and coauthor,
With your bright spirit and wit
there is not another.
And to my academic big brother,
thank you Johan Ugander,
For your generosity in counsel,
support, time, and candor!

Thank you team storylab,
Chris Danforth and Peter Dodds,
For your creativity and passion
(perhaps a sign from the gods?)
First showed me that applied math
is by far the best job,
And without your kick-start,
I wouldn't be here. Full stop.

Now here I start to feel
even more than sentimental,
for the following people
are my utter fundamentals.
Marilyn, Robert, Nathaniel,
Franz and Danielle,
Marshall, Amalie and Province,
Christina and Mikey,
Michael and Mickey,
Mary Lou and Jonn!
Lauren!
Alex!
Dad!
Phoebe!
You people
let me float.
Let me
rise up without fear,
For should I fall or fly
I know you will be here.

Mom, you gave us everything,
and then you gave more.
The will to heal scars,
The lifeblood, the sword.
Your mind, your beauty.
Your love and your strength.
I could write you
a poem of limitless length.
I see you
and I see all that your are.
I see it, I am awed,
and thank you for the stars.

Now to Jonathan, my dear,
with you I stop.
I breathe.
For with you I am me
and with your love I am free.
You have done more than give everything.
You have taught me how to give,
how to love someone fully,
how to love and to live.
For you I will strive to be better,
will strive to be strong,
for I cherish all that you are
and with you I belong.

A thesis is a document
but it's exalted role in our lore
raises it beyond a deed
to something quite more.
Perhaps between these lines
you will see
the sleepless nights, the struggle, the fight.
Here you may read words,
but me, I read time.
And know that I have
given myself to this thing, line by line.
And have grown
to know that I must keep going, keep growing.
Thank you. For it is the connection
with you here
that lets me go for it.

The five main chapters of this thesis were the result of collaborations with my advisors and colleagues, and I would like to give my wholehearted thanks for the pleasure of their time and support. Chapter 2 was coauthored with Ian Lizaragga and Steven Strogatz, and we would additionally like to thank Murray Strogatz for very helpful discussions. Chapter 3 was coauthored with Jon Kleinberg, and we would like to thank Michael Macy, Silvana Toska, and Shaomei Wu for their collaboration and the useful discussions that helped to inspire this chapter. Chapter 4 was coauthored with Lada Adamic, Jon Kleinberg, and Shaomei Wu and was work started while doing an internship at Facebook as part of the Core Data Science team. Chapter 5 was coauthored with Chenhao Tan, Jon Kleinberg, and Lillian Lee, and we would like to thank the students that took the course CS6742 (Fall 2014) at Cornell for their comments, and also to particularly thank John Chavis, Zachary Clawson, Stephen Cowpar, David Eriksson, Hyung Joo Park, and Evan Randles for serving as judges in the prediction tasks.

Finally, thank you to Cornell's Center for Applied Mathematics for supporting my graduate work and for embodying the spirit of Cornell: an institution where any person can find instruction in any study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Networks are a powerful lens for understanding and modeling complex interactions in the natural world. Their power comes from the balance they offer between complexity and simplicity: on one hand, they enable us to model in detail very large systems with many components interacting in non- trivial ways; and on the other hand, insights from the rich mathematical study of networks provide us with a means for characterizing them in simple but important ways. Data are useful only insomuch as we can understand them. This thesis is a quest for gleaning insight from data, and in the quest we will find that networks are an invaluable lens for doing so. We will also find that there are many mirages in this quest for insight. And thus we will continually turn to statistics and machine learning for counsel and objective frameworks for evaluating the significance of our observations. The quest is long and the path is not lain, but therein lies the adventure, and we are well provided for.

Networks have been used to model systems from almost every academic discipline:

- Physicists use networks to model the interactions between atoms in crystals and other materials;
- Neuroscientists use them to model neurons and neuronal pathways in the brain;
- Computer scientists use them to model network flow for routing;
- Economists use them to model the relationships between businesses and institutions in the economy;

Node behavior Usually in a network, scientists are interested in studying the behavior and connections among a set of objects; these objects are called the nodes. For example, if we have a network of neurons, we are interested in understanding and modeling whether or not a given neuron is going to fire. In this case the neuron is the node in the network.

Interactions on edges The network edge is then the pathway via which the behavior of nodes interact with each other. In the network of neurons, the edges may represent neuronal pathways, which are the physical connections that cause a neuron that has just fired to instigate firing in its neighbors.

Community structure When a subset of nodes share some common characteristics we say they form a community, for example by all being more densely connected to each other than to other nodes. A community of nodes may be characterized by their all sharing some common role in the network. For example, in a network of neurons, we might find that there is a community of neurons that all have the role of controlling some aspect of an organism's behavior, such as vision. Communities are a lens that help bring a large, apparently tangled mess of a network into focus. In this thesis we will take a broad view of communities, at times interchanging our reference to them with references to groups and types of objects within a system or dataset, though in some contexts we note that a community may a more specialized connotation.

Throughout this thesis we will take an interest in viewing a system through the right lens of community and group behavior, and see that when we do so patterns emerge that were otherwise hidden in a tangled mess.

This thesis is comprised of five works, each of which is inspired by questions about how a large system with many individual components can be better understood by viewing the system as comprised of a smaller number of interacting groups.

In the spirit of identifying simple patterns in large systems, let's begin by considering a few of the questions that inspired this thesis:

- How can identifying subpopulations in a system of $N >> 1$ fully connected coupled oscillators help us exactly describe an otherwise intractable system?
- How should we use network structure to identify the other members of a node's community?
- How does local social network structure in behavior adoption predict long-term success of web-services?

- How do communication patterns in a collaboration help distinguish contributions that are meaningful from those that aren't?

1.1 Oscillators

1.1.1 Collective Synchronization

Have you ever heard crickets chirping in synchrony on a late summer's evening? If you have, then you have borne witness to the beauty of collective synchronization [118]. You also have collective synchronization to thank for getting a good regular night's sleep as the circadian pacemaker cells in your brain help synchronize and unify the period of your bodily functions [78]. Collective synchronization is a ubiquitous natural phenomenon whereby the interactions in a system of many individual, independent oscillators give rise to their synchronization and collective oscillation at a common frequency. We'll see in this thesis is how a notion of community structure can help us understand complex sets of synchronizing units.

1.1.2 The Kuramoto Model

The history of studying collective synchronization is marked by the contributions of Kuramoto [70] when he showed that for a system of nearly identical, weakly coupled oscillators the long term dynamics are given by:

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^{N} \Gamma_{ij}\left(\theta_j - \theta_i\right), \;\; i = 1, \ldots, N. \tag{1.1}$$

In this case, the system contains N oscillators, where their states are described by their phases on the unit circle $\{\theta_i\}_{i=1}^{N}$, for $\theta_i \in [0, 2\pi)$. Each oscillator has a natural frequency ω_i, which would be its phase velocity in the absence of interactions or coupling with the other oscillators. Typically these natural frequencies are assumed to have a normal or

Lorentzian distribution with mean 0, and thus are symmetrically distributed such that $g(\omega) = g(-\omega)$. Oscillator i is coupled to oscillator j via some interaction function Γ_{ij} defined on the difference in phases between the oscillators: $\theta_j - \theta_i$. When $\Gamma_{ij}(\theta_j - \theta_i) < 0$, it means that the interaction between j and i is slowing i down beyond its natural frequency ω_i. Finally, (1.1) indicates that the effect of the other oscillators on the phase velocity of i, $\dot{\theta}_i$, is the sum of the individual interactions between i and the other oscillators.

The rich history of coupled oscillators is told in depth by Strogatz in [109], but here we briefly summarize the context. The mathematical study of coupled oscillators begins with Wiener in the late 1950s [27, 28] but his formulation was intractable and little progress was made. That is until Winfree motivated a key assumption, that the coupling between oscillators is weak and the oscillators are nearly identical. Winfree's assumption and the corresponding simplification about the relation between an oscillator and the mean-field of a system paved the way for Kuramoto's result in (1.1). But as Strogatz observes in [109], (1.1) was still too complex to be analyzed generally. Thus Kuramoto took the simplification further. He focused on the even simpler system in which all oscillators are coupled to all the other ones with equal weights via the sine of the difference in their phases:

$$\dot{\theta}_i = \omega_i + \frac{K}{N} \sum_{j=1}^{N} \sin\left(\theta_j - \theta_i\right), \quad i = 1, \ldots, N. \tag{1.2}$$

This all-to-all and equally weighted coupling corresponds to the view of the oscillators being connected by a complete graph with equal edge weights. Note that with sinusoidal coupling, when $\theta_j > \theta_i$, $\sin(\theta_j - \theta_i) > 0$ and thus the coupling between i and j tends to bring i closer to j.

The Order Parameter

Each of the N oscillators are identically distributed in the Kuramoto model, (1.2). Thus it is useful to define the following order parameter, which represents the mean position of all N oscillators as they oscillate about the unit circle in the complex plane [109]:

$$re^{\mathrm{i}\psi} = \frac{1}{N}\sum_{j=1}^{N} e^{\mathrm{i}\theta_j}. \tag{1.3}$$

Here we can interpret r as the average radius of oscillators from the center of the unit circle, and ψ as their average phase. Without loss of generality we can assume that we are analyzing the system in the rotating frame of the oscillator's average position, in which case $\psi = 0$.

Kuramoto used this order parameter describing the average behavior to rewrite (1.2). To see this, begin by multiplying both sides of (1.3), continue be equating the imaginary parts of this expression, and conclude by observing the connection with the governing equations in (1.2):

$$\begin{aligned} re^{\mathrm{i}(\psi-\theta_i)} &= \frac{1}{N}\sum_{j=1}^{N} e^{\mathrm{i}(\theta_j-\theta_i)} && (1.4)\\ r\cos(\psi-\theta_i) + \mathrm{i}r\sin(\psi-\theta_i) &= \frac{1}{N}\sum_{j=1}^{N} \big(\cos(\theta_j-\theta_i) + \mathrm{i}\sin(\theta_j-\theta_i)\big) && (1.5)\\ r\sin(\psi-\theta_i) &= \frac{1}{N}\sum_{j=1}^{N} \sin(\theta_j-\theta_i) && (1.6)\\ \dot{\theta}_i &= \omega + Kr\sin(\psi-\theta_i). && (1.7) \end{aligned}$$

With the interpretation of r and ψ mentioned above, we can see that the pull on i from the rest of the population increases as the mean population position r is further from the unit circle. Further, we can observe that when $\theta_i < \psi$ we have $\sin(\psi-\theta_i) > 0$; that is, when oscillator i trails behind the population average, the effect of the average on i is to speed it up.

1.1.3 The Van Hemmen Model: Contributions

In the previous paragraphs we have considered a system of N coupled oscillators, each of which has natural frequencies drawn from the same population, and between each of which there is uniform coupling of some strength K. In this case, the introduction of an order parameter $re^{\mathrm{i}\psi}$ enabled us to view the interactions in the system as one between an oscillator and the population average.

In other cases we may need to model oscillators that are connected in a grid, or where the coupling between oscillators tends to push them away from one other. While some of these systems may not be describable with an order parameter, in some cases we may be able to describe the system in terms of a small number of order parameters, corresponding to the different types of oscillators.

In Chapter 2 of this thesis we consider the Van Hemmen model, a system of coupled oscillators in which each oscillator falls into one of four subpopulations. Van Hemmen's model has two coupling parameters, and depending on the values of J and K it was observed numerically that there were four stable states of the system. For Van Hemmen's model we will utilize the ansatz first suggested by Ott and Antonsen in [89] and introduce an order parameter for each of the four types of oscillators. This combination of techniques will enable us to exactly describe the evolution of the order parameters of the four subpopulations.

The low-dimensional reduction is powerful. We have a system of N oscillators, each with their own random natural frequency ω, but the coupling that connects them with other nodes in the graph means that oscillators are more usefully thought of not as individuals, but as members of a group. The macrobehavior of the system can be described exactly in terms of the dynamics between the groups, and because the number of groups is small, the equations describing these dynamics can be solved exactly. In Chapter 4 we will see that an analogous simplification involving communities of densely connected nodes on a graph will be crucial to learning how to identify those communities.

Chapters 3 and 4 of this thesis will again focus on graphs but from a new perspective than the one in Chapter 2. In Chapter 2 our graphs were fully connected and the state of each node was described by a phase in the complex plane. The weights on the edges that connected nodes were determined by a static property of the node (which community it was in). In the Kuramoto model, each node was in the same community, and in the Van Hemmen model, nodes could be one of four types, and thus their type determined the sign of their coupling relative to the remaining nodes. In Chapters 3 and 4 we consider graphs that are dense but not fully connected: the likelihood of there being an edge connecting two nodes will depend on the (hidden) community affiliations of those nodes. Our goal will be to learn how to identify which nodes constitute the community of a particular node of interest, given just the network.

1.2 Networks with Community Structure

College friends, family friends, summer camp friends, work friends: our personal social networks are a reflection of the variety of our experiences. Identifying the source of our connections with people helps us understand what otherwise would appear to be a tangle of relationships and mutual friendships. In many applications a researcher may be able to observe those friendships without understanding why those connections arose. Community detection has arisen as a means to uncovering those hidden macro structures around which a network is arranged, enabling us to bring info focus what otherwise would appear as a tangled mess.

Communities of revolutionaries

Let us pause to consider a concrete example of a social network. In 2011 Jon Kleinberg and I were studying activism in Egypt via twitter, in collaboration with Michael Macy, a sociologist, Silvana Toska, a government Ph.D. student, and Shaomei Wu, a Cornell Information Science alumna. At the time Egypt was in the midst of a revolution, popularly associated with the string of revolutionary movements in the Middle East dubbed the “Arab Spring”. Most people involved in the political revolution were either secular, Salafi, or Islamicist. From previous research on twitter networks and interactions we knew that users tended to retweet messages from people to whom they were ideologically close. Thus to identify people who were ideologically close to one another amounted to finding groups of users clustered in the network of retweets. Each tweet in the data has the following structure[1]:

- *user*: the user id of the tweet's author,
- *text*: the text of the tweet (up to 140 characters),
- *retweeted*: the user id of the original author if *text* is a retweet.

To construct the network of retweets $G = G(V,E)$, we need to identify the nodes and edges that comprise that graph: in this case, the set of nodes V is the union of all users *and* all retweeted users in the data. For each pair of nodes $(u,v) \in V \times V$, $(u,v) \in E$ if there is some tweet where user u retweeted user v.

Each node in V has some hidden label, corresponding to whether that user is a secular, Salafi, Islamicist, or none of the above. How should we use the network of retweets to understand to which ideological group each user belongs?

1.2.1 Community Detection

This simple question is intimately related to a more general one about nodes in a graph: how should we partition nodes in the network such that nodes within the same partition are close to one another?

With this question we open the door to an expanse of possibilities and a rich literature that sits at the intersection of computer science, mathematics, biology, and sociology. The wide ranging and sustained interest in the community detection, or graph partitioning, problem is a reflection of the ubiquity of networks with community structure in industrial and academic disciplines, and the corresponding diversity of network and community structures therein. As a computer scientist we may be interested in deciding

[1] In reality the tweets had additional metadata not relevant to this discussion, such as the time stamp of when the tweet was authored, and in some cases location data indicating where the tweet author is from, and where they were when the tweet was authored.

how to parallelize our code and so need an approximately balanced partition that minimizes communication between machines [49]. Meanwhile a sociologist may be less interested in community structure defined by some mathematical objective, and more interested in identifying a robust algorithm that accurately reconstructs partitions that correspond to real world groups, however they are structured.

This variety of needs means that a solution to one partitioning problem may be ineffective when applied to another. With this caveat in mind, we should always be careful to identify a plan for benchmarking our solution, aware that it may not be robust to other notions of quality. Here we will entertain our curiosity in considering a handful of intuitive and powerful approaches to community detection, each of which has been successful in a distinct variety of domains, though not necessarily the same ones. We point the interested reader to *Community detection in graphs* by Santo Fortunato [41], which is a thorough review of the methods and motivations up to the year 2010, and is accessible to boot.

Balanced partitions

A natural starting objective for graph partitioning is to identify balanced cuts in a graph. In our motivating example, identifying a balanced minimum cut on the retweet network corresponds to figuring out how to split the two sets of users in half, such that a minimal number of users had been retweeted by someone from the other side. The number of retweets going from one side to the other is thought of as the cost of the partition. More generally we can denote a partition of the nodes V into k sets by $\mathscr{S}$ where $\mathscr{S} = \{S_i\}_{i=1}^{k}$ (for our example, $k = 4$). The cost of a partition on an unweighted graph is the number of edges that go *between* the sets. Formally, we would say that the cost of that partition is:

$$cost(\mathscr{S}) = \sum_{i=1}^{k} \sum_{j=1 \neq i}^{k} \left| S_i \times S_j \bigcap E \right| \tag{1.8}$$

where $|S|$ denotes the number of elements in set S.

It is usually of interest to introduce a constraint that forces the partitions to be nearly balanced. We might fix a parameter $v \geq 1$ and require that each set S_i of the partition contains no more than $v * n/k$ nodes. This approach to balanced graph partitioning is intuitive and powerful. However, it has a couple of limitations:

- It's expensive! In general (and even in special cases of graphs such as trees or grids) there is not even a polynomial time approximation algorithm that guarantees a finite approximation ratio for this problem (unless $P = NP$) [9, 38, 39].
- Is it really getting us what we want? Minimizing the number of edges to be cut is intuitive and in many utilitarian applications makes sense. But in social networks, is there some other graph metric that would better capture real-world community structure?

Practitioners who only have the computational complexity issue to deal with have developed many flexible solutions to it. One of the most popular is to iteratively create a coarsened analogue to the graph of interest by collapsing nodes and edges, partitioning that much smaller graph, and then mapping that partition back onto the original, larger one. METIS is one example of a "multilevel" method that has been very successful and which quickly returns very high quality partitions [62].

But for many community detection problems a minimal cost balanced partition may not be the right objective, so the first issue and its solutions are less germane. Instead we would want to consider alternative metrics that better reflect the community structure of interest.

Modularity

In 2004 Newman and Girvan [87] introduce a new metric for community structure: they asked how the number of edges between two sets compared to the number one would expect in a similar but random graph. They refer to this quantity as *modularity*. Nodes have the same degree distribution in the random graph as in the original one, but in the random graph the edges are randomly rewired so that in expectation there is no community structure.

Newman and Girvan offer a modularity-maximizing algorithm, and demonstrate that it identifies partitions which closely resemble the true partitions in real-world networks, such as the network of co-authorships (where communities are scientific disciplines), or the interactions between characters in the novel *Les Miserables*, where the communities correspond to the novel's numerous subplots. In our network of retweets, if we sought to maximize modularity that would mean we were trying to identify some grouping of Twitter users such that there were many *fewer retweets between* groups than we should naively expect.

Newman later demonstrated that the modularity-optimizing partition could be computed in time $O([n+m]n)$ where m and n are the number of edges and nodes in the graph [86]. In order to understand how his method works, we first need to recall the definition of the adjacency matrix of a graph, $G(V,E)$. We will see throughout our exploration of networks that the adjacency matrix will be a powerful starting point for computing many important and useful graph metrics. Let's refer to the nodes $u \in V$ by their (arbitrary but fixed) index i: $\{i\}_{i=1}^{n} = V$, where $n = |V|$. Then the adjacency matrix $\mathbf{A}$ of an unweighted graph $G(V,E)$ is:

$$A_{ij} = \left\{ 1 \text{ if } (i,j) \in E 0 \text{ else.} \right. \tag{1.9}$$

The degree vector $\mathbf{k}$ of the graph G is then $k_i = \sum_{j=1}^{n} A_{ij}$. Newman's method then has us compute the leading eigenvector of the modularity matrix $\mathbf{B} = \mathbf{A} - \frac{\mathbf{k}^T\mathbf{k}}{2m}$, where $\mathbf{A}$ is the adjacency matrix of the graph and $\mathbf{k}$ is the vector of degrees.

Conductance

Modularity as a metric of community structure pairs well with our intuitive notions, but it is not the only metric that does so. Modularity captures the extent to which nodes inside their own set are less well connected to nodes outside the set than if the graph were random. An alternative concept is the conductance of a cut (S,T), which is the ratio of edges between S and T and the total number of edges landing in the smaller of the two sets. We can define the conductance of a cut (S,T) formally:

$$c(S,T) = \frac{\sum_{i\in S, j\in T} A_{ij}}{min(a(S),a(T))} \tag{1.10}$$

where A_{ij} are the entries of the adjacency matrix $\mathbf{A}$, and $a(S) = \sum_{i\in S}\sum_{j\in V} A_{ij}$. To guide your intuition, you can quickly confirm that conductance is 1 when edges only go between the two sides, and is 0 when edges only stay within their respective side. In general conductance tends to be lower for “good cuts” (cuts with few edges between the two sides), and higher for bad cuts.

Enter PageRank And how can one identify cuts with good (low) conductance? In the early 2000s, in papers concurrent to those on modularity maximization and community detection, Andersen, Chung, and Lang observe that we can recover sets with theoretical guarantees of minimal conductance around seed nodes of interest by computing the Personalized PageRank of a seed node [7, 8]. The Personalized PageRank vector was at that point less than 10 years old but already widely recognized as a heuristic of great value. Along with Rajeev Motwani and Terry Winograd it was introduced by the founders of Google, Larry Page and Sergey Brin, in their seminal paper *The PageRank citation ranking: Bringing order to the web* [91]. PageRank in general is the steady state solution of a random walk on a graph starting from any seed node. In the coming section we will dive more deeply into what PageRank represents, and how it relates to other graph diffusions. These connections are meant to help the reader more deeply understand PageRank as it used in Chapter 3 of this thesis, and the motivation for the questions in Chapter 4. The starting point for understanding these quantities is to think about random walks on graphs.

1.2.2 Random walks and Graph Diffusions

PageRank

Page Rank was originally designed as a means of ranking the relevance of web pages on the web. Formulated by the founders of Google, Larry Page and Sergey Brin, along with their colleagues Rajeev Motwani and Terry Winograd, it was identified as a fast and accurate way of identifying pages that were both important and relevant to search topics.

Let's begin by considering a very simple ranking $\mathbf{r}$ defined on a graph G for each node $i \in V$:

$$r_j = c \sum_{i \in N} r_i \frac{A_{ij}}{\sum_{k=1} A_{ik}} \tag{1.11}$$

where A_{ij} are the entries of G's adjacency matrix $\mathbf{A}$, and

$$A_{ij} = \begin{cases} 1 \text{ if } (i,j) \in E \\ 0 \text{ else.} \end{cases} \tag{1.12}$$

We will see that PageRank is often computed iteratively, and if some nodes only have edges into them this leads to an overall loss of PageRank. Here we use c as a factor for normalization, so that the rank of all nodes in the graph is constant. In fact, we will assume throughout this discussion that all ranks $\mathbf{x}$ are normalized such that $\sum_{i=1}^{n} x_i = 1$.

What is the intuition behind this rank, $\mathbf{r}$? To compute the rank of node j it takes the rank of j's neighbors (the i for which $A_{ij} \neq 0$,) and passes a fraction of each of those ranks r_i onto j. The fraction of r_i allocated to j is $1/\sum_{k=1}^{n} A_{ik}$: this denominator is precisely the out-degree of node i. In other words, i distributes its rank evenly over its neighbors. As a result, r_j will be large when j has lots of neighbors with high rank (large r_i), who don't have too many neighbors themselves (otherwise j will have to share too much).

We can write (1.11) even more concisely if we view this process as a random walk. We can define what we will call the transition matrix **T**:

$$T_{ij} = \begin{cases} 1/d_i \text{ if } (i,j) \in E \\ 0 \text{ else.} \end{cases} \tag{1.13}$$

where $d_i = \sum_{k=1}^{n} A_{ik}$ are the entries to the degree vector **d**. To think about a random walk on a graph envision a surfer who starts at some node in the graph, and randomly follows edges. We call **T** the transition matrix for a random walk because each entry T_{ij} describes the probability of our random surfer transitioning from node i to node j. In some applications the graph may have "weighted edges", in which case we may want to redefine the transition matrix more generally as $T_{ij} = A_{ij}/d_i$.

Returning to our equation for the rank, we see that (1.11) can be usefully rewritten in terms of **T**:

$$r_j = c \sum_{i=1}^{n} T_{ij} r_i \rightarrow \mathbf{r} = c\,\mathbf{T} \cdot \mathbf{r} \tag{1.14}$$

where $\cdot$ denotes the inner product. In this last step we can see that **r** is the eigenvector of the transition matrix, **T**, with eigenvalue c. As Page *et al* say in [91], this rank is nearly the target rank, but it suffers one small but important issue: if there is a set of nodes S with no outbound edges but with inbound edges, S will become a "rank sink". That is, the random surfer, in this model, would eventually end up only surfing around within S. This process "unfairly" disfavors nodes not in S. To counterbalance the unfair effects of rank sinks, Page *et al* suggest an alternative ranking ρ, computed using the transition matrix and a new (democratic) rank source **e**:

$$\rho(\mathbf{e}) = c\,\mathbf{T} \cdot \rho(\mathbf{e}) + (1-c)\mathbf{e}, \tag{1.15}$$

where $\sum_{i=1}^{n} e_i = 1$. At last we have nothing less than the infamous PageRank. Taking **e** to be the uniform vector of $\mathbf{1}/n$, we effectively alter the random walk process such that our random surfer follows a random edge with probability c, or jumps to any random node in the graph with probability $1-c$.

The rank sink motivation for introducing the rank source $\mathbf{e}$ is, while important, potentially underwhelming. But in reality, by tuning the vector $\mathbf{e}$ we unlock the real power of PageRank: personalization. Tuning $\mathbf{e}$ to quantities other than $\mathbf{1}/n$ allows us to identify the rank of pages "as seen by" some particular node or nodes of interest. For example, say we place all the weight of $\mathbf{e}$ on a single node, $s \in V$, so that $e_s = 1$ and $e_i = 0$ for all $i \neq s$. In the corresponding walk, at each step the random surfer either follows an edge with probability c or jumps back to the seed node s with probability $1-c$. The walk's steady state distribution, $\rho(s)$, is interpretable as a ranking of node importance and proximity to s. The parameter c allows us to tune the relative contribution of each metric, where larger c emphasizes overall node importance, and smaller c emphasizes proximity to the seed node.

Representations of PageRank Now that we have begun to appreciate what PageRank represents, let's consider several alternative views of (1.15). For simplicity we will drop the source vector argument, $\rho = \rho(\mathbf{e})$.

$$\rho = c\,\mathbf{T} \cdot \rho + (1-c)\mathbf{e} \tag{1.16}$$

Note that $\sum_j \rho_j$ and thus for any vector $\mathbf{x}$ we have

$$((\mathbf{x} \times \mathbf{1}) \cdot \rho)_i = \sum_j (\mathbf{x} \times \mathbf{1})_{ij} \rho_j = \sum_j x_i \rho_j = x_i \sum_j \rho_j = x_i. \tag{1.17}$$

This leads to an alternate view of (1.15):

$$\begin{aligned} \rho_i &= c\,\sum_j T_{ij}\rho_j + (1-c)e_i && (1.18)\\ &= c\,\sum_j T_{ij}\rho_j + (1-c)\sum_j e_i(\mathbf{1})_j\rho_j && (1.19)\\ \rightarrow \rho &= c\,\mathbf{T}\rho + (1-c)(\mathbf{e}\times\mathbf{1})\cdot\rho && (1.20)\\ &= (c\,\mathbf{T} + (1-c)(\mathbf{e}\times\mathbf{1}))\cdot\rho. && (1.21) \end{aligned}$$

From this point of view $\rho(\mathbf{e})$ is the eigenvector of the matrix $(c\,\mathbf{T}+(1-c)(\mathbf{e}\times\mathbf{1}))$ with eigenvalue 1.

Let us now revisit the interpretation that PageRank, ρ, is the stationary distribution of a random walk. The random walk, we asserted, that leads to PageRank, was the one in which a random surfer with probability c followed a random edge from whichever node it started on, or with probability $1-c$ jumped to a node with probability specified by the source rank vector $\mathbf{e}$. At step 0 of the random walk the surfer is at a node i with probability x_i^0 which is the ith coordinate of the starting vector $\mathbf{x}^0$. Then at step 1 the surfer will be at node i according to x_i^1, where:

$$\mathbf{x}^1 = c\mathbf{T}\mathbf{x}^0 + (1-s)\mathbf{e}. \tag{1.22}$$

And at step k,

$$\mathbf{x}^k = c\mathbf{T}\mathbf{x}^{k-1} + (1-s)\mathbf{e} = (c\mathbf{T}+(1-s)(\mathbf{e}\times 1))\cdot\mathbf{x}^{k-1} = \tilde{\mathbf{T}}\cdot\mathbf{x}^{k-1} \tag{1.23}$$

Here let us make an observation: in the last step we see something closely resembling our equation for PageRank: for very large k if $\mathbf{x}^k$ is converging, we would have that $\mathbf{x}^k = \mathbf{x}^{k-1}$, and thus from (1.23) we have $\mathbf{x}^k = \tilde{\mathbf{T}}x^{k-1}x^k = \tilde{\mathbf{T}}x^k$. While we cannot always guarantee that repeated multiplication by any generic matrix $\mathbf{T}$ will converge, we do know that if $\mathbf{T}$ has a nonnegative dominant eigenvalue, this repeated multiplication would converge to an eigenvector associated with that eigenvalue. Fortunately, from the Perron-Frobenius theorem for nonnegative matrices, which applies to $\tilde{\mathbf{T}}$, we can guarantee that $\tilde{\mathbf{T}}$ has a positive, dominant eigenvalue whose associated eigenvector is positive. If $\tilde{\mathbf{T}}$ has strictly positive entries then a stricter version of the Perron-Frobenius theorem applies, and this version guarantees that the eigenvector corresponding to the dominant eigenvalue is unique. Positivity of $\tilde{\mathbf{T}}$ is guaranteed if, for example, the rank source $\mathbf{e}$ distributes some positive amount of rank to every node. In both cases (nonnegativity and positivity), we have $\sum_j \tilde{T}_{ij} = 1$ for all i and so the Perron-Frobenius theorem guarantees that the dominant eigenvalue is 1. It follows then that as $k \to \infty$ we can guarantee the convergence of $\tilde{\mathbf{T}}^k x^0$ to the eigenvector ρ with eigenvalue 1.

Personalized PageRank With this result in hand, we return to the recursive, iterative form for $\mathbf{x}^k$ in (1.23), and focus on a special case: the starting vector for our iteration $\mathbf{x}^0$ will be the rank source vector $\mathbf{e}$. We will refer to this special case of PageRank as Personalized PageRank, or PPR, and the source vector $\mathbf{s}$. In the corresponding random walk interpretation, the surfer's starting probability distribution is equal to the rank source distribution. This connection allows us to neatly rewrite ρ:

$$\begin{aligned}
\rho &= \lim_{k\to\infty} \mathbf{x}^k && (1.24)\\
\text{where } \mathbf{x}^k &= c\mathbf{T}\mathbf{x}^{k-1} + (1-c)\mathbf{s} && (1.25)\\
&= c\mathbf{T}(c\mathbf{T}\mathbf{x}^{k-2} + (1-c)\mathbf{s}) + (1-c)\mathbf{s} && (1.26)\\
&= c^{k+1}(\mathbf{T})^k\mathbf{s} + (1-c)\sum_{k'=0}^{k}(c\mathbf{T})^{k'}\mathbf{s} && (1.27)
\end{aligned}$$

and thus

$$\begin{aligned}
\rho = (1-c)\sum_{k=0}^{\infty} c^k\,\mathbf{T}^k\cdot\mathbf{s} &= \sum_{k=0}^{\infty} w_k\,\mathbf{T}^k\cdot\mathbf{s} && (1.28)\\
\text{for } w_k = (1-c)c^k &\text{ and } c\in(0,1). && (1.29)
\end{aligned}$$

Here we can see that Personalized PageRank is an infinite weighted sum of the quantities $\mathbf{T}^k\mathbf{s}$ with weights $\{w_k\}_{k=0}^{\infty}$ as specified such that $\sum_{k=0}^{\infty} w_k = 1$. The quantities $\{\mathbf{T}^k\cdot\mathbf{s}\}_{k=0}^{\infty}$ are length-n vectors interpretable as the expected probability of being at a node i on the k-th step of a random walk with starting probability distribution $\mathbf{s}$ and transition matrix $\mathbf{T}$.

In Equation (1.29) it becomes clear that Personalized PageRank vector is just one of a large class of graph diffusions, where the diffusion's defining property is the set of weights $\{w_k\}_{k=0}^{\infty}$ used to weight each step in the random walk. Note that for Personalized PageRank the weight contributed by the kth random walk vector strictly decreases with k. The rate at which they decrease is clearly controlled by the choice of parameter, c, where as we increase c we place increasing importance on longer walks, which inherently carry less information about the original seed set, $\mathbf{s}$. This is intuitive, as c

corresponds to the probability that a surfer should follow a random edge rather than a graph, meaning that it places more weight on nodes' "importance" in the network than their proximity to the seed set. Indeed, with $c = 1$ we place no importance on the seed set and instead recover the original ranking vector introduced in (1.11).

The heat kernel and beyond

The heat kernel is another well known diffusion is this family of graph diffusions:

$$\mathbf{h} = e^{-t}\left(\sum_{k=0} \frac{t^k}{k!}(\mathbf{T}^T)^k \mathbf{s}\right) \tag{1.30}$$

where $w_k = e^{-t}\frac{t^k}{k!}$. There are efficient algorithms for computing the heat kernel rapidly [66], or it can be approximated by computing only the first few steps of the random walk. Unlike Personalized PageRank whose weights strictly decrease with k, the heat kernel has the property that its weights $\{w_k\}_{k=0}^{\infty}$ have an interior maximum for some $k > 0$, the position of which is determined by the diffusion parameter t.

The seed set expansion problem

We now return our focus to the motivating problem, seed set expansion: given a graph, and a handful of known group members, i.e. the "seed set", how should you use the graph to identify the remaining group members? This problem is distinct from the global partitioning problem in that we seek an algorithm that identifies local community structure around a prescribed group of nodes, $S \subset V$. Both of these graph diffusions, the heat kernel and Personalized PageRank, offer a way to compute the importance of nodes in a network relative to a seed set of interest, and their parameters allow us to tune the emphasis that we place on importance versus proximity. And both have been shown to accurately recover communities in real world settings [66, 67], as we discuss at length in Chapter 3. However, neither has been able to achieve performance near the theoretical optimum. This begs the question, are PageRank and heat kernel utilizing the full power of graph diffusions, or is there a set of weights $\{w_k\}_{k=0}^{\infty}$ (or $\mathbf{w}$) that would recover communities more accurately?

To explore this question we will introduce another frame for our approach to seed set expansion. We are given a graph G with nodes V and edges E. Nodes fall into one of two sets: S and $V-S$, and our seed set contained within S. We then assert that nodes in S have, on average, empirical walk count vectors of the form $\mathbf{a}$, while nodes in $V-S$ have, on average, empirical walk count vectors of the form $\mathbf{b}$. For seed set expansion, we seek a vector of weights $\mathbf{w}$ so that when we rank nodes according to $\mathbf{y} = \mathbf{w} \cdot \mathbf{a} - \mathbf{w} \cdot \mathbf{b}$ we will correctly rank nodes in the in-class above those in the out-class. Note here that the weights $\mathbf{w}$ that we use to weight each entry in the walk count vectors are analogous to those that we identified for the heat kernel and Personalized PageRank. That is, the heat kernel and PPR are both maps of random walk count vectors to a score, which can then be used to rank nodes according to their likelihood of being in the in-class.

1.2.3 Seed set expansion: Contributions

This brings us to Chapter 3 of this thesis. In Chapter 3 we identify several real-world web-scale data sets consisting of graphs, where nodes in the graph represent people and are labeled as being members of some social group. The availability of this ground truth data enables us to mathematically explore the question of how which heuristics are most robust in identifying real world social communities. In particular, since we have the ground truth labels associated with each node, we can check the performance of any algorithm by computing how close its output is to the “true” answer. Now that we have a robust framework that will enable us to test and validate a variety of methods, we are left now to identify which methods should be tested.

In Chapter 4 we explore the seed set expansion problem in a more controlled setting. We observe that random walk based methods are consistently the most powerful, scalable techniques in the literature, with some variation in their performance in different settings. The difference between competing random walk based methods amounts to varying the proportional weight that each walk count number contributes to the overall infinite sum. We then ask, for a stochastic block model, what are the optimal coefficients to use in this weighting? Here, we refer to optimal in the sense of those being the geometric discriminant function. We observe that, in fact, the weights of the optimal geometric discriminant vector correspond to Personalized PageRank. With this geometric point of view, we also observe that corrections to those weights via the covariance matrices can also yield substantial improvements.

1.3 Social Interactions and Data Mining

1.3.1 Learning from Data and Predicting the Future

In Part 2 of this thesis we turn our focus to the computational study of social interactions on the web. The digital traces that people leave on the web provide an unprecedented view into the rhythms and patterns of daily life, patterns in our interactions and communication, and the social circles with which we associate. With this data we are presented with opportunities and challenges. The challenges are where the most interesting question arise.

In the first part of this thesis we focused on questions about classification and prediction problems on graphs. In this second part we will leverage graphs as tools to help us understand how people's interactions are related to each other, and to identify interesting patterns of behavior in social networks. We will move beyond thinking about classification of nodes in networks, and instead think about more general data: we will ask questions like,

- *how can we identify when an app will be successful or not?*
- *will this person adopt this app?*
- *what's the difference between important contributions to a group discussion and unimportant ones?*

As in the first part, our we are often faced with a similar task when studying social interactions: what is the best way to tell the difference between things in these groups? That is, how should I use the information available to me to classify my data?

But predicting the future accurately is, to say the least, a very hard problem. If anyone promises they can predict the future for you with perfect accuracy, they are most surely a snake oil salesman. We can, however, often do better than a snake oil salesman by carefully learning from past data, evaluating at the present, and formulating careful estimates of likelihood of how future events will play out. Our ability to do this, one could say, is the great success of science itself.

There are many competing and successful frames for learning to predict and for learning the causal relationships between events. In part two of this thesis we will focus on prediction problems related to observational (i.e. not experimental) data.

To learn how to predict future labels from this data, we will leverage a suite of tools from machine learning and statistics, and in particular will focus on problems that utilize techniques from supervised learning. Here we provide background on the learning framework that we utilize extensively throughout Chapters 5 and 6 of this thesis. Often we will find that the exercise in learning to predict is a powerful tool for learning about the structure and relationship between entities.

1.3.2 Machine Learning

Let us start with an example: we are given a Yelp data set, where entry x_i is a restaurant review, along with a label y_i that tells us whether the restaurant review is favorable or critical. We think that the data x_i encodes information about the review's label, but we do not necessarily know how to use or interpret that information. We want to learn from our data so that we can know which features, such as the number of occurrences of the phrase good or the average sentiment of words in the review, will help us distinguish in the future restaurant reviews of the two types. In general each entry in our data x_i may contain a large number of features, which have unknown relationships with the target label.

Matchmaking Problems with Models What is the best way to learn to predict y_i based on an input x_i?

This depends on a number of things about the problem and motivation:

- Are the labels y_i binary? discrete? continuous? non-scalar?
- How does the dimensionality of x_i compare to the number of labeled examples you have to train?
- What is an acceptable amount of memory and time for training? For evaluation?
- Is it important to do feature analysis?

We can formulate this question more precisely. Let's say that our examples $\{x_i\}_{i=1}^{n}$ are in X, and our labels $\{y_i\}_{i=1}^{n}$ are in y. How should we learn an unknown function $f(X) \rightarrow y$ given our examples $\{x_i, y_i\}_{i=1}^{n}$ and a set of possible functions f? This set of possible functions that we can learn is called our "hypothesis space," and their domain is X.

When we choose a machine learning algorithm, such as logistic regression, support vector machines, or random forests, we are also choosing a hypothesis space. Some algorithms share the same hypothesis space, but are optimizing different objectives when picking the best function.

For example, logistic regression, linear support vector machines, and naive Bayes all seek to find the "best" separating hyperplane in X such that the two classes fall on opposite sides of the plane. But each algorithm has its own indications of what "best" is and its own approach to finding the "best" solution.

In this work we will mainly focus on moderately sized ($dim(x_i) < 10^4$ and $n < 10^5$) data sets, where we are interested in predicting binary or discrete labels. Accordingly this means that we will tend to utilize logistic regression, support vector machines, naive Bayes, or aggregation based prediction methods such as random forests or gradient boosted decision trees.

In many cases we will find that a simple model such as support vector machines or logistic regression are complex enough to accurately model the structure and separation in our data. We expect that the best approach, and the one we have learned to take, is to start with a simple model (such as a regression), and introduce additional levels of complexity as necessary.

Overfitting In some cases we will be able to identify that the model is actually "overfitting" the data: when the data are high dimensional, so is the separating hyperplane. With insufficient data points one can perfectly or nearly perfectly model the separation, while learning nothing that generalizes beyond the training step. In such cases we must restrict our search space, either by pruning features, tuning a parameter in our algorithm, or identifying a different algorithm less prone to overfitting.

Feature selection and importance It will, in some cases, be important that we can later identify which features are "important" in helping to predict correct outcomes, and to more deeply understand why certain examples were classified as they were. Some algorithms, such as logistic regression, offer a natural source for this evaluation in their output: the coefficients in logistic regression can be used to compute the odds of a positive classification versus any one of the predictor variables. Support vector machines, while having other benefits such as flexibility, indicate coefficients that have limited interpretability.

Unsupervised learning We have implicitly narrowed our exploration and discussion of algorithms by deciding to focus on *supervised* learning, where the data is labeled. But in many applications the data may be unlabeled and the researcher may be interested in identifying a natural clustering. In chapter 5, for example, we will observe the variety of login time series for a set of several thousand web-applications. Individually, the variety in volume and time scale, and the apparently random variations, make it difficult to interpret them. But by scaling each time series and "clustering" the data using k-means clustering, we are able to see that the data tend to fall into one of two categories: a sharp incline followed by a sharp decline, or a steady unchanging state. This statistically meaningful reduction of what is otherwise appears to be thousands of noisy curves, into two simple and interpretable ones, demonstrates the power and value of machine learning.

In the case of seed set expansion and community detection we assume, a priori, that nodes in the in-class are closer to the seed set. The question then is to identify a notion of close that is accurate and can be computed in a reasonable amount of time.

Here, in contrast, we do not assume to know the relationship between any given feature and the label of interest. For example, if an app has a majority teen population in 2014, does that mean it is more, less, or equally likely to be successful one year into the future? Rather than assume anything about the relationship, instead we introduce a framework for learning from the data itself.

1.3.3 Structure of Interactions and Success Prediction: Contributions

In Chapter 5 we study adoption and retention of a thousands of apps through the lens of how over 1.4 billion people connected to them with their Facebook accounts. The connection between the app ecosystem and the underlying social network that connects people who use those apps offers us an unprecedented window into the competitive and fast paced world of apps. We find that a simple model motivated by the physics literature is able to accurately describe retention patterns. But we find that adoption behavior is much more nuanced. Depending on the app in question, the local social network of your friends who have adopted an app has an important relationship with whether you are likely to adopt the app in the future. We also observe interesting patterns in the demographic relationships between friends who use apps, and how those relationships relate to whether you adopt the same app as your friends. Finally, we leverage all these interesting features (adoption rates, retention rates, local social network adoption structure, and demographic information) to build a success predictor, that can distinguish between apps that will be successful a year into the future from ones that aren't.

In the sixth and final chapter we study how amateur and professionals mathematicians joined fores on the web to collaboratively solve open research problems in mathematics. We observe substantial differences in the important versus unimportant contributions made by contributors, and are thus able to predict whether or not a comment is important or not based on its sentence structure alone or the local responses it receives. We also observe interesting variations in the timing of communication and the reply structure between mathematicians, depending on whether they are collaborating to solve an open research problem, or a very challenging but non-research problem taken from a mathematical olympiad competition.

Part I

Group dynamics

CHAPTER 2

PHASE DIAGRAM FOR THE KURAMOTO MODEL WITH VAN HEMMEN INTERACTIONS

This chapter is written in collaboration with Ian Lizaragga and Steven Strogatz, and was published in Physical Review E*, Volume 89, Issue 1 in 2014.*

We consider a Kuramoto model of coupled oscillators that includes quenched random interactions of the type used by van Hemmen in his model of spin glasses. The phase diagram is obtained analytically for the case of zero noise and a Lorentzian distribution of the oscillators' natural frequencies. Depending on the size of the attractive and random coupling terms, the system displays four states: complete incoherence, partial synchronization, partial antiphase synchronization, and a mix of antiphase and ordinary synchronization.

2.1 Introduction

In 1967, Winfree [122] discovered that synchronization in large systems of coupled oscillators occurs cooperatively, in a manner strikingly analogous to a phase transition. In this analogy, the temporal alignment of oscillator phases plays the same role as the spatial alignment of spins in a ferromagnet. Since then, Kuramoto and many other theorists have deepened and extended this analogy [2, 70, 97, 109, 110].

Yet one question has remained murky. Can a population of oscillators with a random mix of attractive and repulsive couplings undergo a transition to an "oscillator glass" [28], the temporal analog of a spin glass [16]? Daido [27] simulated an oscillator analog of the Sherrington-Kirkpatrick spin-glass model [104] and reported evidence for algebraic relaxation to a glassy form of synchronization [29, 107, 108], but those results are not yet understood analytically. Others have looked for oscillator glass in simpler models with site disorder (where the randomness is intrinsic to the oscillators themselves, not to the couplings between them) [17, 28, 51, 55, 92]. Even in this setting the existence of an oscillator glass state remains an open problem.

In this paper we revisit one of the earliest models proposed for oscillator glass [17]: a Kuramoto model whose attractive coupling is modified to include quenched random interactions of the form used by van Hemmen in his model of spin glasses [116]. The model can now be solved exactly, thanks to a remarkable ansatz recently discovered by Ott and Antonsen [90]. Their breakthrough has already cleared up many other longstanding problems about the Kuramoto model and its offshoots [19, 51, 54, 55, 68, 72, 80, 83, 89, 96]. For the Kuramoto-van Hemmen model examined here, the Ott-Antonsen ansatz reveals that the model's long-term macroscopic dynamics are reducible to an eight-dimensional system of ordinary differential equations. Two physically important consequences are that the model does not exhibit algebraic relaxation to any of its attractors, nor does it have the vast number of metastable states one would expect of a glass. On the other hand, the frustration in the system does give rise to two states whose glass order parameter is non-zero above a critical value of the van Hemmen coupling strength. Our main results are exact solutions for the model's macroscopic states, their associated order parameters, and the phase boundaries between them.

The governing equations of the model are

$$\dot{\theta}_i = \quad \omega_i + \textstyle\sum_{j=1}^{N} K_{ij} \sin(\theta_j - \theta_i) \tag{2.1}$$

for $i = 1, \ldots, N \gg 1$, where

$$K_{ij} = \frac{K_0}{N} + \frac{K_1}{N}(\xi_i \eta_j + \xi_j \eta_i). \tag{2.2}$$

Here θ_i is the phase of oscillator i and ω_i is its natural frequency, randomly chosen from a Lorentzian distribution of width γ and zero mean: $g(\omega) = \gamma/[\pi(\omega^2 + \gamma^2)]$. By rescaling time, we may set $\gamma = 1$ without loss of generality. The parameters $K_0,\ K_1 \geq 0$ are the Kuramoto and van Hemmen coupling strengths, respectively. The random variables ξ_i and η_i are independent and take the values ± 1 with equal probability.

Simulations of the model (Fig. 2.1) show four types of long-term behavior. (1) *Incoherence* (Fig. 2.1(a)): When K_0 and K_1 are small, the oscillators run at their natural frequencies and their phases scatter. (2) *Partial locking* (Fig. 2.1(b)): If we increase K_0

while keeping K_1 small, oscillators in the middle of the frequency distribution lock their phases while those in the tails remain desynchronized. (3) *Partial antiphase locking* (Fig. 2.1(c)): If instead we increase K_1 while keeping K_0 small, the system settles into a state of partial antiphase synchronization, where half of the central oscillators lock their phases 180 degrees apart while the other half behaves incoherently. (4) *Mixed state* (Fig. 2.1(d)): If both K_0 and K_1 are sufficiently large and in the right proportion, we find a mixed state that combines aspects of the partially locked and antiphase locked states. But note two changes—the central oscillators that behaved incoherently in Fig. 2.1(c) now lock as in Fig. 2.1(b), and the antiphase locked oscillators of Fig. 2.1(c) are now less than 180 degrees apart.

These four states are not new. They were found and analyzed by Bonilla et al. [17] for a variant of Eq. (2.1) with a white noise term and a uniform (not Lorentzian) distribution of natural frequencies. The advantage of the present system is that the stability properties and phase boundaries of the four states can be obtained analytically. Figure 2.2 shows the resulting phase diagram.

We turn now to the analysis. As mentioned above, the Ott-Antonsen ansatz [90] has become standard, so we suppress the intermediate steps in the following derivation (but see [90] for details). The ansatz applies to (2.1) in the continuum limit and restricts attention to an invariant manifold that determines the system's long-term dynamics [89]. On this manifold the time-dependent density $\rho(\theta, t, \omega, \xi, \eta)$ of oscillators at phase θ with natural frequency ω and van Hemmen parameters ξ, η is given by

$$\rho = \frac{1}{2\pi}\left\{1 + \left[\sum_{n=1}^{\infty}(\alpha^* e^{i\theta})^n + \text{ c.c.}\right]\right\} \tag{2.3}$$

where $\alpha = \alpha(t, \omega, \xi, \eta)$ and the asterisk and c.c. denote complex conjugation. This density evolves according to

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial \theta}(\rho v) \quad = \quad 0 \tag{2.4}$$

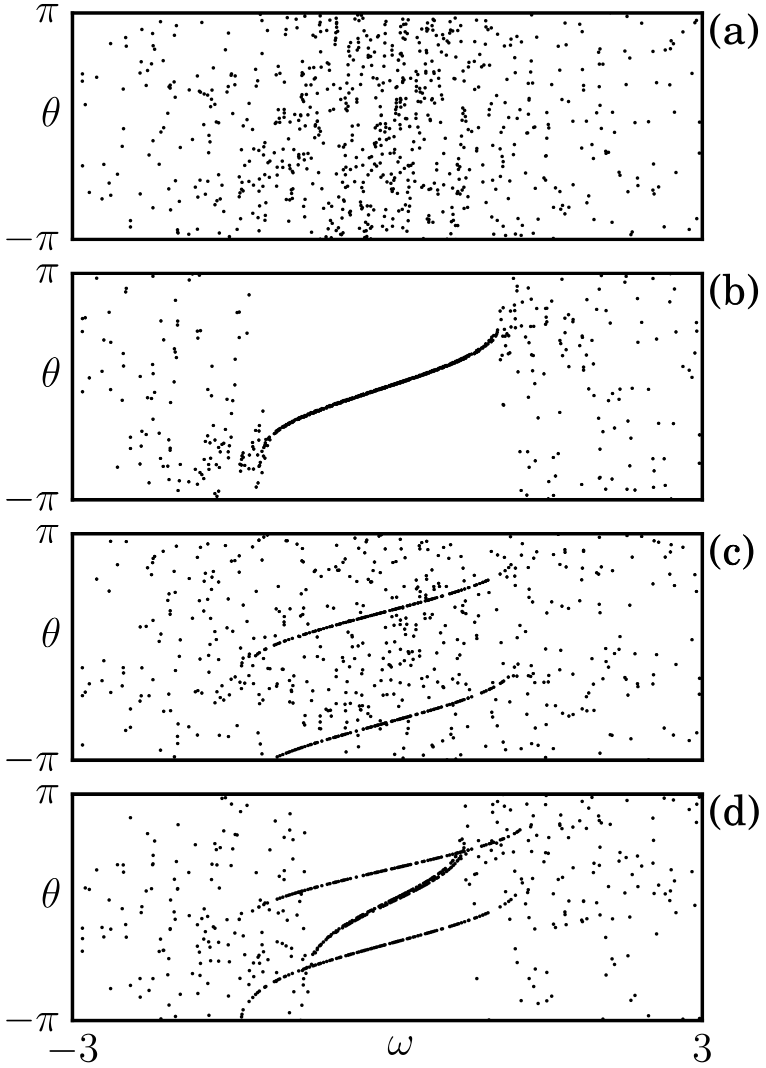


Figure 2.1: Statistical steady states for the Kuramoto-van Hemmen model. Equation (2.1) was integrated numerically for $N = 1000$ oscillators with Lorentzian distributed frequencies and random initial phases, using a fourth-order Runge-Kutta method with a fixed step size of 0.05. Parameter values: (a) Incoherence: $K_0 = 1, K_1 = 1$; (b) Partial locking: $K_0 = 2.5, K_1 = 1$; (c) Partial antiphase locking: $K_0 = 1, K_1 = 2.75$; (d) Mixed state: $K_0 = 2.5, K_1 = 2.75$. Only oscillators with $-3 \leq \omega \leq 3$ are shown.

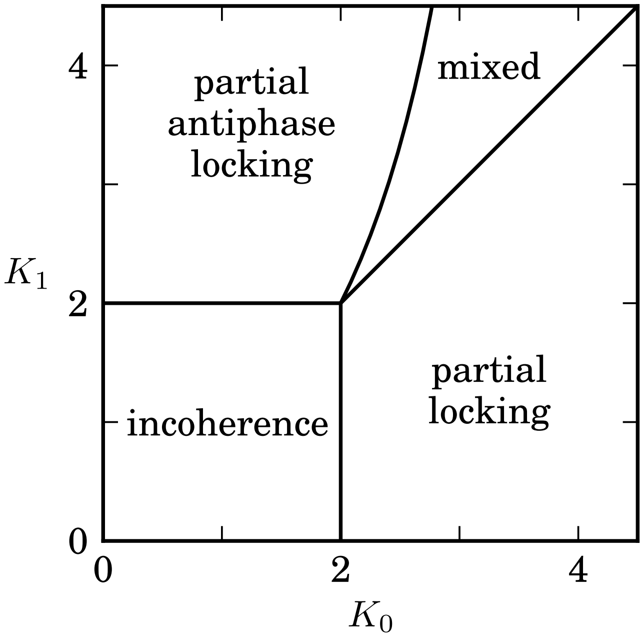


Figure 2.2: Phase diagram for (2.1), (2.2) with $g(\omega) = 1/[\pi(1+\omega^2)]$.

where $v = v(t,\omega,\xi,\eta)$ denotes the velocity field in the continuum limit,

$$v = \omega + \mathrm{Im}[e^{-i\theta}(K_0 Z + K_1 \xi W_\eta + K_1 \eta W_\xi) + \text{ c.c.}] \tag{2.5}$$

and the complex order parameters Z, W_ξ, and W_η are

$$\begin{aligned} Z &= \langle e^{i\theta} \rangle, \\ W_\xi &= \langle \xi e^{i\theta} \rangle, \\ W_\eta &= \langle \eta e^{i\theta} \rangle. \end{aligned} \tag{2.6}$$

The angle brackets $\langle \cdot \rangle$ denote integration with respect to the probability measure $\rho(\theta)d\theta\, g(\omega)d\omega\, p(\xi)d\xi\, p(\eta)d\eta$. The distribution p is normalized so that ξ and η equal ± 1 with equal probability $\frac{1}{2}$.

When (2.3) and (2.5) are inserted into (2.4), one finds that the dependence on θ is satisfied identically if $\alpha(t,\omega,\xi,\eta)$ evolves according to:

$$\begin{aligned}\dot{\alpha} &= -\frac{\alpha^2}{2}\left[K_0 Z^* + K_1\left(\xi W_\eta^* + \eta W_\xi^*\right)\right] + i\omega\alpha \\ &\quad + \frac{1}{2}[K_0 Z + K_1(\xi W_\eta + \eta W_\xi)]. \end{aligned} \tag{2.7}$$

This system is infinite-dimensional, since there is one equation for each real ω. But its macroscopic dynamics are governed by a much smaller, finite-dimensional set of ODEs. The reduction occurs because the different $\alpha(t,\omega,\xi,\eta)$ in (2.7) are coupled only through the order parameters Z, W_ξ, and W_η. Those order parameters in turn are expressible, via (2.6), as integrals involving ρ and therefore α itself. Under the usual analyticity assumptions [90] on α, the various integrals can be expressed in terms of a finite set of α's, and these obey the promised ODEs, as follows.

Consider $Z = \int e^{i\theta}\rho(\theta)d\theta\, g(\omega)d\omega\, p(\xi)d\xi\, p(\eta)d\eta$. To calculate this multiple integral, first substitute (2.3) for ρ and perform the integration over θ to get $Z = \int \alpha\, g(\omega)d\omega\, p(\xi)d\xi\, p(\eta)d\eta$. Second, evaluate the integral $\int_{-\infty}^{\infty}\alpha\, g(\omega)d\omega$ by considering ω as a complex number and computing the resulting contour integral, choosing the contour to be an infinitely large semicircle closed in the upper half plane. The Lorentzian $g(\omega) = 1/[\pi(1+\omega^2)]$ has a simple pole at $\omega = i$, so the residue theorem yields

$$\int_{-\infty}^{\infty}\alpha\, g(\omega)d\omega = \alpha(t,i,\xi,\eta). \tag{2.8}$$

Third, integrate over ξ and η. Since these variables take on the values ±1 with equal probability, Z receives contributions from four subpopulations: (ξ,η)=$(+1,+1)$, $(+1,-1)$, $(-1,+1)$, and $(-1,-1)$. If we define the sub-order parameters for these subpopulations as

$$\begin{aligned}
A(t) &= \alpha(t,i,+1,+1) \\
B(t) &= \alpha(t,i,-1,-1) \\
C(t) &= \alpha(t,i,+1,-1) \\
D(t) &= \alpha(t,i,-1,+1), \qquad (2.9)
\end{aligned}$$

we find that Z is given by

$$Z = \frac{1}{4}(A+B+C+D). \qquad (2.10)$$

Similar calculations show that the glass order parameters can also be expressed in terms of A,B,C,D:

$$\begin{aligned}
W_\xi &= \frac{1}{4}(A-B+C-D), \\
W_\eta &= \frac{1}{4}(A-B-C+D). \qquad (2.11)
\end{aligned}$$

The sub-order parameters A,B,C,D have physical meanings. For example, A can be thought of as a giant oscillator, a proxy for all the microscopic oscillators with $(\xi,\eta) = (+1,+1)$. Likewise, B,C and D represent giant oscillators for the other subpopulations.

The equations of motion for these giant oscillators are obtained by inserting (2.10), (2.11) into (2.7) and analytically continuing to $\omega = i$. The result is the following closed system:

$$\begin{aligned}
\dot{A} &= -\frac{1}{2}A^2[K_0Z^* + \frac{K_1}{2}(A^* - B^*)] - A \\
&\quad + \frac{1}{2}[K_0Z + \frac{K_1}{2}(A - B)] \\
\dot{B} &= -\frac{1}{2}B^2[K_0Z^* + \frac{K_1}{2}(B^* - A^*)] - B \\
&\quad + \frac{1}{2}[K_0Z + \frac{K_1}{2}(B - A)] \\
\dot{C} &= -\frac{1}{2}C^2[K_0Z^* + \frac{K_1}{2}(D^* - C^*)] - C \\
&\quad + \frac{1}{2}[K_0Z + \frac{K_1}{2}(D - C)] \\
\dot{D} &= -\frac{1}{2}D^2[K_0Z^* + \frac{K_1}{2}(C^* - D^*)] - D \\
&\quad + \frac{1}{2}[K_0Z + \frac{K_1}{2}(C - D)]. \qquad (2.12)
\end{aligned}$$

Since A, B, C, and D are complex numbers, the system (2.12) is eight-dimensional.

The four steady states shown in Fig. 2.1 correspond to four families of fixed points of (2.12), each of which is characterized by a simple configuration of A, B, C, D in the complex plane. Figure 2.3 plots those four families schematically on the phase diagram, showing where each exists and is linearly stable. We discuss them in turn.

The incoherent state of Fig. 2.1(a) corresponds to the fixed point at the origin, $A = B = C = D = 0$, with order parameters $Z = W_\xi = W_\eta = 0$. It exists for all $K_0,\ K_1 \geq 0$ but is linearly stable iff (if and only if) $K_0 < 2$ and $K_1 < 2$. This stability region is shown as the square in the lower left of Fig. 2.3.

The partially locked state (Fig. 2.1(b)) corresponds to a configuration where A, B, C and D all equal the same nonzero complex number, as shown in the lower right panel of Fig. 2.3. By rotational symmetry, we can assume that $A = B = C = D = R_{PL} > 0$. Such a state is a fixed point of (2.12) iff $K_0 > 2$ and $R_{PL} = \sqrt{1 - 2/K_0}$, in which case it is linearly stable iff $K_1 < K_0$. (There is a trivial zero eigenvalue associated with the rotational symmetry, so what we really mean is that the state is linearly stable to all perturbations other than rotational ones. Likewise, there is a whole circle of partially

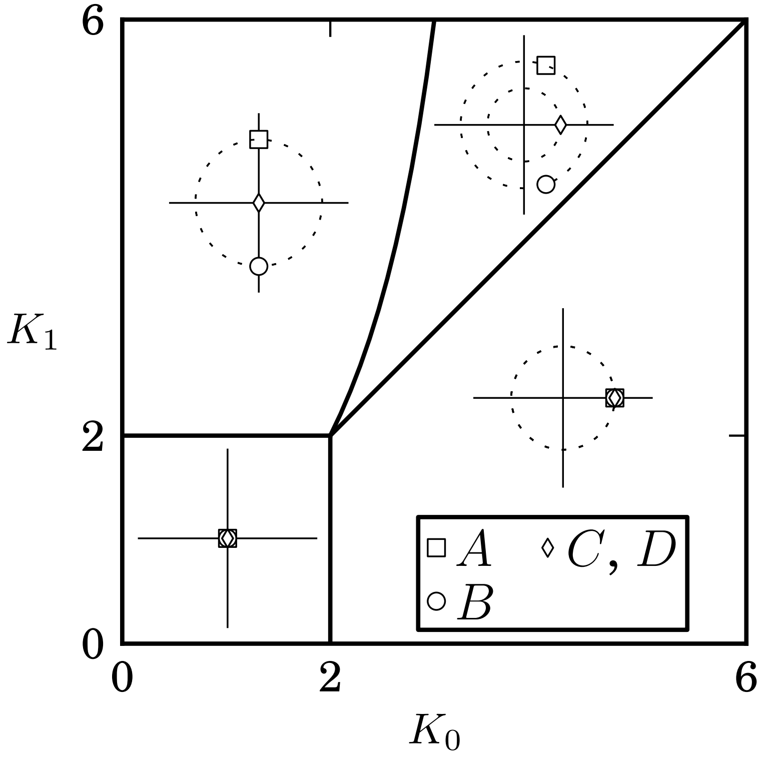


Figure 2.3: Stable fixed points A, B, C, D for the four states. In each panel, the axes show the region of the complex plane with $-1 \leq \text{Re}(z) \leq 1$ and $-1 \leq \text{Im}(z) \leq 1$. Rotationally equivalent fixed points lie on the dashed circles.

locked states, all equivalent up to rotation, as indicated by the dashed circle in the lower right panel of Fig. 2.3.) The order parameters are $Z = \sqrt{1 - 2/K_0}$ and $W_\xi = W_\eta = 0$.

The antiphase state (Fig. 2.1(c)) corresponds to a fixed point where $A = -B = R_A > 0$ and $C = D = 0$. It exists iff $K_1 > 2$ and $R_A = \sqrt{1 - 2/K_1}$. When it exists it is linearly stable iff

$$K_0 < 4K_1/(2 + K_1). \tag{2.13}$$

Finally, the mixed state (Fig. 2.1(d)) corresponds to a configuration where $A = B^*$ and $C = D = R_M > 0$. It exists iff $K_1 > 2$ and $4K_1/(2 + K_1) < K_0 < K_1$ (the wedge in the upper right of Fig. 2.3) and satisfies

$$\begin{aligned}\mathrm{Re}(A) &= \frac{K_0}{2K_1 - K_0}\sqrt{1+\frac{2}{K_1}-\frac{4}{K_0}} \\ \mathrm{Im}(A) &= 2\sqrt{\frac{(K_1-K_0)(K_1(K_1-2)+K_0)}{K_1(2K_1-K_0)^2}} \\ R_M &= \sqrt{1+\frac{2}{K_1}-\frac{4}{K_0}}. \end{aligned} \tag{2.14}$$

We were unable to find the eigenvalues analytically in this final case, but we verified linear stability numerically for a sample of mixed states up to $K_1 = 10^5$.

All the transitions in Fig. 2.3 are continuous (Fig. 4). In particular, the mixed state morphs into the antiphase state on the left side of its stability region, and into the partially locked state on the right side. To verify this, observe that the configuration of A, B, C, D in the mixed state, as parametrized by Eq. (2.14), continuously deforms into the states on either side of it as (K_0, K_1) approaches the relevant stability boundary.

The glass order parameters W_ξ and W_η are nonzero for the antiphase and mixed states, so in that specific sense the model can be said to exhibit a glassy form of synchronization [17]. Moreover, $W_\xi = W_\eta$ for all four states, which confirms a conjecture of Bonilla et al. [17]. On the other hand, the oscillator model (2.1), (2.2) lacks other defining features of a glass, such as a large multiplicity of metastable states and non-exponential relaxation dynamics; the same is true of the original van Hemmen spin-glass model [20].

Experimental tests of the phase diagram predicted here may be possible in a variety of oscillator systems with programmable coupling. Prime candidates are optical arrays [47] or populations of photosensitive chemical oscillators [114] in which the interactions are mediated by a computer-controlled spatial light modulator.

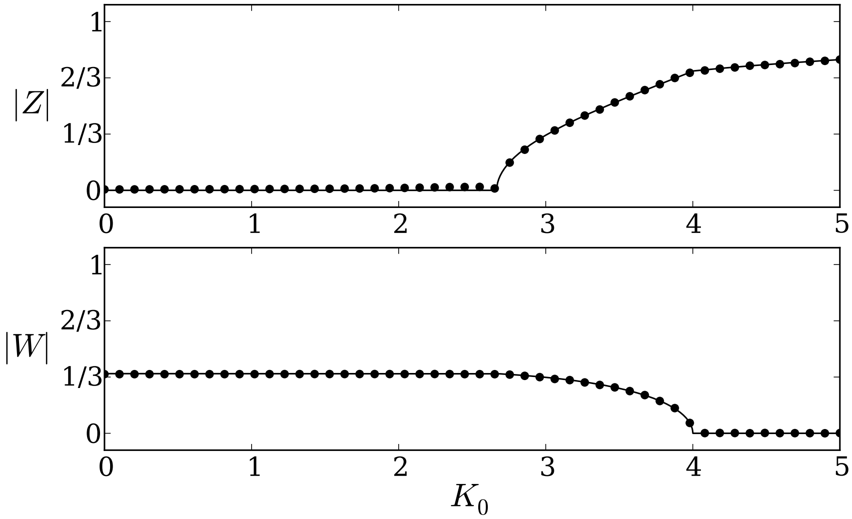


Figure 2.4: Theory vs. simulation for order parameters. Solid line, exact results; circles, simulations for N=50,000 oscillators. For K_1=4, Eq. (2.1) was integrated using an Euler method with step size 0.01. Each combination of $(\xi,\eta)=(\pm 1,\pm 1)$ was assigned $N/4$ oscillators, with natural frequencies taken from a deterministic Lorentzian distribution: $\omega_i = \tan[(\pi/2)(2i-n-1)/(n+1)]$, for $i=1,\ldots,n$ and n=$N/4$. The values of the order parameters are shown at $t=200$, by which time convergence to a statistical steady state has occurred.

CHAPTER 3
COMMUNITY MEMBERSHIP IDENTIFICATION FROM SMALL SEED SETS

This chapter is written in collaboration with Jon Kleinberg and was published in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining in 2014.

In many applications we have a social network of people and would like to identify the members of an interesting but unlabeled group or community. We start with a small number of exemplar group members – they may be followers of a political ideology or fans of a music genre – and need to use those examples to discover the additional members. This problem gives rise to the seed expansion problem in community detection: given example community members, how can the social graph be used to predict the identities of remaining, hidden community members? In contrast with global community detection (graph partitioning or covering), seed expansion is best suited for identifying communities locally concentrated around nodes of interest. A growing body of work has used seed expansion as a scalable means of detecting overlapping communities. Yet despite growing interest in seed expansion, there are divergent approaches in the literature and there still isn't a systematic understanding of which approaches work best in different domains.

Here we evaluate several variants and uncover subtle trade-offs between different approaches. We explore which properties of the seed set can improve performance, focusing on heuristics that one can control in practice. As a consequence of this systematic understanding we have found several opportunities for performance gains. We also consider an adaptive version in which requests are made for additional membership labels of particular nodes, such as one finds in field studies of social communities. This leads to interesting connections and contrasts with active learning and the trade-offs of exploration and exploitation. Finally, we explore topological properties of communities and seed sets that correlate with algorithm performance, and explain these empirical observations with theoretical ones.

We evaluate our methods across multiple domains, using publicly available datasets with labeled, ground-truth communities.

3.1 Introduction

There are many settings in which we are interested in accessing or studying a group of people in a social network, but instead of the full membership of the group we know only a few examples. A natural goal in this case is to expand these examples into a larger set that approximates the full extent of the group, and this goal has been the focus of recent work on *seed set expansion* in networks. Phrasing the problem slightly informally for purposes of this discussion, we are given a graph G that contains a group of nodes C whose identities we'd like to uncover, and we are told the identities of a small subset S of C. Given a budget k, can we find k additional nodes such that as many of them as possible come from C?

Examples of the seed set expansion problem are numerous. For example, recent work studying political activism has started from a small set of representative of each competing ideology, collected through detailed field work, and then attempted to expand these representatives into the larger groups that they come from [119]. Recommendation tools for forming on-line groups have the potential to collect a few initial suggestions from a user and then produce a longer list of recommended group members. Similarly, a marketer may want to expand a set of a few interested consumers of a product into a longer list of people who might also be interested in the product. Seed set expansion has also been used to infer missing attributes in user profile data [82] and to detect e-mail addresses of spammers [123]. Nor are the applications limited to social networks; as we will see below, we could ask similar questions in which we start with a few items such as products for sale, and we then attempt to use a co-purchase network to expand these items into a product category that contains all of them.

It is useful to note a few properties of the seed set expansion problem, consistent with these sources of motivation. First, we focus on cases in which the expansion is guided by an underlying graph structure — the basic premise is that if a person or item *v* is “tightly linked” in the graph to many members of the group *C*, then this provides evidence that *v* may also be a member of the group. Second, the goal in the seed set expansion problem as it has been studied in prior work — and the goal we pursue here — is neither to find the full extent of the group *C* nor to sample uniformly from it, but instead to “collect” a fixed number of members from it with as little error as possible.

The present work: Principles for seed set expansion The number of approaches to seed set expansion has proliferated rapidly, but there is still very little understanding of the principles through which we can reason about trade-offs between different approaches or the types of instances on which we can expect good performances. In this work we seek to begin developing some of the principles underlying the seed set expansion problem. Among the questions that guide our study, we consider the following. Do certain approaches to seed set expansion produce consistently better results than others, across a range of domains? Can characteristics of the initial seed set *S* help us understand when seed set expansion will be effective? And how do structural characteristics of the group *C* affect the quality of the solution?

We can go further in our analysis by taking into account the following issue. If we think about many of the applications that motivate the seed set expansion problem, there is a potentially rich interaction available between the expansion algorithm and the “expert” who can recognize members of the group *C*. Consider for example the problem of identifying members of political movements noted earlier [119]. Here there is a domain expert who has provided the initial representatives of a group, and if we are trying to expand these representative members into a larger set, we may well have the ability to adaptively query the expert — a few nodes at a time — and make future choices based on the result of this feedback. There is thus an opportunity to incorporate such interaction between the algorithm and the domain expert into the formalism of seed set expansion. Such interaction clearly has a structure similar to work in active learning, although we should emphasize that unlike traditional work in that domain, we are not seeking a classification of the full underlying graph, nor do we have a subset of the data available for training; rather, we want to collect a set of nodes from *C* based only on the initial examples *S*.

The present work: Overview of results We first consider a wide range of techniques that have been used in prior work for seed set expansion, applying them to three main datasets: two social networks (a co-authorship network among researchers and the YouTube social network among its users), and a product co-purchase network in which the groups are product categories. We find that measures based on PageRank are by far the most effective. Moreover, almost all of the performance gains from PageRank come from running just two or three iterations of the PageRank update rule — a finding that is novel to the best of our knowledge, and consistent with our analysis of where PageRank is achieving most of its relative performance gains over more local neighbor-based methods, on nodes that are outside the immediate vicinity of the seed set S.

We then consider how properties of the seed set affect performance. In thinking about trade-offs here, it is useful to consider the interaction between the expansion algorithm and the domain expert discussed above: in that context, our motivation is to identify how a domain expert should best use their knowledge to compile a seed set of members. In practice we are highly constrained by those community members of which a domain expert has knowledge. One natural question is to ask whether performance is better when S consists of the highest-degree nodes in C or a uniformly random subset of C. In choosing a large degree seed set, we model a domain expert who returns a list of the most popular or most famous nodes in the network. In choosing a random seed set, we model a domain expert who returns a seed set that is representative of the community, to wit, one potentially consisting of high and low degree nodes. We also consider the effect of seed set size, exploring a basic trade-off: if the seed set is too large a fraction of the group, it can be hard to find the remaining members, but if it is too small, then it is not providing a sufficiently useful set of examples for the full extent of the group.

We similarly look at trade-offs in the structural properties of the group C, finding that denser groups — those with a higher ratio of edges to nodes — tend to result in better performance for seed set expansion.

Finally, we look at different ways of managing the interaction between the algorithm and the domain expert. We find contexts in which regularly interspersing expan-

Dataset	Nodes	Edges	Communities
DBLP	317080, authors	1049866, co-authorship	13477, conferences
Amazon	334863, products	925872, co-purchased	151037, product categories
YouTube	1134890, users	2987624, friendship	8385, user-defined groups

Table 3.1: The number and substantive interpretation of nodes, edges, and communities in each network. All sourced from [128].

sion steps with queries to the expert can outperform approaches in which the queries are batched in larger blocks. We also find that for our objective function of collecting members of C as quickly as possible, asking the expert about nodes on the "margin" of C can be effective in finding the boundaries of the group, but this benefit is more than offset by the downside of querying the expert about a greater number of nodes that turn out not to be in C.

3.2 Setup: data, performance, algorithms

Data We use network data with ground-truth community membership from the Stanford Network Analysis Project (snap.stanford.edu). Table 6.1 gives a summary of the datasets used in this paper; see Yang & Leskovec [128] for additional background on these datasets.

Seed Sets and Performance We are given a graph G that contains a collection of potentially overlapping communities $\mathscr{C}$, and we have an interest in a particular community $C \in \mathscr{C}$. We are given a set of labeled community members $S \subset C$. Thus $C - S$ consists of the unlabeled, not-yet-discovered community members. We have a budget to make a prediction of size k, and we will call the prediction P. We wish to maximize the recall,

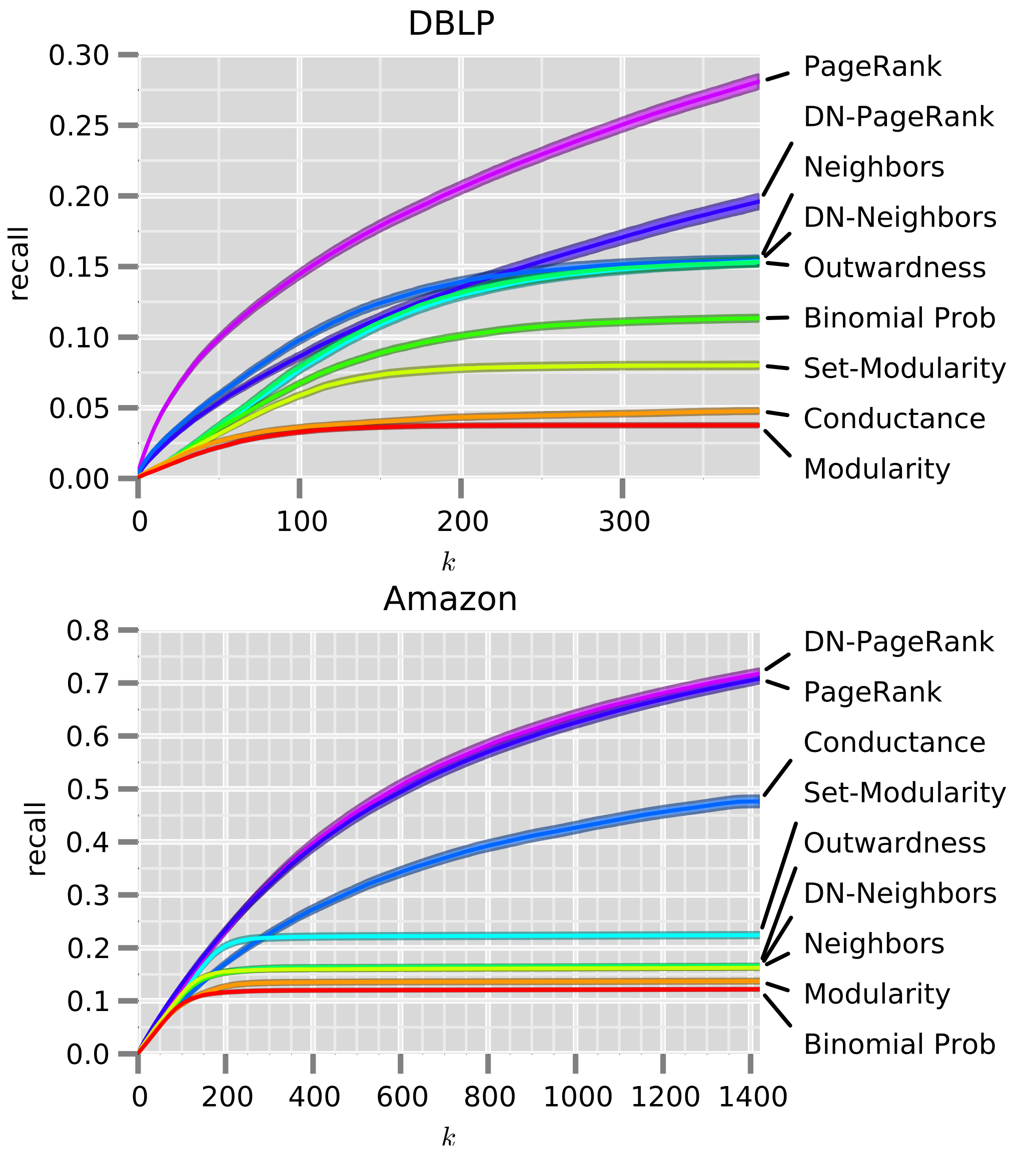


Figure 3.1: Recall averaged over $\mathscr{C}_{3/4}^{600}$. Rankings for YouTube are the same as for DBLP. The envelopes represent two standard errors centered about the mean.

$|P \cap C|/|C - S|$, i.e. the fraction of the unlabeled community recovered by the algorithm.

Unless otherwise specified we choose S to be a random subset of C of size $|C|/10$.

Communities From Table 6.1 we see that there are roughly 10^4-10^5 labeled, ground-truth communities in each dataset. These communities vary in size from 6 to roughly 10^4. In this paper we focus on the 600 communities closest in size to $m^{3/4}$, where m is the size of the largest community; let us call this set $\mathscr{C}^{600}_{3/4}$.[1]

Stopping criteria Here we model the scenario in which a researcher knows that there are approximately k community members, and so they select the top k results from their choice algorithm, A, as the predicted community. That is, we choose a simple stopping criterion of a fixed number of guesses equal to the size of the central value; for example on $C^{600}_{3/4}$ we fix $k = m^{3/4}$ and set our prediction P to be the k top nodes according to A's ranking. As we will see in Figures 3.1, 3.3.3, and 3.5, the relative performance rankings are not very sensitive to the choice of k.

This stopping rule has the advantage that it is not sensitive to the topology of the prediction P, which would be undesirable given that the algorithms we compare produce communities with a variety of typical topologies (this is discussed in detail in [1]). We discuss alternate stopping criteria in §3.7.

[1] Given the same number of guesses, it would be unfair to compare the recall of an algorithm on a community of size 10000 with one of size 100. We find that there are 600 communities centered about this log-space third-quartile all close enough in size that such biases do not taint our results. Rather, thanks to the large number of moderately sized communities, we are able to estimate performances with good standard error estimates. In point of fact, good statistical convergence that distinguishes the various algorithms can be achieved with only 20 communities, and so our consideration of all 600 provides a large extra margin.

3.3 Results: prediction algorithms

3.3.1 PageRank's success

Figure 3.1 shows the recall values for a wide range of algorithms detailed in §3.7 and the appendix. Variants of PageRank — in which we rank by the stationary probability of a random walk with restarts originating at the seed set — are the clear winners. This is consistent with PageRank's success in other applications, but it is nonetheless perhaps surprising that it is so much more powerful than other methods that have been used for this problem. We also note that in two of our three domains, pure unnormalized PageRank significantly outperforms variants such as degree-normalized PageRank (DN-PageRank); this poses an interesting contrast to the fact that DN-PageRank rather than pure PageRank has typically been the preferred method for seed set expansion.

We now consider three questions that suggest insights into the structure of the problem and how to the use these approaches in practice.

1. Why does PageRank outperform other methods such as neighbor counting?
2. In computing PageRank with the power method, how many iterations (random walk steps) does it need to take to achieve this high performance?
3. Could variations beyond DN-PageRank and PageRank achieve even better performance?

3.3.2 Whom does PageRank find?

In addition to the success of PageRank and its variants in Figure 3.1, it is also striking to see how PageRank climbs smoothly with k in contrast with neighbor-counting methods that flatten abruptly as we increase.

Looking into this behavior helps us understand where PageRank gets some of its power. In particular, we ask which true positives are found by PageRank compared to Neighbors, categorizing based on whether they belong to ego(S), the set of nodes

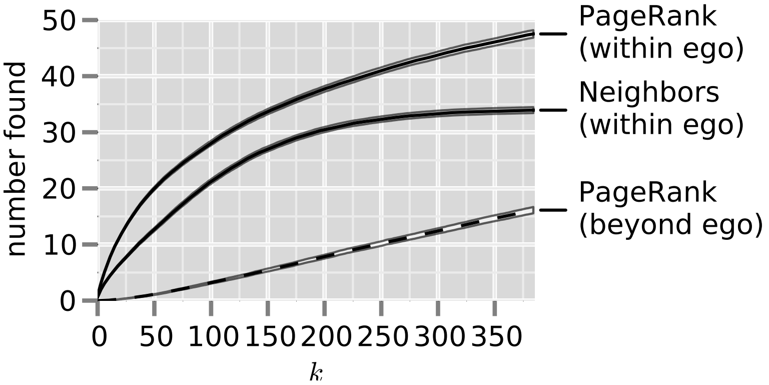


Figure 3.2: Number of community members found within ego(S), the seed set's ego network, and outside of it for positive PageRank and positive neighbor counting. Results are averaged over $\mathscr{C}^{600}_{3/4}$. Envelopes represent two standard errors centered about the mean.

directly adjacent to the seed set S. Neighbor-counting methods cannot effectively use information about nodes outside ego(S). Is this hindrance the sole factor underlying PageRank's advantage over Neighbors?

Figure 3.2 untangles this issue: first, we see that PageRank's rate of discovery of members within ego(S) is significantly higher than that of Neighbors; second, we see that it finds true positives outside the ego at a constant linear rate. So in addition to having this broad reach beyond the ego, which we expected, PageRank is even better at identifying which members of the seed set's immediate neighbors are true positives. PageRank's success over neighbor-counting is thus both inside *and* outside ego(S).

3.3.3 How many steps does it take to get to the community?

Next we consider a question regarding PageRank itself: in computing PageRank with the power method, how many random walk steps are needed for PageRank to realize its maximum performance? This is a basic question about PageRank's iterative nature, and the concrete performance measures underlying our problem formulation make it natural to evaluate the question in this context.

The results in Figure 3.3.3 indicate that after only three random walk steps PageRank's performance has converged to its upper limit, and it is already close to this limit after two steps. It is striking that most of PageRank's power on these networks comes from just its first few iterations. To appreciate this we consider PageRank's interpretation at each step and the corresponding performance. 0-step PageRank represents random guessing. 1-step PageRank is closely related to DN-Neighbors[2] — and indeed the performance curve of 1-step PageRank has the same "flattening out" that stood out in the performance curve for neighbor-counting, as well as a comparable final performance value. 2-step PageRank reaches one step beyond ego(S) and in Figure 3.3.3 we see that in the transition from 1 to 2 steps PageRank's performance exhibits a dramatic increase, nearly reaching its full potential. This indicates that most of the members found by PageRank are within 2-steps of the seed set. Finally, 3-step PageRank yields PageRank's full potential, and t-step PageRank continues at this level as $t \to \infty$.

3.3.4 Variations on PageRank

To normalize or not to normalize Not to normalize. As mentioned, essentially the only PageRank-derivative used in the literature for community detection by seed set expansion has been DN-PageRank [8, 100, 121, 128]. Yet PageRank yields much higher performance than degree-normalized PageRank in DBLP and Youtube and they reach a tie on the Amazon network.

In Figure 3.1 we find that in DBLP and Youtube (not pictured) unnormalized PageRank, or simply PageRank, find true community members with greater accuracy than degree-normalized PageRank does. This performance increase is robust after controlling for and considering all community sizes, and it is true in both "easy to detect" and "hard to detect" communities. Indeed, PageRank is best or tied for best on roughly 80% of the communities, on an instance-by-instance basis.

On the Amazon product network, in contrast, PageRank and DN-PageRank reach a

[2] But instead of normalizing by the target's user degree, the normalization happens with respect to the outgoing nodes

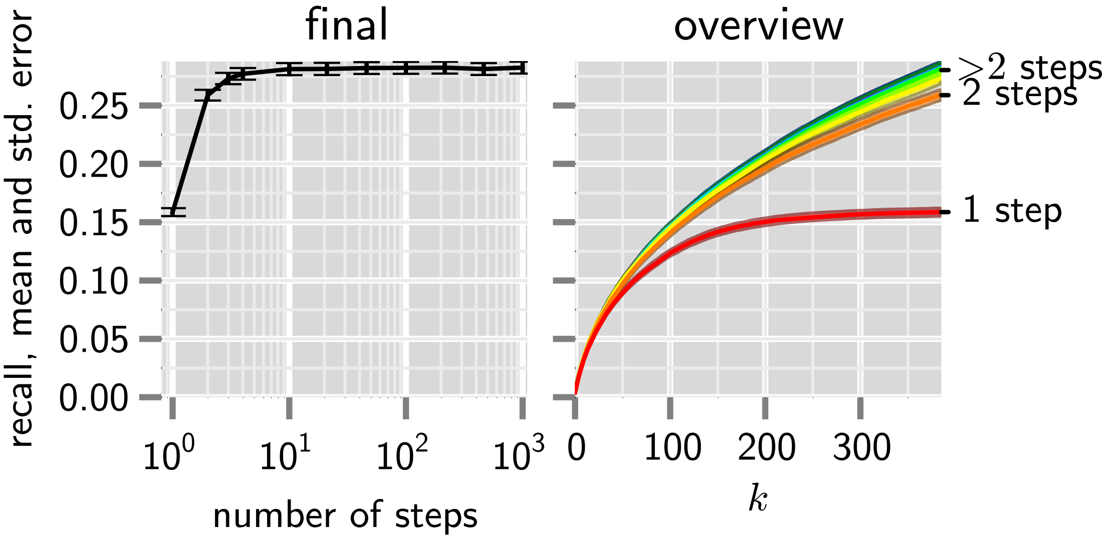


Figure 3.3: Comparison of PageRank performance for a variety of walk lengths. For each community the same random seed set was used as the walk length was varied. Results are averaged over $\mathscr{C}^{600}_{3/4}$ and the envelopes represent two standard errors centered about the mean.

statistical tie. It would be interesting to understand the differences in domain that lead to this, including the natural contrast that Amazon is a network on items for purchase, rather than a social network on people as in both DBLP and YouTube.

A continuum: degree-normalization to amplification Finally we note that DN-PageRank and PageRank are two special cases of using the sorting metric $\rho \cdot d^x$ with $x = -1$ for normalization and $x = 0$ for pure PageRank. It's therefore natural to consider the performance for all values of x, and we show this for all three datasets in Figure 3.4.

As we see there, the optimal exponents for DBLP and YouTube are both close to 0, indicating the power of unnormalized PageRank on these two social networks, whereas in Amazon the results were statistically indistinguishable for exponents $x \in [-1, 0]$. It is interesting to note that the optimal exponent x in DBLP is in fact slightly positive — in other words, rather than normalizing PageRank, the optimal strategy is to inflate it

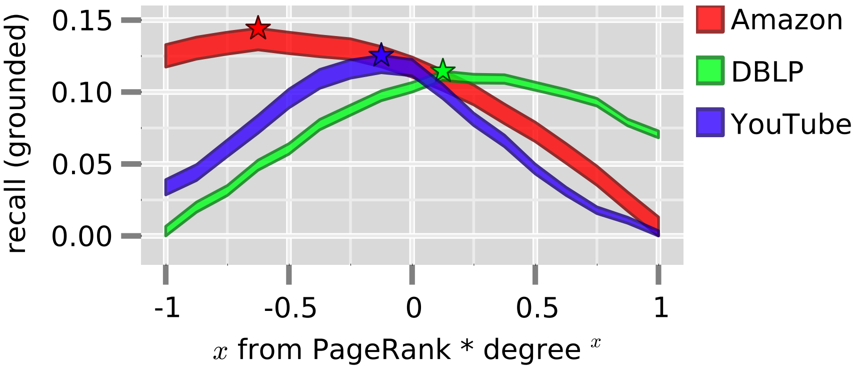


Figure 3.4: Mean performance as a function of x, where x is a variable exponent in the sorting heuristic PageRank*degree^x. The performances have been shifted vertically such that the lowest performance is grounded at 0, resulting in shifts for Amazon, DBLP, and YouTube of 0.52, 0.18, and 0.049, respectively. The star symbol indicates a curve's maximum. Results are averaged over $C^{600}_{3/4}$ and the envelopes represent two standard errors centered about the mean.

slightly. This can perhaps be motivated by the fact that many of the false positives being recovered by the algorithm are low degree nodes.[3]

3.3.5 Combining Multiple Measures

A natural extension of the current framework is to treat each of these network measures as a feature, and to choose nodes for C by training a classifier on the labeled examples and classifying the remaining nodes each according to their corresponding feature vec-

[3] Who are the false positives?
Note that because the communities in these datasets are overlapping, the nodes recovered by the algorithm should really be classified as being one of three types: true positives, false positives, and *neutral positives*. Neutral positives are nodes that are in some community together with the seed node, simply not in the community of interest C. In that sense, when the algorithm recovers a neutral positive it is accurately discovering information about the graph's community structure. If we relabel the original group of false positives into neutral positives or false positives, we find that in DBLP the 'real' false positives have a much lower average degree and very low variance in degree compared to the neutral or true positives. This distinction is not evident if one only considers the original binary labeling of being in the target community or not. That PageRank is making most of its big mistakes on low-degree nodes motivates the slight degree-amplification that we see is optimal for DBLP in Figure 3.4.

tor. We tried this using a support vector machine (SVM) on a feature in which node v's features are the values of the network measures (PageRank, Neighbors, etc.) evaluated on v. For example, for a simple classifier that combines Neighbors and PageRank, the feature vector for a node u was $[ego(u), \rho(u)]$ and the feature matrix used to train the classifier was $(|S|+|N|) \times 2$. The result is interesting in the negative direction: we were not able to realize any performance gains by combining multiple measures. Rather, the higher-dimensional classifiers performed only as well as its best-performing single-dimensional classifier submember. For example, the 2D SVM consisting of PageRank and Neighbors performed as well as PageRank, and the 2D SVM consisting of Neighbors and BinomProb performed as well as Neighbors.

To construct the classifier we build a feature vector for each of the PageRank-based and Neighbor-based methods, e.g. all the algorithms except Conductance and Modularity. (The latter were excluded because they do not assign attributes for every node in the graph – only for ones local to the seed set.) We build the feature vectors by seeding each of the algorithms with 25% of the input nodes; we reserve such a large fraction so as to emphasize teaching the algorithm about the attributes of 'hidden' positive members, rather than seeded ones (which will typically have much larger values of, for example, PageRank). To choose the C value for the SVM classifier we perform 3-fold cross validation with a 75/25 train/test split. We consider linear and radial basis function kernels and normalize all features to have unit $||L_2||$ norm before training.

Negative examples and information Note that to train the SVM we require both positive and negative examples, and so for the learning framework we introduce the notion of a negative seed set, T. Much like S, the seed set of known community members, T consists of nodes that are known from the outset (e.g. thanks to a domain expert's knowledge) to be non-members. To choose the negative seed set we tested the same heuristics as we did for the positive seed set, namely random nodes as in §3.2 and higher degree nodes as in see §3.4.1.

The introduction of the negative examples lead us to consider the possibility that the information about their non-membership could help improve classification. For example, just as we expect nodes tightly knit with the positive seed set S to more likely be members themselves, we expect nodes tightly knit to the negative seed set T to be less likely to be members.

We used SVMs to empirically verify both of these intuitions. That is, we seed PageRank with the negative seed set T and call the resulting metric on the nodes Negative-PageRank. For the purposes of this discussion, we call the original PageRank seeded with S Positive-PageRank. We then train an SVM using these two attributes as node features, and find that the SVM's weight vector has a positive coefficient for the Positive-PageRank feature and a negative coefficient for the Negative-PageRank feature, as expected.

However, the introduction of Negative-PageRank ultimately had no significant effect, neither improving nor hurting the performance of the classifier. The same is true regarding analogous versions of Negative-Neighbors and Negative-BinomProb).

3.4 Results: seed sets

Having looked at the relative performance of different algorithms for seed set expansion, we now consider the effect that different structural properties of the seed set itself can have on performance.

3.4.1 Heuristics for seed set selection

We begin by considering the effect of the node degrees in the seed set. In Figure 3.5 we see that for seeding PageRank it is highly advantageous to use a random positive seed set compared with one consisting of high-degree nodes. Though we have not pictured it here, this result holds for all domains, community sizes, and high-performing algorithms. It is true for the neighbor counting metrics as well (with the exception of binomial probability), however for the neighbor counting metrics the improvement is not as striking.

In many settings we should expect to have relatively little control over which mem-

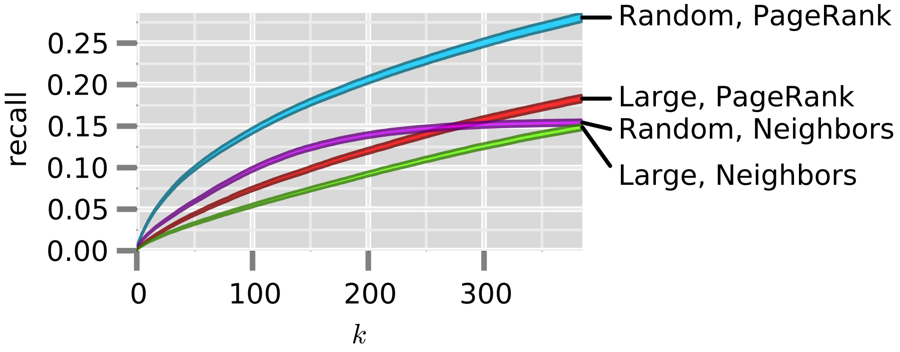


Figure 3.5: Comparison of algorithm performance for positive seed sets composed of random versus high-degree nodes, using PageRank and Neighbors. Results are averaged over $\mathscr{C}_{3/4}^{600}$ from DBLP and the envelopes represent two standard errors centered about the mean.

bers are in the seed set: the community is hidden to us and the seed set consists of those members for whom we happen to have labels. However, the particular contrast we analyze here, between random and high-degree nodes, corresponds naturally to two distinct scenarios for interacting with a domain expert: if the expert knows the most popular or most famous nodes in the community, this would lead to a high-degree seed set, while if the expert returns a more representative seed set, this would be modeled by a set consisting of random nodes. Some experience indicates that experts will often have a tendency to identify high-degree members, which, we see here, is not in fact the most effective way to gather a seed set for further expansion.

This lesson is an important heuristic to consider. Given that our recommendation is to use PageRank over more local methods when possible, it would also be advantageous for domain experts to heuristically search for nodes with a more diverse degree distribution, rather than searching for and validating the membership of those with highest degree. Note that even when this is not possible, and PageRank is seeded with a large degree seed set, it still outperforms Neighbors as a method.

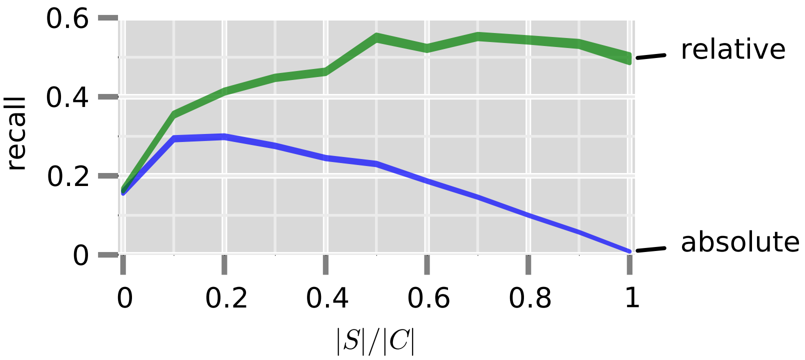


Figure 3.6: Performance as a function of the fraction of the community used to seed PageRank for all DBLP communities. For each community C ten seed sizes were tested with uniform spacing between 1 and $\lfloor 0.99|C| \rfloor$. We distinguish between two types of recall: relative recall is $|P \cap C|/|C-S|$, i.e. the fraction of unlabeled nodes that were discovered; whereas absolute recall is $|P \cap C|/|C|$, the fraction of the total community recovered during the evaluation stage. Envelopes represent two standard errors centered about the mean.

3.4.2 Seed set size and performance

In Figure 3.6 we examine the algorithm's performance as a function of the fraction of the community C used in the seed set S, $|S|/|C|$. We evaluate performance using two measures of recall: the *relative recall* $|P \cap C|/|C-S|$, and the *absolute recall* $|P \cap C|/|C|$. We find that the relative recall eventually plateaus as $|S|/|C|$ is increased whereas the absolute recall has an interior maximum.

Intuitively we expect this interior maximum in absolute performance: by starting off with very little of the community C we will be lacking sufficient information about C and will find it difficult to accurately characterize and identify additional members. In the other extreme, if we begin with all but a few of the members, we are inherently limited in the number of additional members we can discover. Thus we expect there to be some internal maximum in absolute performance, as we see at $|S|/|C| = 0.1$ in Figure 3.6.

For relative recall, in contrast, we find that performance simply plateaus after a certain point, meaning that the additional information is neither helping nor hindering our relative rate of discovery.

3.4.3 Internal seed set structure

Finally, we consider how well the seed set is connected both internally and to the (unobserved) remainder of the community. Although we have seen that seed nodes with overall high degree point to reduced performance, we find here that performance is significantly higher when a large fraction of the seed nodes' edges are to nodes that lie in the community. Figure 3.7 (right panel) shows the performance in terms of this fraction; the left panel of the figure establishes a related point, that performance is better when the community C has a high ratio of internal edge density to external edge density.

These findings highlight several points. First, it suggests that in practice one can form an *a priori* estimate of the success of seed set expansion on a per-instance basis, based on structural properties of the seed set and/or the community, if one has estimates about these edge density parameters. Second, the strong relation to performance forms an interesting connection to mathematical work on community detection. The literature has emphasized that a good mathematical definition of a community is a set of nodes whose internal edge density is higher than its external one. It is interesting then that in these communities which are defined only by a shared qualitative property of member nodes that this canonical metric emerges as being correlated with high performance.

Finally, as with some of our earlier findings about seed set structure, we cannot necessarily control the seed set properties with which we make our prediction. But it does suggest a heuristic in which one can try to elicit from the domain expert a set of seed nodes that have good internal connectivity into the rest of the community, relative to their connectivity to the rest of the graph. The community results are a reminder that when searching for nodes in a community we should only expect high prediction accuracy if we can also expect that the members for whom we are searching form a densely connected subset of the graph.

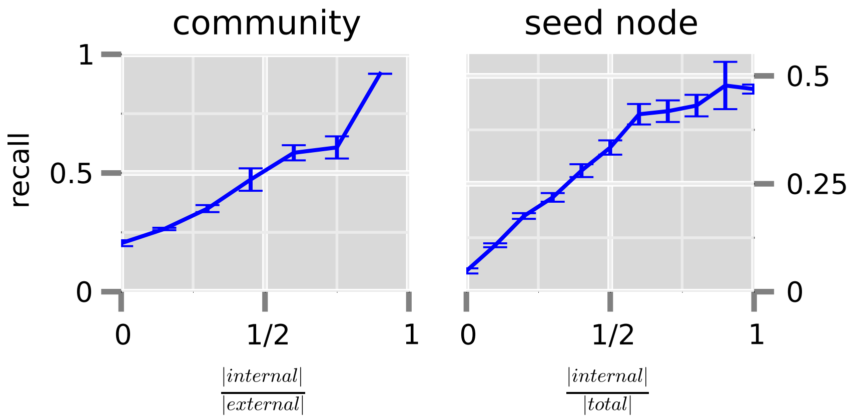


Figure 3.7: Left: Performance as a function of the target community's ratio of internal to external edge density. Right: Performance as a function of the fraction of edges the seed has within the community. Both plots use PageRank as the sorting metric and test on $C^{600}_{3/4}$.

3.5 Richer Interactions for Producing Labels

We have been thinking about seed set expansion in terms of interaction with a domain expert who is able to provide us with an initial set of examples. In this section we enrich this interaction by asking the expert to initially label some number of selected nodes beyond the seed set before we make further guesses about nodes likely to belong to the community. Thus, there are three types of nodes here, in order: (i) the initial seed set; (ii) the nodes explored in the *interactive* phase, when the expert is being actively queried; and (iii) the nodes explored in the *non-interactive* phase, when a set of nodes is guessed and only evaluated afterward for membership in C.

This second, additional round of interaction introduces several further questions. (1) How many nodes should be labeled in the interactive phase? We'll call this quantity the query budget. (2) Which nodes should be labeled in the interactive phase? We'll call this function the strategy. (3) Should the nodes in the interactive phase be labeled all at once? Or is there a performance gain to be had by introducing a feedback loop, in which at every iteration b nodes (strictly fewer than the query budget) are chosen according to the strategy, labeled, and used to refine the strategy input on the next round?

Finally, there are two ways of evaluating the algorithm, and we will consider both. One approach, in keeping with the initial motivation for seed set expansion, is to say that all nodes after the initial seed set count toward the performance of the algorithm, including those in the interactive phase. The other approach, more akin to a training/test split, is to say that the nodes in the interactive phase are purely for calibration, and the performance of the algorithm is only evaluated by its success in the non-interactive phase.

The former case is natural for settings where the goal is simply to collect members of the community — for example, in the case of a marketer who simply wants as large a set of likely purchasers as possible. In this case, there is an interesting trade-off between collecting nodes likely to belong to C in the interactive phase versus asking the expert about nodes that are less likely to belong to C, but which will help refine the boundary of C. This will be one of the main trade-offs we explore in this section.

Computational experiments We fix the size of the query budget to be that of the size of the seed set, that is, 10% the size of the community. We now focus on questions (2) and (3) above.

We consider four heuristics for choosing which nodes to query based on the values of one of our classifiers for membership in C: (1) nodes on the boundary of the decision function; (2) nodes most likely to be positive, as predicted by the classifier; (3) nodes most likely to be negative, as predicted by the classifier; (4) random nodes. We model this selection process by viewing (1)-(4) as having the associated probabilities $(p_0, p_+, p_-, p_*) = p$, and in each step of the interactive phase selecting, for example, $b_+ = p_+ b$ nodes predicted to be positive. We measure performance throughout this 3-dimensional parameter space ($p_0 + p_+ + p_- + p_* = 1$). We start with two observations that allow us to simplify our discussion and exploration of this parameter space:

1. It is never advantageous to have $p_* > 0$, that is, there is no benefit to querying

random nodes instead of putting that probability mass on some other dimension. This is not surprising and in fact this parameter was introduced as a baseline.

2. It is never advantageous to have $p_- > 0$, that is, there is no benefit to querying nodes that are most certainly not in the community. While this is not very surprising we did find it worth considering the possibility that having very clear examples of what community members do not look like could improve the classifier.

These observations imply that, for the scenarios we have defined, $p_- > 0$ and $p_* > 0$ are both dominated by strategies with $p_- = p_* = 0$. Thus we are left with a one-dimensional parameter space, $1 = p_+ + p_0$. In Figure 3.8 we explore how the performance is affected by labeling boundary nodes (via p_0), positive nodes (via p_+), or some combination thereof. We find that the optimal strategy depends on how performance is being measured:

1. If nodes found in both the interactive and non-interactive phases are counted towards performance, then $p_+ = 1$ is optimal.
2. If only nodes from the non-interactive phase count towards performance , then querying all nodes on the classifier boundary is optimal ($p_+ = 0$).

In the next section, we will see how the trade-off between these two results is reflected in the contrast between two different ways of formulating the problem of identifying nodes in a community.

Note, however, that when the interactive phase does not count towards performance, only a small performance gain is had by $p_0 = 1$ compared to $p_0 = 0$, though the gain is statistically significant (compare the performance of the lower curve on the left and right extremes in Figure 3.8). In contrast, when the goal is to maximize the number of community members collected (the upper curves in Figure 3.8), $p_+ = 1$ is a clearly dominant strategy.

Finally we address question (3), whether there is a performance gain to be had by introducing a feedback loop in the labeling stage. We find that it is advantageous to use

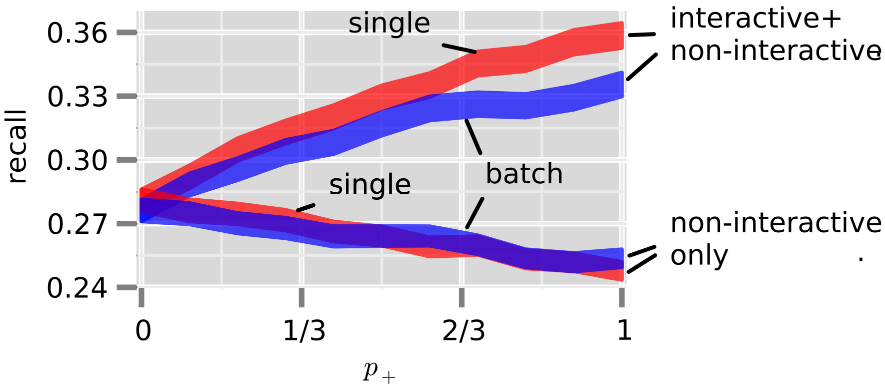


Figure 3.8: Performance of PageRank with two querying strategies and two evaluation metrics. In *batch* mode, we use the query budget in one sweeping request. *Single* mode is the other extreme: we query the expert for one node's label, reseed PageRank with the updated label, and repeat for as many steps as the query budget allows. Here we fix the query budget to be 10% the expected size of the community. The lower *non-interactive only* curves indicate the final performance of the classifier, not including any positive examples recovered during the interactive phase. The upper *interactive* + *non-interactive* curves include true positives recovered during both the interactive and non-interactive phases.

smaller block sizes when nodes found in the interactive phase count toward performance and the strategy $p_+ = 1$ is used (which is the optimal one for this case). In all other cases we find that the performances are statistically indistinguishable. The performance gain in the first case is all had in the interactive phase; that is, smaller block sizes do not improve the final classifier, but they do yield improved likelihood of finding positive nodes in the interactive stage. For the other scenarios there are either few positive nodes to be found by that strategy (i.e. querying on the boundary) or the nodes found in this stage do not count towards performance and so are irrelevant for evaluation purposes. This is also true in both cases that we discussed above: querying the most positive nodes ($p_+ = 1$) and querying on the boundary ($p_0 = 1$).

3.6 Seed Set Expansion: Theoretical Results

If we abstract beyond the specific methods used for seed set expansion, our discussion thus far has highlighted a number of themes implicit purely in the formulation of the problem itself — the difference between collecting a fixed number of nodes from a group C and finding the full set C; the trade-off, as in the previous section, between exploring in the vicinity of nodes known to be in C versus exploring near the estimated boundary of C; and the role of negative information, about non-membership in C.

We now consider a theoretical framework that seeks to highlight how these trade-offs and contrasts work across the different problem formulations. At the top level, it will be based on the distinction between the following two problems: *enumeration*, in which we want to find the full set C; and *seed set expansion*, in which we want to collect "many" members of C but not the full set.

Basic Set-up To set up these problems, let us assume we are given an undirected n-node graph $G = (V, E)$, and a subset $C \subseteq V$ is specified by a *membership oracle* that takes a node $v \in V$ and reports whether or not $v \in C$. We are also given a seed set $S \subseteq C$ of nodes that we know at the outset to belong to C. Finally, we will make the assumption that C is a connected set of nodes in G.[4]

In these terms, *enumeration* is now the problem of finding all the nodes of C using as few queries as possible to the membership oracle. *Seed set expansion*, in contrast is the problem in which, given a "budget" k, we want to find as many nodes of $C - S$ as possible using at most k queries to the membership oracle.

A Motivating Example To get an initial picture of the contrasts between these two problem formulations, let's consider them on an extremely simple graph, the n-node

[4] We view this as a reasonable approximation to the real problem in practice, since many of the groups C we are interested in studying will have a giant component $\tilde{C} \subseteq C$. Unless we have seed notes in the smaller components, it is not reasonable in any case to try discovering them; thus, we can view our problem as operating on this giant component. Indeed, some formulations of the seed set expansion problem have explicitly described it as searching for a specific component of the group.

cycle, which just consists of nodes $v_0, v_1, \ldots, v_{n-1}$ such that v_i is connected to v_{i-1} and v_{i+1} (addition modulo n). The group C we are trying to discover is a connected subset of the cycle (and hence a contiguous interval of nodes on it).

Suppose we are given a single seed node $v_j \in C$. Then the optimal algorithm for the seed set expansion problem it to begin querying nodes for membership in C starting at v_j and moving in a clockwise direction. The first time we come to a node $v_\ell \notin C$, we know we have fallen off one end of the interval defined by C. We then go back to v_j and do the same thing in the counter-clockwise direction. In this way, either we discover all of C (if our budget k is large enough), or else we collect nodes at almost full efficiency; aside from the node v_ℓ, every node we query is in C, and so we collect at least $k-1$ nodes with our budget of k.

An efficient algorithm for the enumeration problem has a quite different structure. First, for the enumeration problem it is natural to make an extra assumption that we didn't need for the seed set expansion problem — that we also know a node $z \notin C$. (Otherwise, we would have to begin with an essentially brute-force search for a non-member of C.) Given $s \in C$ and $z \notin C$, there are two paths of C that run between them: one clockwise from s to z, and the other one counter-clockwise from s to z. We perform binary search on the first of these paths to find a pair of adjacent nodes (v, w) such that $v \in C$ and $w \notin C$. We do the same on the other path, and thus find the two endpoints of the interval defining C in $O(\log n)$ queries to the membership oracle.

Contrasting algorithms Let us now contrast the approaches to these two problems. When $k \ll \log n$, seed set expansion collects nodes of C with almost no waste (i.e. almost no querying of non-members of C), while the efficient algorithm for enumeration could spend its first $\Omega(\log n)$ without ever identifying another member of C. On the other hand, when $k \gg \log n$, seed set expansion is collecting nodes of C one-by-one, whereas after the initial investment of $O(\log n)$ probes, the algorithm for enumeration implicitly knows all of C even though it hasn't visited all of its nodes explicitly.

These two contrasting strategies also relate to some other themes from earlier sections. As in the previous section, the seed set expansion algorithm does well by focusing attention near the nodes of C that it already knows about, whereas the enumeration algorithm focuses attention on farther-away nodes as it attempts to find the boundary of C. And negative information is not particularly relevant for the seed set expansion algorithm, whereas it is crucial for the efficiency of the enumeration algorithm — a reflection of the role that negative information played in our empirical results as well.

Thus far, however, these insights are all based on an extremely simple instance of the problem — finding a contiguous interval on the cycle. Do the same contrasts apply in other graphs? We now prove two theorems establishing that in fact they do, in a very strong sense: for *every* graph G, one can obtain contrasting and asymptotically optimal bounds for seed set expansion and enumeration that naturally extend the results we obtained for the simple case of the cycle.

We stress that in this theoretical analysis, we are focusing on comparing the consequences of the two problem formulations, seeking algorithms for each that are asymptotically optimal in the worst case, rather than trade-offs among specific heuristics for the problems.

General theorems We continue with the set-up defined at the outset, with an arbitrary graph $G = (V, E)$, a connected set of nodes $C \subseteq V$ that we want to discover, and a given seed set $S \subseteq C$. A key structure for analyzing our algorithms will be the set of edges $\delta(C)$, consisting of all edges that have one end in C and the other end not in C; a central quantity for parametrizing the performance of the algorithms will be $c = |\delta(C)|$.

We begin with the generalization for the seed set expansion problem, essentially showing that there is an algorithm that can collect at least $k - c$ nodes of C. This is the same sense in which the algorithm on the cycle was collecting nodes a perfect efficiency except for the queries in which it fell off the end of C.

Theorem 1. *Given a budget of k queries to the membership oracle, there is an algorithm that finds at least* $\min(k - c, |C - S|)$ *nodes in $C - S$. (In other words, it either finds at least $k - c$ nodes of $C - S$, or else it finds all of $C - S$.) This is asymptotically tight in the worst case.*

Proof. We show that the guarantee in the statement of the theorem will be achieved by any algorithm with the following structure: for k iterations, look for a node $v \in C$ connected by an edge to a node w whose membership in C is not yet known, and query w. If at any point before the k iterations are over, there are no such pairs (v, w), then the algorithm can stop and declare that it has found all of C.

Let us first verify why the algorithm is correct when it concludes it has found all of C. Since $G[C]$ is connected, as long as some nodes of C have not yet been found, there must exist an edge (v, w) such that $v, w \in C$, with node v already known to belong to C and node w not yet known to belong to C. Hence as long as all of C has not yet been found, the algorithm can execute another iteration.

Now, in each of the k iterations for which the algorithm does not discover a new node in $C - S$, it instead finds an edge $(v, w) \in E$ for which $v \in C$ and $w \notin C$. Thus $(v, w) \in \delta(C)$. Since there are only c edges in $\delta(C)$, there can be at most c iterations in which the algorithm does not find a new node in $C - S$; in the remaining iterations, at least $k - c$ in total, it must discover a new node in $C - S$, and this establishes the performance guarantee of the algorithm.

To see why the bound of $k - c$ is tight in the worst case, consider the star graph, equal to a tree with a central node v connected to $n - 1$ other nodes $w_1, w_2, \ldots, w_{n-1}$. If $S = \{v\}$ and C consists of v plus all but c of the leaf nodes, then in the worst case the algorithm will discover all c nodes not in C before moving on to any nodes in $C - S$.

We now give the contrasting generalization for the enumeration problem — that all of C can be found with $O(c \log n)$ queries. For this algorithm, as in the case of the cycle, we need to assume the presence of negative information; in particular, we assume there is a set of nodes $Z \subseteq V - C$ that is rich enough in its coverage that Z contains at least one node from each component of $G - C$.

Theorem 2. *Given seed set $S \subseteq V$ and negative set $Z \subseteq V$ satisfying the assumptions above, there is an algorithm to find all the nodes of C using at most $O(c \log n)$ queries to the membership oracle. This is asymptotically tight in the worst case.*

Proof. The algorithm works as follows. Let s be any node in the given set $S \subseteq C$; we say that an s-Z *path* is any path whose first node is s and whose last node belongs to Z. Edges will get *marked* during the execution of the algorithm; initially all edges start out unmarked. While there is an s-Z path P in G consisting entirely of unmarked edges, we perform binary search over the ordered sequence of nodes on P to find an edge (v,w) on P for which $v \in C$ and $w \notin C$. We then mark this edge (v,w). This is one iteration of the algorithm, and it uses $O(\log n)$ queries to the membership oracle in order to perform the binary search.

How many iterations can there be? Each iteration marks an edge in $\delta(C)$ that was previously unmarked; since $|\delta(C)| = c$, there can be at most c iterations. It follows that in total the algorithm performs at most $O(c \log n)$ queries.

Let F be the set of marked edges when the algorithm terminates, and let $U \subseteq V$ be the connected component of $G - F$ that contains the node s. We claim that $U = C$. Indeed, suppose there were a node $u \in C$ such that $u \notin U$. Then since $G[C]$ is connected, there is an s-u path consisting entirely of nodes in C, and hence using no marked edges. This contradicts the assumption that $u \notin U$. Conversely, suppose there were a node $u \in U$ such that $u \notin C$. Let z be a node of Z that belongs to same component of $G - C$ that u does. There is a u-z path Q such that all nodes belong to $V - C$; since each marked edge contains a node of C, there are no marked edges on Q. Concatenating Q with an s-u path using no marked edges, we get an s-z path using no marked edges, contradicting the termination of the algorithm.

These arguments show that $C \subseteq U$ and $U \subseteq C$, so $U = C$ and hence the algorithm produces the correct output set C.

Finally, we argue that there exist instances with a graph $G = (V,E)$ and a set $C \subseteq V$ for which $\Omega(c \log n)$ queries are required. One such graph is a collection of c parallel paths each of length n/c, that each run from s to z, but which otherwise have no nodes or edges in common. Any set C obtained by choosing a prefix of each s-z path, and taking the union of these c prefixes, will have $s \in C$ and $z \notin C$, with $\delta(C) = c$. There are $\Omega(n^c)$ such sets C, and hence any algorithm that uniquely identifies one of them through a sequence of yes/no questions to an oracle must make at least $\Omega(\log n^c) = \Omega(c \log n)$ queries in the worst case.

3.7 Related Work

The seed set expansion problem has its roots in a number of overlapping areas, including the problem of identifying central nodes in social networks [53, 63] and finding related and/or important Web pages from an initial set of query results [65, 91].

In particular, the PageRank algorithm broadened from its initial focus on Web search [91] to also include methods for finding nodes "similar" to an initial root, by starting short random walks from the root and seeing which other nodes were likely to be reached [56]. Spielman & Teng developed methods that started with a seed node and sorted all other nodes by their degree-normalized PageRank with respect to this seed [106]; they also introduced ideas based on truncation of small values, leading to a method known as PageRank-Nibble. Anderson & Lang and Andersen *et al.* built on these methods to formulate an algorithm for detecting overlapping communities in networks [7, 8]; in our evaluation, their method serves as our version of DN-PageRank, short for degree-normalized, personalized PageRank. DN-PageRank was adopted by Leskovec *et al.*[77] and Yang & Leskovec for global and local community detection. In a large comparison study they established DN-PageRank as competitive with METIS [61], a sophisticated and popular graph partitioning algorithm. Finally, Abrahao *et al.* observe that from among approximately ten popular community detection algorithms, ground-truth communities are structurally most similar to the communities discovered by random walk methods [1].

In parallel with the development of PageRank-based methods, another line of work explored methods for seed set expansion by adding nodes to a growing community (or removing them) if a target measure such as conductance or modularity is improved by doing so. Clauset [23] used this idea by adding single nodes to increase modularity; Luo *et al.*[79] allowed for addition and deletion of larger sets; and Mislove *et al.*[82] used greedy node addition to reduce conductance.

Finally, a number of approaches evaluated nodes based on the number of neighbors they had in and out of the community, adding nodes to the community when they optimized a function of these two quantities. Bagrow [11] did this for a measure called *outwardness*, defined as the degree-normalized difference between neighbors inside and outside the community. Mehler & Skiena [81] used several variations of neighbor counting methods for seeded community detection, the main ones being pure neighbor count, neighbor ratio, and binomial probability of neighbor distribution. More recently in 2013 Weber *et al.* used another variation of a neighbor-counting metric to infer the political ideology of Twitter users, based on which community a user retweeted more frequently.

In an analysis of the effect of seed-set structure, Whang, Gleich, & Dillon [121] systematically compared several sophisticated approaches for choosing the seed sets with which to seed PageRank-based measures for community detection. Their methods outperform existing sophisticated methods, but do not significantly outperform the use of random nodes.

Finally, essentially all seed set expansion algorithms need to make a decision about the choice of *stopping criterion* — when does one stop expanding the set? Such a criterion can be treated relatively independently from the choice of expansion rule. Andersen & Lang [8] and Yang & Leskovec [77] choose the first nodes that represent a set with a locally minimal conductance (given that additions happen in the order induced by sorted, DN-PageRank). Mehler & Skiena continue until the mining rate of reserved labeled nodes passes below a certain threshold; Bagrow [11] looks for transitions and cusps in the modularity that one expects at a community border. Others such as Mislove *et al.* [82], Luo *et al.* [79], and Clauset[23] greedily add and subtract nodes from the predicted community until a local maximum is reached. In 2012 Yang & Leskovec used PageRank-type measures to empirically compare different topological parameters to identify community boundaries in real-world data sets. They found that the result was somewhat domain dependent, but that either the set's conductance or its triad participation ratio were, most reliably, high accuracy stopping rules.

3.8 Conclusion

The seed set expansion problem has been gaining visibility as a general-purpose framework for identifying members of a networked community from a small set of initial examples. But subtle trade-offs in the formulation and underlying methods can have a significant impact on the way this process works, and in this paper we have identified several such principles about the relative power of different expansion heuristics, and the structural properties of the initial seed set. Our investigations have involved analyses of datasets across diverse domains as well as theoretical trade-offs between different problem formulations.

There are a number of interesting directions for further work. In particular, the power of PageRank-based methods raises the question of whether these are indeed the "right" algorithms for seed set expansion, or whether they should be viewed as proxies for a richer set of probabilistic approaches that could yield strong performance. Second, the contrast between seed sets consisting of random nodes versus those consisting of high-degree nodes suggests that deeper structural contrasts may be present as well; a richer understanding of the seed sets that lead to the most effective expansions to a larger community could provide useful insights for the application of these methods. And finally, as noted earlier in the paper, nodes in a network tend to belong to multiple communities simultaneously, and a robust way of expanding several overlapping communities together is a natural question for further study.

3.9 Summary of algorithms

Here we provide brief summaries of the algorithms used in the main text; for more details see citation. We distinguish between three types of algorithms:

Neighbor counting (a) Outwardness, the degree-normalized difference between the number of edges a node has within and without of the labeled community [11];
(b) Neighbors, the number of neighbors one has in the labeled community [81];
(c) DN-Neighbors, the degree-normalized version of Neighbors [81];
(d) BinomProb, the binomial probability that a node is in the community, given the number of neighbors it has in the labeled community [81].

Greedy structural optimization (e) Modularity, greedy algorithm: in each step add the node that yields the highest increase in the set modularity of the predicted community, [23];

(f) SetModularity, greedy algorithm: in each step add the nodes that yield a positive increase in set modularity, then remove the set of all nodes whose removal precipitates an increase, [79];

(g) Conductance, greedy algorithm: in each step add the node that yields the most negative change in conductance, [82].

PageRank (h) PageRank, implemented here with personalization and computed using the power method and jumpback probability $\alpha = 0.10$, see [56] or [5] for implementation details. For comparison with [7, 128] we also implemented a version with ε truncation (semi-accurate estimate of PageRank), however we found that below a $\varepsilon \approx 1/|G|$) there were no significant differences in performance, and past this ε, the performance steeply plummets to approach that achieved by random guessing.

(i) DN-PageRank, the degree-normalized version of PageRank, [106], also see footnote [5].

[5] Let ρ^t be the tth random walk vector given that the initialization set S is the set of known community members.

Let $\chi(S)$ be an indicator vector where $\chi_i(S) = 1$ if $i \in S$ and 0 otherwise. Let A be the adjacency matrix, where $A_{ij} = 1$ if j links to i and 0 otherwise. The degree of node j is given by $d_j = \sum_i A_{ij}$. The random walk is initialized with $\rho^0 = \chi(S)/|S|$. In step $t+1$ each node i distributes $\alpha \rho_i^t$ probability mass uniformly over the seed set S and $(1-\alpha)\rho_i^t$ probability mass over its neighbors. The corresponding probability transition matrix $M(S)$ is:

$$M_{ij}(S) = \alpha \frac{\chi_i(S)}{|S|} + (1-\alpha)\frac{A_{ij}}{d_j}.$$

CHAPTER 4
BLOCK MODELS AND PERSONALIZED PAGERANK

This chapter is written in collaboration with Johan Ugander and Jon Kleinberg and is currently in preparation for publication.

Methods for ranking the importance of nodes in a network have a rich history in machine learning and broadly across domains that analyze structured data. A recent line of work has formalized the evaluation of these ranking methods though the *seed set expansion problem* in networks: given a seed set S of nodes from a community of interest in an underlying graph, can we expand it to find the rest of the community? In this work we start from the recent observation that the most powerful techniques for this problem, personalized PageRank and heat kernel methods, operate in the space of *landing probabilities* of a random walk rooted at the seed set, ranking nodes according to different weighted sums of the landing probabilities. However, both weight schemes have previously lacked an a priori relationship to the actual seed set objective. In this work we develop a principled framework for evaluating ranking methods by studying seed set expansion applied to the stochastic block model. We derive the optimal gradient for separating the landing probabilities of the two classes in a stochastic block model, and find, surprisingly, that it is asymptotically equivalent to personalized PageRank for a specific choice of the personalized PageRank parameter α that depends on the parameters of the stochastic block model. This connection provides a novel formal motivation for the success of personalized PageRank in seed set expansion and node ranking generally. We use this connection to propose more advanced techniques that incorporate higher moments of landing probabilities; we show that this strengthening yields much better results for stochastic block models, and for real-world data it is competitive with and in some cases outperforms the strongest available heuristics for the problem.

The challenge of contextually ranking nodes in a network has emerged as a problem of canonical significance in many domains, with a particularly rich history of study in social and information networks. An active line of work has recently focused on the problem of *seed set expansion* in networks [8, 11, 67, 81, 100, 121, 128], providing a particularly clean version of this general challenge.

The premise underlying the seed set expansion problem is a natural one: We are given a graph G representing some form of social or information network, and there is a hidden community of interest that we would like to find, corresponding to an internally well-connected set of nodes. We know a small subset S of the nodes in this community, and from this "seed set" S we would like to expand outward to find the rest of the community. This problem arises in a wide range of domains, including settings where we are trying to rank web pages in relation to other web pages, to identify a social group from a set of example members provided by a domain expert, or to help a user automatically populate a group they are defining in an online social-networking application.

A recent focus in the work on this problem has been the power of approaches based on random-walk methods, including versions of *personalized PageRank* [48, 56, 67] and physical analogues based on the heat equation for graphs [21, 66]. These techniques can be viewed as operating on the following quantities: for each node v in the graph, and each number of steps k, we let r_k^v denote the probability that a random walk on the graph ends up at v after exactly k-steps, starting from a particular seed node in S (or a node chosen uniformly at random from S). Methods based on PageRank and heat kernels then combine these values $\{r_k^v\}$ using particular functional forms as *discriminant functions*: they produce a "score" for each node v, with the structure $score(v) = \sum_{k=1}^{\infty} w_k r_k^v$ for coefficients $\{w_k\}$, and the seed set is expanded by considering nodes in decreasing order of their scores [66, 67]. Geometrically, these rankings amount to sweeps through the space of landing probabilities with hyperplanes normal to some vector, where personalized PageRank and heat kernel amount to two different choices of normal vectors.

These methods are elegant in their formulation and also appear to be both quite powerful as well as scalable [59, 66, 67]. At the same time, their success leaves open a number of very basic questions. In particular, if we think of the *landing probabilities* $\{r_k^v\}$ over nodes v and steps k as providing us with a rich set of features relevant to membership in the community of interest, then it becomes clear that personalized PageRank and heat kernel formulations are simply specific, and apparently arbitrary, ways of combining these features using hand-constructed coefficients $\{w_k\}$.

Motivations for the specific form of these measures have come from several domains. For personalized PageRank, the *random surfer model* [91] proposes a justification in the context of Web page ranking by arguing that it agrees with the landing probabilities of a user who navigates the Web by randomly clicking on links and returning to their starting point with some probability. Closer to the current context, both personalized PageRank and heat kernel methods have been the subject of *Cheeger-type results* [7, 22] relating their output to the structure of sparse cuts in the underlying graph; and recent work has related personalized PageRank to solutions of problems with min-cut objectives [44]. Even here, however, there has not been an argument that any of these measures are optimally combining the random walk landing probabilities under a specific objective, nor a direct connection between any of the these measures and the problem that seed expansion seeks to solve.

Is there a principled reason why the expressions for PageRank or the heat kernel represent the “right” way to combine the information coming from random walks, or could there be better approaches? And is there a formal framework available for deriving or at least motivating effective ways of combining random walk probabilities? Given the diverse and important applications where PageRank and heat kernel methods have seen successes, we consider a broader examination of the space of methods for combining available random walk information, appreciating that the approaches in existing work are simply particular points in that space.

The pivotal observation we pursue in this work is that a basic model of separable structure in graphs known as the stochastic block model [50] can be used to derive principled methods for ranking nodes in the space of landing probabilities. We focus our attention on a two-block stochastic block model, where one block of nodes corresponds to the community of a labelled seed set, while the other block of nodes corresponds to its complement, the remainder of the graph. In this setting the problem of finding the hidden community of interest has a correct answer with respect to an underlying graph generation model, and hence methods for combining landing probabilities of random walks can be evaluated according to their ability to find this answer.

For this two-block stochastic block model we make the surprising observation that the optimal weights for sweeping between the centroids of the landing probabilities for the two blocks is asymptotically concentrated (for large graphs) on the weights of personalized PageRank, for a specific choice of the PageRank parameter corresponding to parameters of the stochastic block model. This connection between personalized PageRank and stochastic block models is a novel bridge between two otherwise disconnected literatures, and gives a strong motivation for using personalized PageRank in contextual ranking and seed set expansion problems.

Beyond merely geometric rules, we observe block models can be used to propose more advanced scoring methods in the space of landing probabilities, and our analysis points to important ways in which personalized PageRank can be strengthened. Although its geometric gradient is optimally oriented with regards to the landing probability centroids, personalized PageRank does not account for the variance or covariance of these landing probabilities, e.g. how the 2-hop landing probabilities from a given seed correlate with the 3-hop landing probabilities. We derive weights that correctly incorporate these variances and covariances and we show that relative to the stochastic block model benchmark, this new family of measures significantly outperforms personalized PageRank and the other techniques (including heat kernel methods).

Finally, we take our new methods to real-world data with known community structure. We find that the performance of our more advanced methods is closely comparable to PageRank and heat kernel methods for identifying small fractions of the seed community, and we outperform these heuristics for finding large fractions of the community. This success suggests that methods based on principled foundations can match and, for some important parts of the problem space, outperform the leading heuristics for seed set expansion.

4.1 Discriminant Functions for Stochastic Block Models

In this section we present principled derivations for how to score nodes of a graph in the space of random walk landing probabilities when the underlying graph comes from a stochastic block model. We term the scoring functions we derive *discriminant functions*, a phase coined by Fisher to describe functions for dichotomous classification [40].

The *stochastic block model* [50], also known as the planted partition model [24, 36], is a distribution over graphs that generalizes the Erdős-Rényi random graph model $G(n,p)$ [37] to include a planted block structure. It can be described in terms of the following process for constructing a random graph $G=(V,E)$. There is a partition of the nodes into k disjoint sets (blocks) $V_1,...,V_k$, where $|V_i|=n_i$, together with a a $k \times k$ matrix P whose entries are in $[0,1]$. The entry P_{ij} of matrix P describes the independent probability of a node in V_i being connected to a node in V_j: $\Pr((u,v)\in E|u\in V_i, v\in V_j)=p_{ij}$.

A stochastic block model is thus completely described by the parameters $N=(n_1,...,n_k)$ and P, and we let $G(N,P)$ denote the distribution over graphs with given parameters. We allow self-loops and derive results for both directed and undirected graphs, where the latter case implies that P is a symmetric matrix. The Erdős-Rényi random graph model $G(n,p)$, with n and p scalars, is an undirected one-block special case.

Let $G((n_a,n_b),P)$ denote a two-block block model, where block V_a denotes the community of the seed set (the *seed community*) and block V_b denotes the remainder of the graph. Our key insight regarding this model is that even though it is very simple, the nodes in block V_a will tractably differ from nodes in block V_b in terms of their random walk landing probabilities. As a result, basic discriminant functions in the space of landing probabilities can be used to classify whether or not nodes belong to the seed community.

A *discriminant function* is a function that assigns a score to each point in a feature space, and these scores can be used for classifying or ranking points as belonging to one of two underlying binary classes. We will focus on two particular classes of discriminant functions: geometric discriminant functions and Fisher discriminant functions. *Geometric discriminant functions* perform an intuitive linear sweep through the feature space from one centroid a to the other centroid b, $f(r)=w^T r$ for $w=(a-b)$, scoring points based on their inner product with the vector connecting the two centroids. Points closer to the centroid a then have a higher score. *Fisher discriminant functions* employ a descriptive model where the two classes of points are described by their first two moments using multivariate Gaussians $N(a, \Sigma_a)$ and $N(b, \Sigma_b)$ in the feature space, scoring points based on their relative probabilities of belonging to the two Gaussians. The special case of $\Sigma_a = \Sigma_b = I$ is equivalent to a geometric discriminant function, but in general the Fisherian approach is capable of accounting for heterogeneous variance across features and covariance between features to make a more principled discrimination.

Geometric discriminant functions

Recall that r_k^v, the k-step landing probability of a node v given a seed node in an underlying graph, is the probability that a random walk beginning at that seed node will be at v after exactly k steps. We are interested in mapping each node v to a vector of its first K landing probabilities $(r_1^v, r_2^v, \ldots, r_K^v)$, and asking what these vectors look like in graphs generated by a stochastic block model.

Let $(a_1, ..., a_K)$ denote the centroid of the landing probabilities for nodes in block V_a, and let $(b_1, ..., b_K)$ denote the centroid of the landing probabilities for nodes in block V_b. The geometric discriminant function $f(r) = (a-b)^T r$ then amounts to a sweep of the space of landing probabilities, and will let us rank each node $v \in V$ in terms of its propensity for belonging to the seed community based on purely geometric arguments regarding the node's landing probabilities $(r_1^v, \ldots, r_K^v) \in [0,1]^K$. In this notation personalized PageRank assigns scores according to the infinite sum $\sum_{k=1}^{\infty} \alpha^k r_k^v$, for a parameter $\alpha \in (-1,1)$, and the heat kernel method assigns scores by $\sum_{k=1}^{\infty} \frac{e^{-t} t^k}{k!} r_k^v$ for a parameter $t > 0$. Truncating these methods to a finite walk length K, both methods then amount to linear discriminant functions for particular weight vectors $w_{PPR}(\alpha) = (\alpha, \alpha^2, ..., \alpha^K)$ and $w_{HK}(t) = (e^{-1}t, \frac{e^{-2}t^2}{2}, ..., \frac{e^{-t}t^K}{K!})$. Note that the PageRank parameter α is often interpreted as the teleportation probability of a teleporting random walk, which assumes that α is non-negative, but under the above interpretation the Personalized PageRank score function is well-defined for $-1 < \alpha < 0$ as well.

We now establish the following asymptotic equivalence between personalized

PageRank and geometric classification of stochastic block models in the space of random walk landing probabilities, the main theoretical result of our work.

Proposition 1. *Let G_n be any n-node graph generated from a stochastic block model $G((n_a,n_b),P)$ with $n_a = \lambda n$ and $n_b = (1-\lambda)n$, $\lambda \in (0,1)$ fixed and $p_{ij} > 0$, $\forall i,j$. Let $\hat{w}$ be the geometric discriminant weight vector in the space of landing probabilities (1-step through K-step, K fixed) between the two block classes of G_n.*

For any $\varepsilon > 0$, $\delta > 0$, there exists an n sufficiently large such that

$$||n_a\hat{w} - n_a\Psi||_1 \leq \varepsilon$$

with probability at least $1-\delta$, where Ψ is a vector with coordinates specified by the solution to a two-dimensional linear homogeneous recurrence relation.

Corollary 1. *Let G_n be generated from a stochastic block model that is either undirected and with equal expected degree for all nodes or directed with equal expected in-degrees and out-degrees for all nodes. Then Proposition 1 holds for $n_a\Psi = w_{PPR}(\alpha_*)$, the Personalized PageRank weight vector with $\alpha_* = \frac{n_b p_{bb} - n_b p_{out}}{n_b p_{bb} + n_a p_{out}}$.*

The bulk of the heavy lifting for the above proposition and corollary is delivery by the following lemma, a proof of which is given in the supplementary material.

Lemma 1. *For any $\varepsilon > 0$, $\delta > 0$, there is an n sufficiently large such that the random landing probabilities $(\hat{a}_1,....,\hat{a}_K)$ and $(\hat{b}_1,...,\hat{b}_K)$ for the two blocks of a stochastic block model on n nodes with $n_a = \lambda n$ and $n_b = (1-\lambda)n$, $\lambda \in (0,1)$ fixed and matrix P fixed with $p_{ij} > 0$, $\forall i,j$ satisfy the following conditions with probability at least $1-\delta$ for all $k > 0$:*

$$n_a\hat{a}_k \in \left[(1-\varepsilon)\frac{f_k}{f_k+g_k}, (1+\varepsilon)\frac{f_k}{f_k+g_k}\right] \text{ and} \tag{4.1}$$

$$n_b\hat{b}_k \in \left[(1-\varepsilon)\frac{g_k}{f_k+g_k}, (1+\varepsilon)\frac{g_k}{f_k+g_k}\right], \tag{4.2}$$

where f_k and g_k are given by

$$f_k = \frac{-\lambda_1^k \mathbf{u}_{f1}\mathbf{u}_{g2} + \lambda_2^k \mathbf{u}_{f2}\mathbf{u}_{g1}}{-\mathbf{u}_{f1}\mathbf{u}_{g2} + \mathbf{u}_{f2}\mathbf{u}_{g1}},\ g_k = \frac{(-\lambda_1^k + \lambda_2^k)\mathbf{u}_{f2}\mathbf{u}_{g2}}{-\mathbf{u}_{f1}\mathbf{u}_{g2} + \mathbf{u}_{f2}\mathbf{u}_{g1}}, \tag{4.3}$$

with

$$\mathbf{u}_f = \begin{pmatrix} \frac{(d_{aa} - d_{bb}) - \sqrt{(d_{aa} - d_{bb})^2 + 4d_{ab}d_{ba}}}{2d_{ba}} \\ 1 \end{pmatrix}, \tag{4.4}$$

$$\mathbf{u}_g = \begin{pmatrix} \frac{(d_{aa} - d_{bb}) + \sqrt{(d_{aa} - d_{bb})^2 + 4d_{ab}d_{ba}}}{2d_{ba}} \\ 1 \end{pmatrix}, \tag{4.5}$$

$$\lambda_1 = \frac{1}{2}\left((d_{aa} + d_{bb}) - \sqrt{(d_{aa} - d_{bb})^2 + 4d_{ab}d_{ba}}\right), \tag{4.6}$$

$$\lambda_2 = \frac{1}{2}\left((d_{aa} + d_{bb}) + \sqrt{(d_{aa} - d_{bb})^2 + 4d_{ab}d_{ba}}\right), \tag{4.7}$$

and $d_{ij} = n_i p_{ij}$.

With this lemma in hand, we now prove Proposition 1.

Proof (of Proposition 1). First we will use the lemma to show that the coordinates of the weight vector $\hat{w} = \hat{a} - \hat{b}$ are concentrated as specified. From Lemma 1 we have that for any $\varepsilon_1 > 0, \delta > 0$ there exists an n sufficiently large such that

$$(1 - \varepsilon_1)\frac{f_k}{f_k + g_k} < n_a \hat{a}_k < (1 + \varepsilon_1)\frac{f_k}{f_k + g_k} \tag{4.8}$$

$$(1 - \varepsilon_1)\frac{g_k}{f_k + g_k} < n_b \hat{b}_k < (1 + \varepsilon_1)\frac{g_k}{f_k + g_k}. \tag{4.9}$$

with probability at least $1 - \delta$ and f_k and g_k are specified by (4.27)–(4.31). As a result, whenever this containment holds we have that the coordinates of the geometric weight vector $\hat{w} = \hat{a} - \hat{b}$ obeys

$$|\hat{w}_k - \Psi_k\ | \leq \varepsilon_1 \Phi_k, \tag{4.10}$$

where

$$\Psi_k = \frac{1}{n_a}\frac{f_k}{f_k + g_k} - \frac{1}{n_b}\frac{g_k}{f_k + g_k}, \tag{4.11}$$

and

$$\Phi_k = \frac{1}{n_a}\left(\frac{f_k + \frac{n_a}{n_b} g_k}{f_k + g_k}\right) \leq \frac{1}{n_a}\frac{\lambda}{1-\lambda}. \tag{4.12}$$

The inequality comes from the fact that $n_a = \lambda n$ and $n_b = (1-\lambda)n$, and the bound is independent of k. Notice also that Φ_k is clearly positive. Setting $C = \frac{\lambda}{1-\lambda}$, we see that (4.10) becomes

$$|n_a \hat{w}_k - n_a \Psi_k| < C\varepsilon_1. \tag{4.13}$$

This last expression furnishes us with a coordinate-wise bound for each of the K coordinates, and under the containment event of Lemma 1 we have that they all hold jointly with probability $1-\delta$. Choosing $\varepsilon_1 < \varepsilon/(CK)$ achieve the requisite bound on the 1-norm of the vector, proving the proposition. □

In the special case of all nodes in the graph having equal expected degrees, the following corollary establishes that the optimal geometric classifier is asymptotically equivalent to personalized PageRank for a particular choice of α.

Proof (of Corollary 1). To consider the special case of all nodes have equal expected degree, we note that for undirected graphs:

$$n_a p_{aa} + n_b p_{ab} = n_a p_{ba} + n_b p_{bb} \tag{4.14}$$

and $p_{ab} = p_{ba}$, while for directed graphs:

$$n_a p_{aa} + n_b p_{ab} = n_b p_{bb} + n_a p_{ba} \tag{4.15}$$

$$n_a p_{aa} + n_b p_{ba} = n_b p_{bb} + n_a p_{ab} \tag{4.16}$$

and thus again $p_{ab} = p_{ba}$. For both cases we can then rewrite $p_{ab} = p_{ba} = p_{out}$. Under these conditions the eigenvectors and eigenvalues, stated generally in (4.28)–(4.31), simplify to the following:

$$\mathbf{u}_f = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \ \mathbf{u}_g = \begin{pmatrix} \frac{n_a}{n_b} \\ 1 \end{pmatrix}, \ \begin{cases} \lambda_1 = n_b(p_{bb} - p_{out}), \\ \lambda_2 = n_b p_{bb} + n_a p_{out}. \end{cases} \tag{4.17}$$

The solution to the recurrence relations in (4.27) then simplifies to:

$$f_k = \frac{n_b \lambda_1^k + n_a \lambda_2^k}{n_a + n_b}, \ g_k = \frac{-n_b \lambda_1^k + n_b \lambda_2^k}{n_a + n_b}. \tag{4.18}$$

Notice that $f_k + g_k = \lambda_2^k$. Finally, we have that $n_a \Psi_k$ simplifies to

$$n_a \Psi_k = \left(\frac{1}{f_k + g_k} \right) \left(f_k - \frac{n_a}{n_b} g_k \right) = \frac{\lambda_1^k}{\lambda_2^k} = \left(\frac{n_b p_{bb} - n_b p_{out}}{n_b p_{bb} + n_a p_{out}} \right)^k. \tag{4.19}$$

Personalized PageRank with $\alpha_* = \frac{n_b p_{bb} - n_b p_{out}}{n_b p_{bb} + n_a p_{out}}$ employs precisely the weights $(\alpha_*)^k$. □

A few remarks are in order. First, the scalar factor n_a that differs between $\hat{w}$ and $w_{PPR(\alpha_*)}$ does not change the relative ranking of the nodes, since ranking according to the discriminant function $f_1(r) = w^T r$ and $f_2(r) = n_a w^T r$ is equivalent. Second, the criteria that the stochastic block model be dense ($p_{ij} > 0$ and fixed) is a necessary part of the proof, and it is unclear if a similar result holds for a sparse model.

In the special case of balanced blocks where $n_a = n_b = n$, $p_{aa} = p_{bb} = p_{in}$, and $p_{ab} = p_{out}$, sometimes known as the *affiliation model* [42], we succinctly obtain $\alpha_* = \frac{p_{in} - p_{out}}{p_{in} + p_{out}}$. This simple expression provides a useful interpretation of the choice of α in personalized PageRank: α close to 0 is ideal for identifying the seed community where the planted partition surrounding the seed node is very good, meaning $p_{in} \gg p_{out}$. Meanwhile, α is close to 1 is best for when the planted partition is very weak.

In Figure 1 we see that the theoretical coefficients show near-perfect agreement with the empirical coefficient means for an example block model on $n = 500$ nodes. Our centroid derivations have not assumed that the block model be assortative, and the derivations are equally accurate for disassortative block models where nodes in the two blocks are more strongly connected across blocks than within blocks. From Figure 1 we also see that the empirical variance of the coefficients can be highly non-uniform, with the 1-step landing probabilities showing much greater variation than the landing probabilities after subsequent steps. This observation motivates our next approach, where we explicitly consider these heterogeneous variances, as well as covariances between the landing probabilities of different step lengths.

Fisher discriminant functions

The above geometric approach can be viewed as a special case of a more general probabilistic approach to deriving discriminant functions proposed by Fisher, and we will now derive such functions that consider both the centroids and covariances of the sets of landing probabilities.

We derive Fisher discriminant functions in which the landing probabilities for the two communities are described by multivariate Gaussians $N(a, \Sigma_a)$ and $N(b, \Sigma_b)$ for the seed community and remainder community, respectively. Here a and b are the same centroids as we derived earlier. Note that we are not assuming these point sets are actually multivariate Gaussian, but simply using the Gaussians to capture the first two moments (mean and covariance) of the point sets.

What follows are standard derivations for Fisher discriminant functions. Let $z^u \in \{0, 1\}$ be the assignment of each node $u \in V$ to one of of the two blocks, with $z^u = 1$ denoting the seed node block. The probabilities of a given $r = (r_1, ..., r_K)$ belonging to

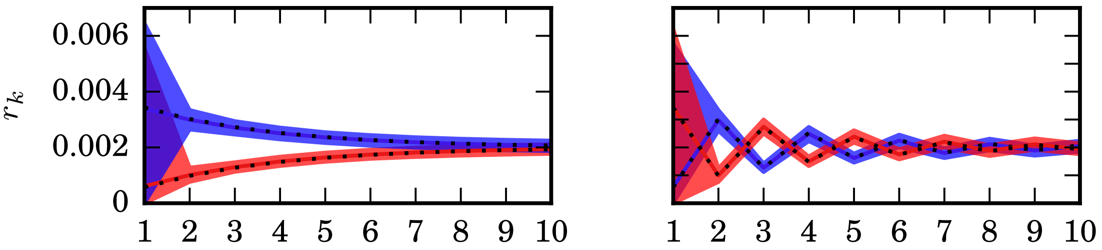


Figure 4.1: Left: theoretical (dotted lines) and empirical centroids $(a_1, ..., a_K)$ (blue) and $(b_1, ..., b_K)$ (red) for a stochastic block model with $n_a = n_b = 250$, $p_{aa} = p_{bb} = 0.6$, $p_{ab} = 0.1$, shown with empirical [5%, 95%] quantiles. Right: the same illustration for a disassortative block model where $p_{aa} = p_{bb} = 0.1$, $p_{ab} = 0.6$.

each block are then:

$$p(r|z=1) \propto |\Sigma_a|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(r-a)^T \Sigma_a^{-1}(r-a)\right), \tag{4.20}$$

$$p(r|z=0) \propto |\Sigma_b|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(r-b)^T \Sigma_b^{-1}(r-b)\right). \tag{4.21}$$

Let us introduce the parameter $\pi = p(z=1)$ to denote the probability that a given node is in the seed community. When the parameters of the stochastic block model are known it is clear that $\pi = n_a/(n_a+n_b)$. The log of the likelihood ratio then becomes:

$$\begin{aligned} g(r) = \ln \frac{p(r|z=1)p(z=1)}{p(r|z=0)p(z=0)} &= \underbrace{(\Sigma_a^{-1}a - \Sigma_b^{-1}b)^T}_{w} r + r^T \underbrace{(\frac{1}{2}\Sigma_b^{-1} - \frac{1}{2}\Sigma_a^{-1})}_{W} r \\ \underbrace{+ -\frac{1}{2}\left(a^T\Sigma_a^{-1}a - b^T\Sigma_b^{-1}b + \ln\frac{|\Sigma_a|}{|\Sigma_b|}\right) + \ln\frac{\pi}{1-\pi}}_{w_0} &= w^T r + r^T W r + w_0, \end{aligned} \tag{4.22}$$

where we've identified the vector w, matrix W, and scalar w_0 to simplify notation. In ranking contexts we we can safely ignore w_0, which is constant for all nodes. Equation (4.22) thus provides a quadratic discriminant function for ranking seed community membership in a manner that accounts for the covariance structure of the different landing probabilities, e.g. how the landing probability at a node u after k steps covaries with the landing probability after $k+1$ steps.

Special case discriminant functions

If we assume $\Sigma_a = \Sigma_b = \sigma^2 I$, we recover the earlier geometric discriminant function

$$g_1(r) = \sigma^{-2}(a-b)^T r + C, \tag{4.23}$$

up to an arbitrary additive constant C, and observe that the earlier geometric approach corresponds to a uniform and independent variance assumption on the two point clouds in the space of landing probabilities. In a slightly more general setting assuming $\Sigma_a = \Sigma_b = \Sigma$, meaning that the two covariance matrices are identical but otherwise arbitrary, Equation (4.22) reduces to

$$g_2(r) = [\Sigma^{-1}(a-b)]^T r + C, \tag{4.24}$$

again up to an arbitrary additive constant C. This discriminant function is still linear, but can have a very different form than $g_1(r)$.

While we have shown that personalized PageRank takes a principled approach to ranking seed community membership, accounting for the covariance structure of the landing probabilities suggests that much better linear discriminant functions with the form of Equation (4.24) rather then Equation (4.23) — let alone quadratic functions with the form of Equation (4.22) — exist for graphs where the structure can reasonably be motivated as coming from a two-block stochastic block model.

Learning model parameters

The above theoretical derivations assumed known parameters, but in practice the parameters of the stochastic block model that inform the choice of α as well as the covariance matrices must be learned. Recent results have developed consistent estimators for the parameters of stochastic block models for an observed graph, with two separate estimation regimes relevant to our work. In the first regime, a string of results have culminated in a tidy closed-form consistent estimator for the parameters of a two-block stochastic block model with known block size [4, 5, 42]. In a second regime where the block sizes are unknown, results have been derived that show the consistency of the maxima of a tractable composite likelihood function [6]. We will focus on the former regime of known block sizes, and will further focus on the special case of $p_{aa} = p_{bb} = p_{in}$, $p_{ab} = p_{out}$ known as the affiliation model, for which consistent estimators $\hat{p}_{in}$ and $\hat{p}_{out}$ [5] are reproduced in the supplementary material.

Given the parameter estimates $\hat{p}_{in}, \hat{p}_{out}$ and $n_a = n_b$ known, we can also estimate the covariance matrices of the random walk landing probabilities, Σ_a and Σ_b, from simulations of an adequate number of stochastic block models with the learned parameters.

4.2 Results for Stochastic Block Models and Real-world Networks

We now evaluate the results of our approach to contextual ranking by testing our discriminant functions against the widely used personalized PageRank and heat kernel heuristics. Our evaluation is two-fold: we first evaluate the performance at seed set expansion on graphs that have been generated by stochastic block models, the domain where our principled approach has sound theoretical foundations, and then evaluate our performance on real-world networks.

To start with the first mode of evaluation, in Figure 4.2A we show the cumulative recall for a stochastic block model with $n_a = n_b = 64$ nodes and $p_{in} = 0.6$, $p_{out} = 0.4$. The curves measure the recall of the classification methods when attempting to return a seed block of m nodes, as a function of m. We see that our quadratic discriminant function has considerably improved recall, identifying the first 64 nodes with nearly perfect precision. In contrast we see that personalized PageRank (with $\alpha = 0.85$, a standard choice) and heat kernel (with $t = 2$) perform the task with comparable recalls. The "linear" classification is a classification according to $(a-b)^T r$, akin to a choice of α of $\frac{p_{in}-p_{out}}{p_{in}-p_{out}} = 0.2$, which performs slightly better than $\alpha = 0.85$. We also see that our intermediate method that utilizes a single covariance matrix for the two classes exhibits a recall that is nearly identical to the fully quadratic method, but achieves this performance with a linear discriminate function.

Continuing with our analysis of generated networks, we examine how well classifications based on our discriminant functions correlate with the underlying true partitioning of the stochastic block models being considered. While exact recovery of planted partitions has been shown possible in regimes where the the two blocks are well separated (where $|p_{in} - p_{out}|$ is large enough) [24, 36, 105], recent work has shown that recovering a partition that is correlated with the underlying partition is still possible even in some contexts when exact recovery is impossible, up to a recently identified resolution limit [34, 84]. For a sparse graph with $n_a = n_b$ nodes and average degree $\langle d \rangle$, recovering a partition correlated with the ground truth partition is possible in the limit of large graphs if and only if: $p_{out}/p_{in} < (\langle d \rangle - \sqrt{\langle d \rangle})/(\langle d \rangle + \sqrt{\langle d \rangle})$. In Figure 4.2B, we show the Pearson correlation r between the recovered partition and the ground truth partition for various discriminant functions on a stochastic block model with $n = 128$ nodes, $n_a = n_b$, and average degree $\langle d \rangle = 16$. The asymptotic resolution limit for these parameters is $p_{out}/p_{in} < 0.6$, and we see that our covariance-adjusted methods are capable of recovering a correlated partition up to this limit. Meanwhile we see that personalized PageRank and heat kernel perform very poorly by comparison.

As a further illustration of the improved performance of our algorithm for recovering partitions correlated with the ground truth, the remainder of Figure 4.2 shows heat maps of the Pearson correlation of various methods as a function of p_{in} and p_{out}. We clearly see that our normalized linear classification performs significantly better through the space of stochastic block model parameters; the quadratic classification (not shown in this figure) produces a heat map that's very similar to the one for the normalized linear classification, though with even slightly higher performance. We note that all these methods require knowledge or estimation of the underlying parameters – in Figure 4.2C we see that when the discriminant function is configured with a fixed set of parameters that are far from the true values the classification can be quite poor.

These performance results above show that when studying networks that originate from a stochastic block model our principled discriminant functions significantly outperform the heuristics of personalized PageRank and heat kernel methods. Next we consider the extent to which our methods also perform well on real-world graphs that do not originate from stochastic block models but instead feature real-world community

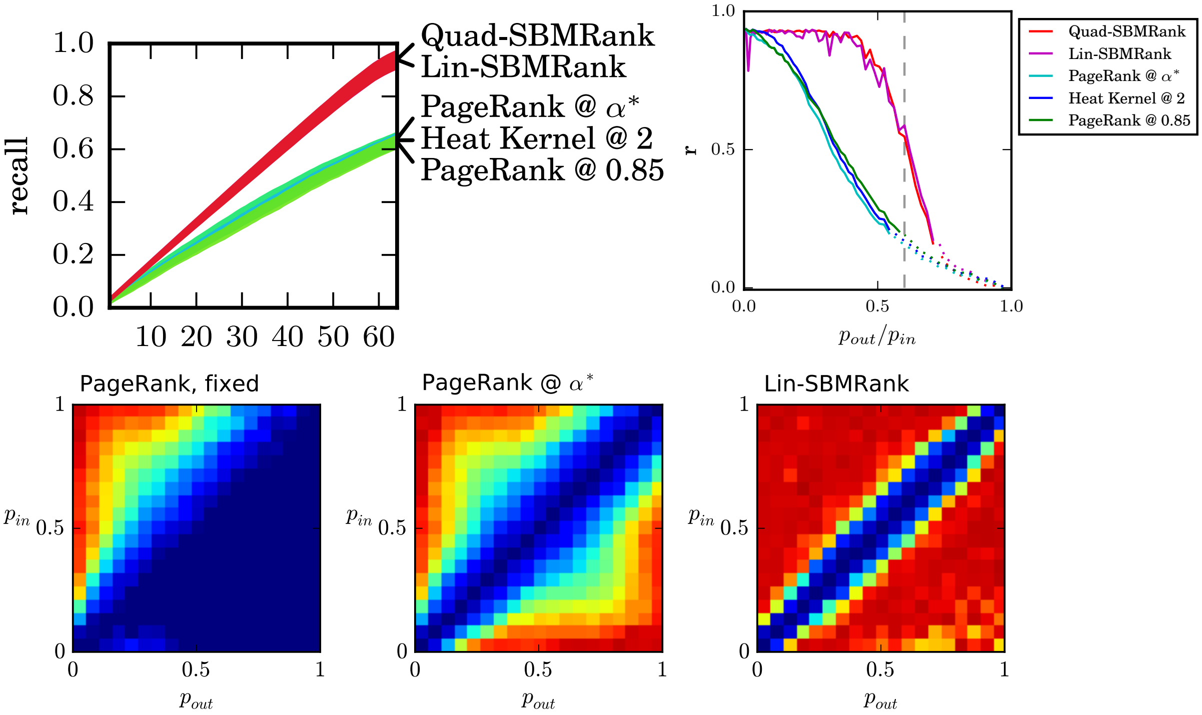


Figure 4.2: (a) The cumulative recall of a stochastic block model (with $n_a = n_b = 64$ nodes and $p_{in} = 0.6$, $p_{out} = 0.4$). Here Quad-SBMRank is our quadratic discriminant classifier and Lin-SBMRank is our normalized linear discriminant classifier. (b) The Pearson correlation r between the recovered model labeling and the true labeling for a stochastic block model on $n = 128$ nodes, with $n_a = n_b = 64$ and expected degree $\langle d \rangle = 16$, as a function of the out/in balance p_{out}/p_{in}. The vertical dashed line shows the asymptotic resolution limit ($p_{out}/p_{in} = 0.6$) of this stochastic block model. We measure the bootstrapped p-values for the correlation r and indicate $p \leq 0.01$ with a solid line and $p > 0.01$ with a dotted line. (c)-(e) Heatmaps: r as a function of $p_{aa} = p_{bb} = p_{in}$ and $p_{ab} = p_{out}$ in four settings. Red is high ($r = 1$) and blue is low ($r = 0$). Using a personalized PageRank discriminant function ranking based on (c) a fixed choice of $p_{in} = 0.6$, $p_{out} = 0.1$ and (d) true values of p_{in} and p_{out}. (e) Using our normalized linear function $g_2(r) = \Sigma^{-1}(a-b)^T r$ for the true values of p_{in} and p_{out}. The quadratic discriminant function, not shown, is similar to the linear function.

structure.

We consider three publicly available network datasets possessing ground truth community labelings: a political blog network exhibiting a partition into two dense clusters of liberal and conservative bloggers [3]; the Berkeley-Stanford host graph [59], the link network between hosts from the UC Berkeley and Stanford University web domains; and the co-authorship network of DBLP [128], a database of computer science papers. The political blogs network contains 1493 blogs of which 758 are liberal and 732 are conservative; the DBLP graph contains 317,080 researchers and 1,049,866 co-authorships, with a total of 13,477 ground truth communities in the dataset. In the case of the political blog and Berkeley-Stanford host network, we investigate performance using 50 different seed sets from each of the two communities; for DBLP, we select a collection of communities of an approximately fixed intermediate size and evaluate performance using seed sets from each of these communities. As in [67], we focus on the 100 communities closest to $(C_{max})^{3/4} = 385$ in size, where C_{max} is the size of the largest community. For each dataset we average our results over 100 recall curves, the number of nodes identified in the target community as a function of the total number m of nodes reported.

The political blog and Berkeley-Stanford host networks are closer to the set-up of the stochastic block model, in that they consists of two principal clusters with high internal density. We note, of course, that it differs from the assumptions of the stochastic block model in that the links are far from randomly generated. For both of these networks we observe (Figure 4.3) the interesting property that our quadratic classification method outperforms personalized PageRank and heat kernel for approximately the latter half of possible m values — roughly once the goal is to find at least half the members of the target community. It is an interesting open question to find a deeper theoretical basis for the relative shapes of the cumulative recall curves, in that the quadratic methods gets off to a slower start than the personalized PageRank and heat kernel heuristics, but the heuristics lose their advantage with increasing m, and are eventually overtaken.

The DBLP dataset has a fairly different structure, in that it contains many overlapping communities, and each community is very small relative to the size of the full graph. As a result, many of the assumptions on which our principled methods are based do not really hold for DBLP. Despite this, we see in Figure 4.3 that our methods are equal in power, to within the threshold of statistical significance, to personalized PageRank and heat kernel. This equivalence holds across the full range of values of m. Thus, even in settings that are quite distant from the assumptions that motivate our methods,

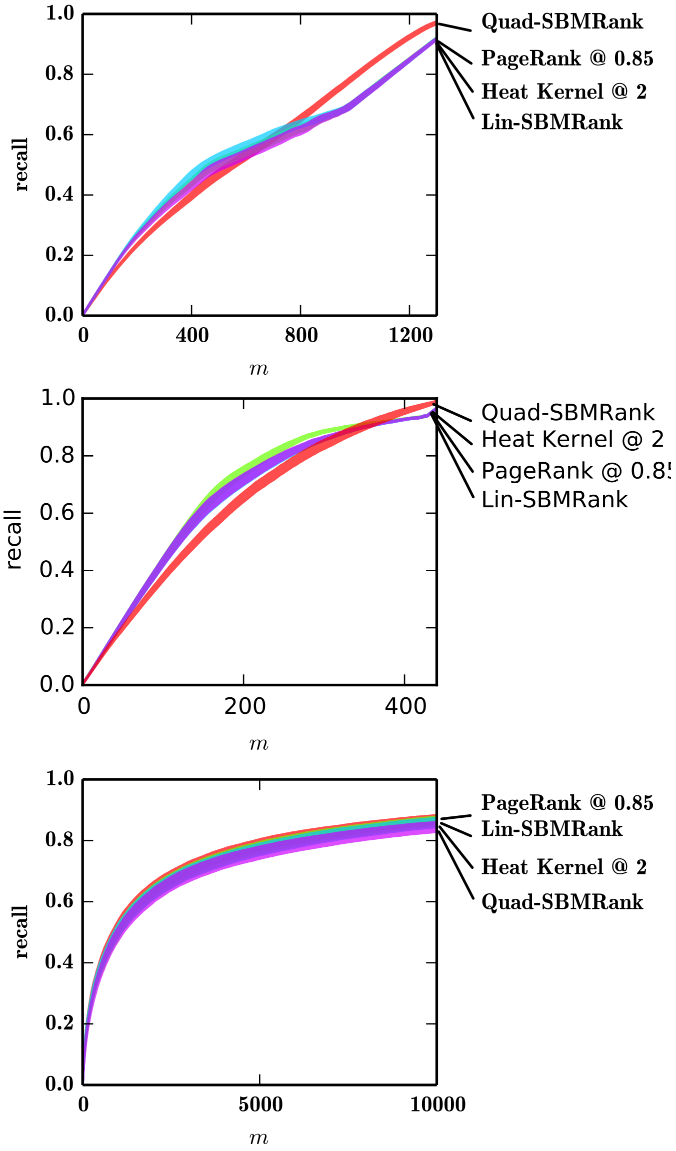


Figure 4.3: The cumulative recall of various discriminant functions on (a) the political blog network, (b) the Berkeley-Stanford host network, and (c) the DBLP co-authorship network. Here Quad-SBMRank is our quadratic discriminant classifier and Lin-SBMRank is our normalized linear discriminant classifier. Tables of values for the recall at specific target expansion sizes and 5%/95% confidence intervals can be found in the supplementary material. For the political blogs network and the Berkeley-Stanford host network we see performance gains from using the quadratic discriminant function classifier at various target expansion sizes.

they provide performance that is comparable to the strongest known heuristics for the problem.

4.3 Discussion

This work contributes a principled motivation for the established success of personalized PageRank as an approach to contextually ranking nodes in graphs. Specifically, we show that it arises as the optimal geometric discriminant function for classifying nodes belonging to a hidden seed community in a stochastic block model. Personalized PageRank and heat kernel based methods both approach contextual ranking by forming linear discriminant functions in the space of random walk landing probabilities. Building on our observed connection between stochastic block models and personalized PageRank, we contribute advanced principled approaches to classification in the space of random walk landing probabilities. Our most advanced classifier uses a quadratic discriminant function that accounts for the full covariance structure of the landing probabilities. We see that it dramatically outperforms personalized PageRank and heat kernel methods for recovering seed sets of synthetic networks generated from stochastic block models, while also matching or slightly outperforming these heuristics in real world networks.

Both the connection between personalized PageRank and stochastic block models and the competitive performance of our advanced principled classifiers on real-world data are genuinely surprising, and we view both the connections and principled avenue for improvements as opening the door on a wide range of new research questions. Can the recent rigorous results for the resolution limit of stochastic block models [84] provide insights into a broader class of contextual ranking problems? Are there other spaces of random walk landing probabilities — such as the landing probabilities of non-backtracking random walks [69] — that can provide additional new approaches to ranking on graphs? Is there a graph model for which heat kernel methods emerge as some optimal choice of discriminant function? There are also a host of further questions that would serve to improve the details of the specific approach we outline here. Can the joint distribution of random walk landing probabilities be modeled more explicitly than by a multivariate Gaussian that approximates just the first two moments? The potential application of our quadratic discriminant classifier to diverse contextual ranking problems also suggests revisiting the broad range of applied problems where personalized PageRank has found previous successes.

4.4 Proof of Lemma 1

Lemma 2. *For any $\varepsilon > 0$, $\delta > 0$, there is an n sufficiently large such that the random landing probabilities $(\hat{a}_1, ..., \hat{a}_K)$ and $(\hat{b}_1, ..., \hat{b}_K)$ for the two blocks of a stochastic block model on n nodes with $n_a = \lambda n$ and $n_b = (1-\lambda)n$, $\lambda \in (0,1)$ fixed and matrix P fixed with $p_{ij} > 0$, $\forall i, j$ satisfy the following conditions with probability at least $1-\delta$ for all $k > 0$:*

$$n_a \hat{a}_k \in \left[(1-\varepsilon)\frac{f_k}{f_k+g_k}, (1+\varepsilon)\frac{f_k}{f_k+g_k}\right] \text{ and } \tag{4.25}$$

$$n_b \hat{b}_k \in \left[(1-\varepsilon)\frac{g_k}{f_k+g_k}, (1+\varepsilon)\frac{g_k}{f_k+g_k}\right], \tag{4.26}$$

where f_k and g_k are given by

$$f_k = \frac{-\lambda_1^k \mathbf{u}_{f1}\mathbf{u}_{g2} + \lambda_2^k \mathbf{u}_{f2}\mathbf{u}_{g1}}{-\mathbf{u}_{f1}\mathbf{u}_{g2} + \mathbf{u}_{f2}\mathbf{u}_{g1}}, \; g_k = \frac{(-\lambda_1^k + \lambda_2^k)\mathbf{u}_{f2}\mathbf{u}_{g2}}{-\mathbf{u}_{f1}\mathbf{u}_{g2} + \mathbf{u}_{f2}\mathbf{u}_{g1}}, \tag{4.27}$$

with

$$\mathbf{u}_f = \begin{pmatrix} \dfrac{(d_{aa}-d_{bb}) - \sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}}{2d_{ba}} \\ 1 \end{pmatrix}, \tag{4.28}$$

$$\mathbf{u}_g = \begin{pmatrix} \dfrac{(d_{aa}-d_{bb}) + \sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}}{2d_{ba}} \\ 1 \end{pmatrix}, \tag{4.29}$$

$$\lambda_1 = \frac{1}{2}\left((d_{aa}+d_{bb}) - \sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}\right), \tag{4.30}$$

$$\lambda_2 = \frac{1}{2}\left((d_{aa}+d_{bb}) + \sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}\right), \tag{4.31}$$

and $d_{ij} = n_i p_{ij}$.

Proof. We first introduce some useful notation. Let us define the following walk counts from the seed to each node, which are random variables under the randomness of the block model:

$$\widehat{A}_k^u = \#\text{ paths from } s \text{ to } u \in V_a \text{ of length } k,$$
$$\widehat{B}_k^u = \#\text{ paths from } s \text{ to } u \in V_b \text{ of length } k.$$

The seed node s is given and fixed, and therefore suppressed in our notation. We denote the number of walks of length k originating at s and ending in V_a and V_b, respectively, as:

$$\widehat{A}_k = \textstyle\sum_{u\in V_a} \widehat{A}_k^u, \qquad \widehat{B}_k = \textstyle\sum_{u\in V_b} \widehat{B}_k^u. \tag{4.32}$$

We see that the random aggregate landing probabilities, respectively the probabilities that a k-step walk starting at a seed node in V_a ends in V_a, and the probability that it ends in V_b, are then:

$$\widehat{a}_k = \frac{1}{n_a}\frac{\widehat{A}_k}{\widehat{A}_k+\widehat{B}_k}, \quad \widehat{b}_k = \frac{1}{n_b}\frac{\widehat{B}_k}{\widehat{A}_k+\widehat{B}_k}, \tag{4.33}$$

where $\hat{w}_k = \hat{a}_k - \hat{b}_k$ are the coordinates of the weight vector we seek to characterize.

Our proof strategy is to show that these two quantities concentrate around expressions given in terms of the solutions f_k and g_k of a recurrence. The values $f_k/(f_k+g_k)$ and $g_k/(f_k+g_k)$ that we will show that they concentrate around are notably not their expectations.

An obstruction to simply taking the expectations of the walk counts $\hat{A}_k$ and $\hat{B}_k$ (and showing concentration around the ratio of expectations) is that counting length-k walks for $k > 1$ requires counting walks that possibly revisit edges, creating a dependence between walk counts of different lengths. The recurrence solutions f_k and g_k that we will analyze can in fact be thought of as the expected walk counts on a slightly different random graph model, where the edges are independently resampled after each walk step. What our analysis effectively shows is that the walk counts on the stochastic block model, our model of interest, concentrate on the expected walk counts of that alternative model. This connection between models is mentioned only as an optional pedagogical tool, and is not essential to understanding our proof.

In Lemma 3 we introduce the recurrence relations:

$$\begin{cases} f_l = d_{aa}f_{l-1} + d_{ab}g_{l-1} \\ g_l = d_{ba}f_{l-1} + d_{bb}g_{l-1} \end{cases} \tag{4.34}$$
$$f_0 = 1, g_0 = 0.$$

and demonstrate that when $d_{ij} > 0, \forall i,j$ we have the general closed-form solutions for f_k and g_k specified by (4.27)–(4.31).

We use Lemma 3 with $d_{ij} = n_i p_{ij}$. As required by the lemma, the matrix $\mathbf{R}$ to be diagonalizable since $n_i > 0, p_{ij} > 0$ for $i,j \in \{a,b\}$ under the assumptions of our proposition statement.

We now return to the walk count random variables $\hat{A}_k$ and $\hat{B}_k$ in a graph G drawn from the stochastic block model. Suppose we are given $\varepsilon > 0$ and $\delta > 0$ as in the statement of the lemma, and we seek bounds for a specific walk length $k \leq K$. We choose $\gamma_2 > 0$ small enough that $(1-\gamma_2)/(1+\gamma_2) \geq 1-\varepsilon$ and $(1+\gamma_2)/(1-\gamma_2) \leq 1+\varepsilon$; we then choose γ small enough that $(1-\gamma)^k \geq 1-\gamma_2$ and $(1+\gamma)^k \leq 1+\gamma_2$.

Let $\hat{M}_{uv}$ be a matrix of independent Bernoulli random variables, indicating the edge event when (u,v) is an edge in the graph G. Notice that $\sum_{u \in V} \hat{M}_{uv}$ is the random out-degree of node v. We observe that for each $j \in \{a,b\}$, each node $v \in V_j$ has in expectation a total of d_{ij} edges to nodes in V_i, where

$$d_{ij} = \mathbb{E}\left[\sum_{u \in V_i} \hat{M}_{uv}\right] = \sum_{u \in V_i} p_{ij} = n_i p_{ij}.$$

When the expectations p_{ij} are fixed in n we can use standard multiplicative Chernoff bounds to bound the probabilities of $4n$ bad events. We have that for any $\gamma > 0$ and any i, $j \in \{a,b\}$:

$$\Pr\left(\sum_{u \in V_i} \hat{M}_{uv} \notin [(1-\gamma)d_{ij}, (1+\gamma)d_{ij}]\right) \leq Ce^{-n} \tag{4.35}$$

for some constant C for any $v \in V_j$. Across all i, j pairs there are $4n$ bad events, and we want to lower bound the probability of there being no bad event. By the union bound we have that

$$\Pr\left(\sum_{u \in V_i} \hat{M}_{uv} \in [(1-\gamma)d_{ij}, (1+\gamma)d_{ij}], \forall v \in V_j, \forall i, j\right) \geq 1 - 4Cne^{-n}. \tag{4.36}$$

Thus, it is clear that for any $\gamma > 0$ and any $\delta > 0$, there exists an n sufficiently large such that the probability that none of the degrees exceed a multiplicative factor of $(1 \pm \gamma)$ is at least $1-\delta$. Assuming this containment succeeds, the rest of the proof argument is deterministic, under the assumption that

$$\sum_{u\in V_i} \hat{M}_{uv} \in [(1-\gamma)d_{ij}, (1+\gamma)d_{ij}], \forall v \in V_j, \forall i, j. \tag{4.37}$$

The next step of our proof strategy is to show that we also have

$$\hat{A}_k \in [(1-\gamma_2)f_k, (1+\gamma_2)f_k] \text{ and } \hat{B}_k \in [(1-\gamma_2)g_k, (1+\gamma_2)g_k] \tag{4.38}$$

whenever the stated containment event holds.

We offer a proof by induction. First we define a new set of variables:

$$\hat{H}_k^u = \begin{cases} \hat{A}_k^u \text{ if } u \in V_a, \\ \hat{B}_k^u \text{ if } u \in V_b. \end{cases} \tag{4.39}$$

We then begin with the base case, furnishing an upper bound on $\hat{A}_1$:

$$\begin{aligned} \hat{A}_1 = \sum_{u\in V_a} \hat{H}_1^u = \sum_{u\in V_a}\sum_{v\in V} \hat{M}_{uv}\hat{H}_0^v & \quad (4.40)\\ = \sum_{v\in V_a} \hat{H}_0^v \sum_{u\in V_a} \hat{M}_{uv} + \sum_{v\in V_b} \hat{H}_0^v \sum_{u\in V_a} \hat{M}_{uv} & \quad (4.41)\\ \le \sum_{v\in V_a} \hat{H}_0^v (1+\gamma)d_{aa} + \sum_{v\in V_b} \hat{H}_0^v (1+\gamma)d_{ab} & \quad (4.42)\\ =(1+\gamma)d_{aa} = (1+\gamma)f_1. & \quad (4.43) \end{aligned}$$

Using a similar set of steps one can easily see that $(1-\gamma)f_1 \le \hat{A}_1$ and $(1-\gamma)g_1 \le \hat{B}_1 \le (1+\gamma)g_1$ also hold.

Next, for our induction we assume that

$$\hat{A}_k \in [(1-\gamma)^k f_k, (1+\gamma)^k f_k], \tag{4.44}$$

$$\hat{B}_k \in [(1-\gamma)^k g_k, (1+\gamma)^k g_k], \tag{4.45}$$

and want to show that the above implies that

$$\hat{A}_{k+1} \in [(1-\gamma)^{k+1} f_{k+1}, (1+\gamma)^{k+1} f_{k+1}], \tag{4.46}$$

$$\hat{B}_{k+1} \in [(1-\gamma)^{k+1} g_{k+1}, (1+\gamma)^{k+1} g_{k+1}]. \tag{4.47}$$

We upper-bound $\hat{A}_{k+1}$:

$$\begin{aligned}
\hat{A}_{k+1} = \sum_{u \in V_a} \hat{H}^u_{k+1} = \sum_{u \in V_a} \sum_{v \in V} \hat{M}_{uv} \hat{H}^v_k & \qquad (4.48) \\
= \sum_{v \in V_a} \hat{H}^v_k \sum_{u \in V_a} \hat{M}_{uv} + \sum_{v \in V_b} \hat{H}^v_k \sum_{u \in V_a} \hat{M}_{uv} & \qquad (4.49) \\
\leq \hat{A}_k (1+\gamma) d_{aa} + \hat{B}_k (1+\gamma) d_{ab} & \qquad (4.50) \\
\leq (1+\gamma)^{k+1} f_k d_{aa} + (1+\gamma)^{k+1} g_k d_{ab} & \qquad (4.51) \\
= (1+\gamma)^{k+1} f_{k+1}, & \qquad (4.52)
\end{aligned}$$

where in the last inequality we use the induction hypothesis. We observe that $\hat{A}_{k+1} \leq (1+\gamma)^{k+1} f_{k+1}$, and similar steps furnish the lower bound $(1-\gamma)^{k+1} f_{k+1} \leq \hat{A}_{k+1}$ and that $(1-\gamma)^{k-1} g_{k+1} \leq \hat{B}_{k+1} \leq (1-\gamma)^{k+1} g_{k+1}$, completing the proof by induction. As a result, we have:

$$\hat{A}_k \in [(1-\gamma)^k f_k, (1+\gamma)^k f_k] \text{ and } \hat{B}_k \in [(1-\gamma)^k g_k, (1+\gamma)^k g_k]. \tag{4.53}$$

Since γ and γ_2 were chosen such that $(1-\gamma)^k \geq 1-\gamma_2$ and $(1+\gamma)^k \leq 1+\gamma_2$ we then have that

$$\hat{A}_k \in [(1-\gamma_2) f_k, (1+\gamma_2) f_k] \text{ and } \hat{B}_k \in [(1-\gamma_2) g_k, (1+\gamma_2) g_k], \tag{4.54}$$

as desired in (4.38). We also have that for $\hat{A}_k + \hat{B}_k$,

$$\hat{A}_k + \hat{B}_k \in [(1-\gamma_2)(f_k + g_k), (1+\gamma_2)(f_k + g_k)]. \tag{4.55}$$

Finally, since ε satisfies $(1-\gamma_2)/(1+\gamma_2) \geq 1-\varepsilon$ and $(1+\gamma_2)/(1-\gamma_2) \leq 1+\varepsilon$, we have

$$\frac{\hat{A}_k}{\hat{A}_k+\hat{B}_k} \in [(1-\varepsilon)\frac{f_k}{f_k+g_k},(1+\varepsilon)\frac{f_k}{f_k+g_k}],$$
$$\frac{\hat{B}_k}{\hat{A}_k+\hat{B}_k} \in [(1-\varepsilon)\frac{g_k}{f_k+g_k},(1+\varepsilon)\frac{g_k}{f_k+g_k}].$$

This final containment holds whenever the original containment in (4.37) holds, with probability at least $1-\delta$, completing the proof.

□

4.5 Proof of Lemma 2

Lemma 3. *Suppose that the matrix* $R = \begin{pmatrix} d_{aa} & d_{ba} \\ d_{ab} & d_{bb} \end{pmatrix}$*, where* $d_{ij} \geq 0$*, is diagonalizable and that* $\mathbf{u}_f$ *and* $\mathbf{u}_g$ *are the eigenvectors that correspond to eigenvalues* λ_1 *and* λ_2 *of* R*:*

$$R = UD^kU^{-1}\, for\, D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix};\, U = \begin{pmatrix} \mathbf{u}_f & \mathbf{u}_g \end{pmatrix}. \tag{4.56}$$

Then

$$\mathbf{u}_f = \begin{pmatrix} \frac{(d_{aa}-d_{bb})-\sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}}{2d_{ba}} \\ 1 \end{pmatrix}, \tag{4.57}$$

$$\mathbf{u}_g = \begin{pmatrix} \frac{(d_{aa}-d_{bb})+\sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}}{2d_{ba}} \\ 1 \end{pmatrix}, \tag{4.58}$$

$$\lambda_1 = \frac{1}{2}\left((d_{aa}+d_{bb})-\sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}\right), \tag{4.59}$$

$$\lambda_2 = \frac{1}{2}\left((d_{aa}+d_{bb})+\sqrt{(d_{aa}-d_{bb})^2+4d_{ab}d_{ba}}\right), \tag{4.60}$$

Further, the closed-form solution to the two-dimensional, first-order recurrence relations,

$$\begin{cases} f_l = d_{aa}f_{l-1} + d_{ab}g_{l-1} \\ g_l = d_{ba}f_{l-1} + d_{bb}g_{l-1} \end{cases} \tag{4.61}$$
$$f_0 = 1, g_0 = 0.$$

is given by

$$f_k = \frac{-\lambda_1^k \mathbf{u}_{f1}\mathbf{u}_{g2} + \lambda_2^k \mathbf{u}_{f2}\mathbf{u}_{g1}}{-\mathbf{u}_{f1}\mathbf{u}_{g2} + \mathbf{u}_{f2}\mathbf{u}_{g1}},\ g_k = \frac{(-\lambda_1^k + \lambda_2^k)\mathbf{u}_{f2}\mathbf{u}_{g2}}{-\mathbf{u}_{f1}\mathbf{u}_{g2} + \mathbf{u}_{f2}\mathbf{u}_{g1}}. \tag{4.62}$$

Proof. We begin with the original recurrence relations:

$$f_k = n_a p_{aa} f_{k-1} + n_b p_{ba} g_{k-1}, \tag{4.63}$$
$$g_k = n_a p_{ab} f_{k-1} + n_b p_{bb} g_{k-1}. \tag{4.64}$$

These recurrences form a first-order homogeneous matrix recurrence:

$$\begin{bmatrix} f_k \\ g_k \end{bmatrix} = \begin{bmatrix} n_a p_{aa} & n_b p_{ba} \\ n_a p_{ab} & n_b p_{bb} \end{bmatrix} \begin{bmatrix} f_{k-1} \\ g_{k-1} \end{bmatrix}. \tag{4.65}$$

Letting $\mathbf{R} = \begin{bmatrix} d_{aa} & d_{ba} \\ d_{ab} & d_{bb} \end{bmatrix}$ and $\mathbf{C}_k = \begin{bmatrix} f_k \\ g_k \end{bmatrix}$ we have the simple recursion $\mathbf{C}_k = \mathbf{R}\,\mathbf{C}_{k-1}$. By induction we have that $\mathbf{h}_k = \mathbf{R}^k\mathbf{h}_0$, where $\mathbf{h}_0$ are the initial conditions.

We seek to diagonalize $\mathbf{R}$. When $\mathbf{R}$ is diagonalizable we have $\mathbf{R}^k = (UDU^{-1})^k = UD^kU^{-1}$, where D is a diagonal matrix with the eigenvalues of $\mathbf{R}$, λ_1 and λ_2, along the diagonal, and U is a matrix with the corresponding eigenvectors of $\mathbf{R}$, $\mathbf{u}_f$ and $\mathbf{u}_g$, as its columns. We will derive U and D exactly (below) and thus can derive $\mathbf{h}_k$ exactly for all k.

When the parameters are such that $\mathbf{u}_f$ and $\mathbf{u}_g$ are linearly dependent, U^{-1} does not exist and diagonalization is not possible. In Table 4.1 we describe the parameter sets for which diagonalization is not possible, noting that no such parameter sets exist when all the entries of R are positive. As long as the parameter values do not satisfy the conditions described in Table 4.1, diagonalization is possible and the recurrence has the following general solution:

$$f_k = \frac{\lambda_f \mathbf{u}_{f1}(g_0\mathbf{u}_{g1} - f_0\mathbf{u}_{g2}) - \lambda_g \mathbf{u}_{g1}(g_0\mathbf{u}_{f1} - f_0\mathbf{u}_{f2})}{\mathbf{u}_{f2}\mathbf{u}_{g1} - \mathbf{u}_{f1}\mathbf{u}_{g2}} \tag{4.66}$$

$$g_k = \frac{\lambda_f \mathbf{u}_{f2}(g_0\mathbf{u}_{g1} - f_0\mathbf{u}_{g2}) - \lambda_g \mathbf{u}_{g2}(g_0\mathbf{u}_{f1} - f_0\mathbf{u}_{f2})}{\mathbf{u}_{f2}\mathbf{u}_{g1} - \mathbf{u}_{f1}\mathbf{u}_{g2}}. \tag{4.67}$$

By plugging in the initial conditions $f_0 = 1$ and $g_0 = 0$ we can check that Equations (4.66) and (4.67) reduce to the Equations in (4.62).

□

$n_a p_{aa}$	$n_b p_{bb}$	$n_a p_{ab}$	$n_b p_{ba}$
≥ 0	$n_a p_{aa}$	> 0	0
≥ 0	$n_a p_{aa}$	0	> 0

Table 4.1: Parameter values for which **R** is not diagonalizable. Note that these parameters are ruled out if nodes have the same expected degree , as in the case of interest, or if the graph is undirected. The first two columns indicate that the expected in-degree of nodes in the two communities must be equal. The last two columns indicate that there must be a strictly positive expected number of edges going from one community to the other, whilst there are strictly no edges going in the reverse direction. Note that these parameter values cannot be achieved for undirected graphs. Assuming that $n_a > 0$ and $n_b > 0$ the first set of values correspond to all probabilities but p_{ab} being 0and $p_{ab} > 0$, or all but p_{ba} being strictly positive and p_{ba} being 0.

4.6 SBM Parameter estimation

The following consistent estimators for the parameters of a stochastic block model $G((n_a, n_b), P)$ where $p_{aa} = p_{bb} = p_{in}$, $p_{ab} = p_{out}$, also known as the affiliation model, are due to Allman et al. [5]:

$$\hat{p}_{out} = \frac{(s_3 - s_2 s_3)m_1^3 + (s_2^3 - s_3)m_2 m_1 + (s_3 s_2 - s_2^3)m_3}{(m_1^2 - m_2)(2s_2^3 - 3s_3 s_2 + s_3)}, \tag{4.68}$$

$$\hat{p}_{in} = \frac{m_1 + (s_2 - 1)\hat{p}_{out}}{s_2}, \tag{4.69}$$

where

$$s_2 = n_a^2 + n_b^2, \tag{4.70}$$

$$s_3 = n_a^3 + n_b^3, \tag{4.71}$$

$$m_1 = \frac{1}{n(n-1)} \sum_{i,j=1,i\neq j}^{n} X_{ij}, \tag{4.72}$$

$$m_2 = \frac{1}{n(n-1)(n-2)} \sum_{i,j,k=1,i\neq j\neq k}^{n} X_{ij} X_{ik}, \tag{4.73}$$

$$m_3 = \frac{1}{n(n-1)(n-2)} \sum_{i,j,k=1,i\neq j\neq k}^{n} X_{ij} X_{ik} X_{jk}. \tag{4.74}$$

4.7 Details of Belief Propagation

Belief Propagation (BP) is a message passing algorithm for the inference of graphical models (joint distributions of random variables with conditional dependencies represented by graphs). Belief Propagation infers marginal distributions of the unobserved variables (nodes) of such models. When the graph underlying a graphical model is a tree, BP known to converge on a fixed point that minimizes an objective function known as the Bethe Free Energy of the joint probability distribution of the model. Belief Propagation has also been widely applied beyond the context of trees, in what is sometimes called "loopy belief propagation," named after the presence of cycles (”loops”) in the graph.

Belief Propagation has recently been adapted for the inference of latent labels for graphs realized from Stochastic Block Models (SBMs) [Decelle 2012]. The connection between BP and SBMs follows from the fact that the Bethe Free Energy of the joint probability distribution of the latent labels on a realization of an SBM – what BP minimizes on trees – is an upper bound on the negative log-likelihood of the SBM parameters, and it can be shown that any global minimum of the free energy is a global minimum of the negative log-likelihood ["14"]. If a graph realized under the SBM is a tree then Belief Propagation will quickly converge upon the maximum likelihood estimate of the SBM parameters (these parameters include the assignment of nodes to block classes). For a graph realized under the SBM that is not a tree, the convergence upon the global maximum of the likelihood under BP is no longer guaranteed. In fact, BP is not guaranteed to converge on any solution at all, though some sufficient conditions for convergence are known. Despite the lack of rigorous results, BP is widely understood to still find good solutions (solutions with near-maximal likelihood) in practice even when the graph realized by an SBM is not a tree.

Belief Propagation for SBMs is a much more empowered algorithm than the classification algorithms SBMRank and QuadSBMRank that we present and study in the main text of this work. The algorithms we present and study are all restricted to simple discriminative classification rules in the space of random walk landing probabilities, classifying individual nodes independently based on these probabilities. In contrast, Belief Propagation performs a global joint inference of all node labels.

Furthermore, BP has only been shown to do well on very simple SBMs, very little data (e.g. karate), and the assumptions for BP-SBM to work are strong (see recent work trying to relax class balance). TODO: write this paragraph once we formalize our results for equal out/in-degree vs non-uniform degree.

TODO Meanwhile, algorithms other than BP have recently been proposed that are poly-time and rigorus results show they reach the resolution limit... Mossel-Neeman-Sly 15, Massouli STOC14. These algo's are not practical. See p4 ("nor have we implemented it") of MNS and also their comment about M14 on p7.

4.7.1 BP-SBM algorithm

We now present a distilled derivation of SBM-BP and sparse-SBM-BP, for completeness. The existing literature is messy/physicsy.

...producing an algorithm we will call SBM-BP.

In graphs that are sparse, a heuristic can be applied to BP to yield an efficient algorithm requiring just $O(m)$ time steps instead of $O(n^2)$.

[Taylor approximations]

...we will call this latter algorithm sparse-SBM-BP.

4.8 Numerical considerations for Covariance Matrices

The covariance matrices describe the covariances between the landing probabilities for a random walk starting at the seed node and walking 1 to K steps. For large step counts the landing probabilities begin to converge upon the stationary distribution of a random walk, meaning that the covariance between step $K-1$ and K becomes very high. In general the last several columns of the covariances matrices become strongly collinear for large values of K.

To mitigate against ill-conditioned matrix inversions, we restrict the maximum number of steps K included in our landing probability space in a manner that keeps the condition numbers $\kappa(\Sigma_a)$ and $\kappa(\Sigma_b)$ both below 10^{10}. In practice this empirically amounts to performing our analysis in the space of landing probabilities for the first $K = 6$ steps.

4.9 Details of EM Algorithm

The theoretical results in Section 2 of the main text assumed that the parameters (n_a, n_b) and P of our two-block stochastic block model were known, allowing us to derive the theoretical centroids a and b as a function of these parameters. We note that we could also, in principle, derive the theoretical covariance matrices Σ_a, Σ_b as a function of these parameters (n_a, n_b) and P. Such a derivation would however have little value in learning contexts where the parameters of the presumed underlying stochastic block model would anyway be unknown.

In this supplement we present a straightforward expectation maximization (EM) algorithm for learning the parameters of the descriptive model of the landing probability point clouds that emerge from a stochastic block model. The algorithm we present is the standard EM algorithm for learning the parameters of a mixture of Gaussians. We note that we are not making an assumption of Gaussianity in the space of landing probabilities, but simply choosing to model the landing probability points clouds by their first two moments (means and variances), as opposed to merely their first moments (means).

We obtain from a graph $G = (V, E)$ a set of landing probability vectors $\{r^u\}_{u \in V}$ that we conceive of as a mixture of two multivariate Gaussians, aiming to learn the parameters. Let a and b be the centroids, Σ_a and Σ_b be the respective covariances matrices, and π be the proportion of nodes in the seed node block. Let $z^u \in \{0, 1\}$ be the unobserved membership assignments for each u. Our EM algorithm, iterating in steps $t = 1, ..., T$ is then as follows:

E-step:

$$\begin{aligned} z^u_{(t+1)} &= \frac{\pi_{(t)} p(r^u | z = 1, a_{(t)}, \Sigma_{a,(t)})}{\pi_{(t)} p(r^u | z = 1, a_{(t)}, \Sigma_{a,(t)}) + (1 - \pi_{(t)}) p(r^u | z = 0, b_{(t)}, \Sigma_{b,(t)})} \\ &= \frac{1}{1 + \frac{1-\pi_{(t)}}{\pi_{(t)}} \frac{p(r^u | z=0, b_{(t)}, \Sigma_{b,(t)})}{p(r^u | z=1, a_{(t)}, \Sigma_{a,(t)})}} \end{aligned} \tag{4.75}$$

M-step:

$$\pi_{(t)} = \frac{1}{|V|} \sum_{u \in V} z^u_{(t)}, \tag{4.76}$$

$$a_{(t)} = \frac{\sum_{u \in V} z^u_{(t)} r^u}{\sum_{u \in V} z^u_{(t)}}, \tag{4.77}$$

$$\Sigma_{a,(t)} = \frac{\sum_{u \in V} z^u_{(t)} (r^u - a_{(t)})(r^u - a_{(t)})^T}{\sum_{u \in V} z^u_{(t)}}, \tag{4.78}$$

$$b_{(t)} = \frac{\sum_{u \in V} (1 - z^u_{(t)}) r^u}{\sum_{u \in V} (1 - z^u_{(t)})}, \tag{4.79}$$

$$\Sigma_{b,(t)} = \frac{\sum_{u \in V} (1 - z^u_{(t)})(r^u - b_{(t)})(r^u - b_{(t)})^T}{\sum_{u \in V} (1 - z^u_{(t)})}. \tag{4.80}$$

In the E-step above we have provided a manipulation that improves numerical stability.

The EM iteration procedure requires initialization, where we can choose to either initialize with a full set of in-seed community assignment weights $(z^1_{(0)}, ..., z^n_{(0)})$, or initialize with a full set of parameter estimates $\pi_{(0)}, a_{(0)}, b_{(0)}, \Sigma_{a,(0)}$, and $\Sigma_{b,(0)}$. The approach to initialization that we take in this work is in fact halfway between these two choices, making use of our earlier observation that the commonly employed personalized PageRank ranking procedure, configured with some choice of α, corresponds to a restricted version of our discriminant function that disregards the covariance structure.

We assume that we have a target size community that we are seeking to construct, which determines (n_a, n_b) in the block model and therefore $\pi_{(0)} = \frac{n_a}{n_a + n_b}$ in the initialization. In so much as we are comfortable prescribing a value of α for personalized PageRank, we can use the equations in (6) from the main text to initialize our EM algorithm given that choice of α by setting:

$$a_{k,(0)} = \frac{1}{n_a + n_b}(1 + \frac{n_b}{n_a}\alpha^k), \quad b_{k,(0)} = \frac{1}{n_a + n_b}(1 - \alpha^k). \tag{4.81}$$

For the covariance matrices, a basic approach to initialization would be to simply use the identity matrices $\Sigma_{a,(0)} = \Sigma_{b,(0)} = I$ that personalized PageRank would use. In our more advanced approach, we consider a stochastic block model with initial choices of (n_a, n_b) and P (with P chosen to correspond to the the choice of α) and simulate a stochastic block model and initialize the matrices $\Sigma_{a,(0)}$ and $\Sigma_{b,(0)}$ with the sample covariance matrices of the blocks V_a and V_b from the simulation. This initialization performs slightly better in practice than initializing with identity matrices.

From these initial choices of the parameters π, a, b, Σ_a, and Σ_b, we can now compute the first E-step to deduce the in-seed community assignment weights imputed by personalized PageRank, and then undertake our EM iteration from there.

4.9.1 Equal covariance special case

In situations were we seek a purely linear discriminant function, we can achieve this goal while still pursuing covariance adjustment by estimating a single common covariance matrix for the two classes. The covariance estimation M-step for the covariance matrix is then:

$$\begin{aligned}\Sigma_{(t)} = \frac{1}{|V|}\sum_{u\in V}[z^u_{(t)}(r^u - a_{(t)})(r^u - a_{(t)})^T \\ +(1-z^u_{(t)})(r^u - b_{(t)})(r^u - b_{(t)})^T],\end{aligned} \tag{4.82}$$

and the procedure is otherwise analogous, using $\Sigma_{(t)}$ in place of $\Sigma_{a,(t)}$ and $\Sigma_{b,(t)}$ in the E-step. The resulting discriminant function is then $g_2(r) = [\Sigma^{-1}(a-b)]^T r + C$, as in Equation (12) from the main text, and we call this our *normalized linear discriminant*, reflecting the presence of the inverse covariance matrix to normalize relative variances, but the absence of any quadratic term due to cancellations between the identical covariance matrices.

4.9.2 Numerical considerations

In the above EM algorithm, we provided a simple manipulation that improves the numerical stability of the E-step. As a more significant numerical consideration, the probabilities $p(r^u|z=1,a,\Sigma_a)$ and $p(r^u|z=0,b,\Sigma_b)$ both involve expressions that invert the covariance matrices Σ_a and Σ_b. These inversions occur both in the quadratic and linear portions of the discriminant functions, meaning that even when we impose that $\Sigma_a = \Sigma_b$, we are forced to tread carefully in order to maintain numerical stability. UPDATE AND SYNC WITH OTHER COMMENTS.

4.10 Numerical Results for SBMs and Real-World Networks

Here we supplement the graphical performance results presented in Section 3 of the main text with tables of numerical performance values and the 5%/95% confidence intervals. These results are discussed in the main text, with the discussion corresponding to Figure 2A (the SBM), and Figure 3 (the real-world networks).

Part II

Data mining

CHAPTER 5

THE LIFECYCLES OF APPS IN A SOCIAL ECOSYSTEM

This chapter is written in collaboration with Lada Adamic, Shaomei Wu, and Jon Kleinberg and was published in Proceedings of the 24th International Conference on World Wide Web in 2015.

Apps are emerging as an important form of on-line content, and they combine aspects of Web usage in interesting ways — they exhibit a rich temporal structure of user adoption and long-term engagement, and they exist in a broader social ecosystem that helps drive these patterns of adoption and engagement. It has been difficult, however, to study apps in their natural setting since this requires a simultaneous analysis of a large set of popular apps and the underlying social network they inhabit.

In this work we address this challenge through an analysis of the collection of apps on Facebook Login, developing a novel framework for analyzing both temporal and social properties. At the temporal level, we develop a retention model that represents a user's tendency to return to an app using a very small parameter set. At the social level, we organize the space of apps along two fundamental axes — *popularity* and *sociality* — and we show how a user's probability of adopting an app depends both on properties of the local network structure and on the match between the user's attributes, his or her friends' attributes, and the dominant attributes within the app's user population. We also develop models that show the importance of different feature sets with strong performance in predicting app success.

5.1 Introduction

There is, or is likely soon to be, a webservice or app for virtually every component of modern life. They are diverse and ubiquitous; they constitute both a backdrop and chronicle of everyday experience. And they represent a broad change in overall patterns of Internet use — both the research community and the media have increasingly begun discussing the "appification of the Web" [1]. Yet empirical opportunities to consider them as a complete ecosystem have been limited, and as a result we still know very little about the population structure of apps — their inherent diversity, their lifecycles, and the ways in which users engage with them.

The high-level characteristics of app engagement as a form of Web use are still the subject of much discussion and refinement, but certain properties emerge independent of any one particular app's functionality — these include *temporal* properties, based on long-running patterns of individual usage and engagement over time, and *social properties*, in which an individual will typically be a user of many apps with overlapping functionality, in a broader social environment that is bootstrapped to create within-app social activity.

To address these issues, we study the collection of apps on Facebook Login, making use of anonymized aggregate daily usage logs of the apps and web services accessible through this mechanism. We undertake our analysis on two levels of scale — the individual level, focusing on the properties of user behavior over time and in relation to other users; and the app level, modeling the overall usage level of the app and the social structure on its users.

At the temporal level, we develop a user retention model, showing how with a small number of parameters we can approximate the probability that a user who adopts an app at time t will continue to be using it at a future time $t+\Delta$. The model exposes the ways in which usage decay has a time-dependent component, and provides us with a compact set of parameters representing a particular app's engagement profile that can then be used in higher-level tasks. When we consider the app's user population as a whole, we are led to natural lifecycle modeling and prediction questions — given an app's history up to a given point in time, how well can we predict its number of users going forward? Interesting recent work of Ribeiro [98] considered this question using time-series data for several large Web sites; we show how a broad range of feature categories — including our derived retention parameters, together with individual characteristics

[1] https://sites.google.com/site/appweb2012/

of the app's users and the social network structure on its full user population — can lead to strong prediction results across a wide diversity of apps.

At the level of app social structure, we show how the space of all the popular apps on Facebook Login can be organized in a two-dimensional representation whose axes correspond to *popularity* — the raw number of users — and *sociality* — the extent to which users of the app have friends who are also users of the app. This representation exposes certain global organizing principles in the full app population, including a pair of complementary "frontiers" to the space — one containing apps whose sociality is relatively fixed independent of their popularity, and one in which the sociality of the app's user population is not much greater than that of a random set of Facebook users of comparable size.

Finally, we perform an analysis of social characteristics at the individual user level, analyzing the Facebook users who are one step away from an app in the social network — a set we can think of as the "periphery" of the app, containing people who are not yet users of the app, but have friends who are users. For a person in an app's periphery, we can attempt to predict future adoption of the app based on individual characteristics and network structure. We find that apps are diverse in the way in which the structure on a user's friends is related to adoption probabilities, and we find an interesting effect in the interactions among individual characteristics: a user's probability of adopting an app depends on the three-way relationship among their own attributes, the attributes of their friends who use the app, and the modal attributes of the full population of app users.

5.2 Data

The data for this study comes from anonymized logs of Facebook Login daily activity, collected between January 2009 and June 2014. Facebook Login is a secure way for Facebook users to sign into their apps without having to create separate logins.

The various analyses in this paper required different slices of these logs, both considering the observation window and the apps being observed. Table 1 summarizes the different subsampled data sets that will be referred to throughout this work. The data for this study has granularity of one day; that is, we have logs about whether an individual uses a specific app on each day. All user level data is de-identified.

tag	selection criteria	time period	size
APPS_{RAND}	random $\propto$ DAU(2014-06)	Jan. 2014 - Jun. 2014	83,000 apps
APPS_{POP}	most popular by MAU(2013-06)	Jan. 2009 - Jun. 2014	2,319 apps 1.4×10^9 users

Table 5.1: Summary of data sets considered in this paper. DAU and MAU refer to Facebook Daily Active Users and Monthly Active Users, respectively. Our initial overview analyses consider APPS_{RAND}, while our subsequent and in-depth analyses consider APPS_{POP} and, as occasion permits, various subsets of it ($\text{APPS}_{POP\{X\}}$). Unless otherwise noted, subsampling in this work is done on apps, not on users.

The frequency with which the Facebook Login service is called, and hence daily activity is registered, depends on several factors. Web-based activity relies on authentication tokens that expire on the order of hours, while mobile apps can optionally request tokens that are valid for days, provided the user does not change their password. For some apps, we do see a periodic activity, typically 7 days apart, consistent with longer-term authentication tokens being refreshed. This periodicity is a small effect relative to the overall activity, as we show below. This is likely because other activity, such as posting updates or retrieving public profiles or friend lists, again requires reconnecting. Therefore, Facebook Login provides a reasonable proxy of daily use of the app. It allows us to characterize the app's adoption and retention.

5.3 Social Properties of Apps

5.3.1 Popularity and sociality

One question that has been raised previously is how big of a role the social network plays in the adoption of apps. This parameter has been inferred indirectly by Onnela and Reed-Tsochas [88] in their study of the very early adoption of Facebook apps. It is also estimated in the model proposed by Ribeiro [99], where individuals can drive their friends' adoption and re-engagement.

However, these prior studies did not directly measure whether app adoption was in fact correlated on the network, and so we turn to this task presently. In particular, we would like to place apps in a low-dimensional space that can provide a view for how they are distributed across the social network of users.

To do this, we begin with two basic definitions

- We say that the *popularity* of an app, denoted $p(x)$, is the probability that an individual selected uniformly at random from Facebook's population is a user of the app.
- We say that the *sociality* of the app, denoted $p(x|y)$, is the probability that a member of Facebook is a user of the app given that they have at least one friend using the app.

Studying the distributions of $p(x)$ and $p(x|y)$, and how they are jointly distributed across apps, allows us to ask a number of questions. In particular, how socially clustered is the app? And how does it depend on the type of app, or characteristics of the app's users?

Note that if $p(x|y)$ is very high for an app, it means that its user population in a sense "conforms" to the structure of the underlying social network.

Moreover, $p(x|y)$ can in principle be high even when $p(x)$ is low — this would correspond to an app that is popular in a focused set of friendship circles, but not on Facebook more broadly. On the other hand, if $p(x|y)$ is not much more than $p(x)$, then it says that users of the app are spread out through the social network almost as though each member of Facebook independently flipped a coin of bias $p(x)$ in order to decide whether to become a user of the app — there would be no effect of the social network at all.

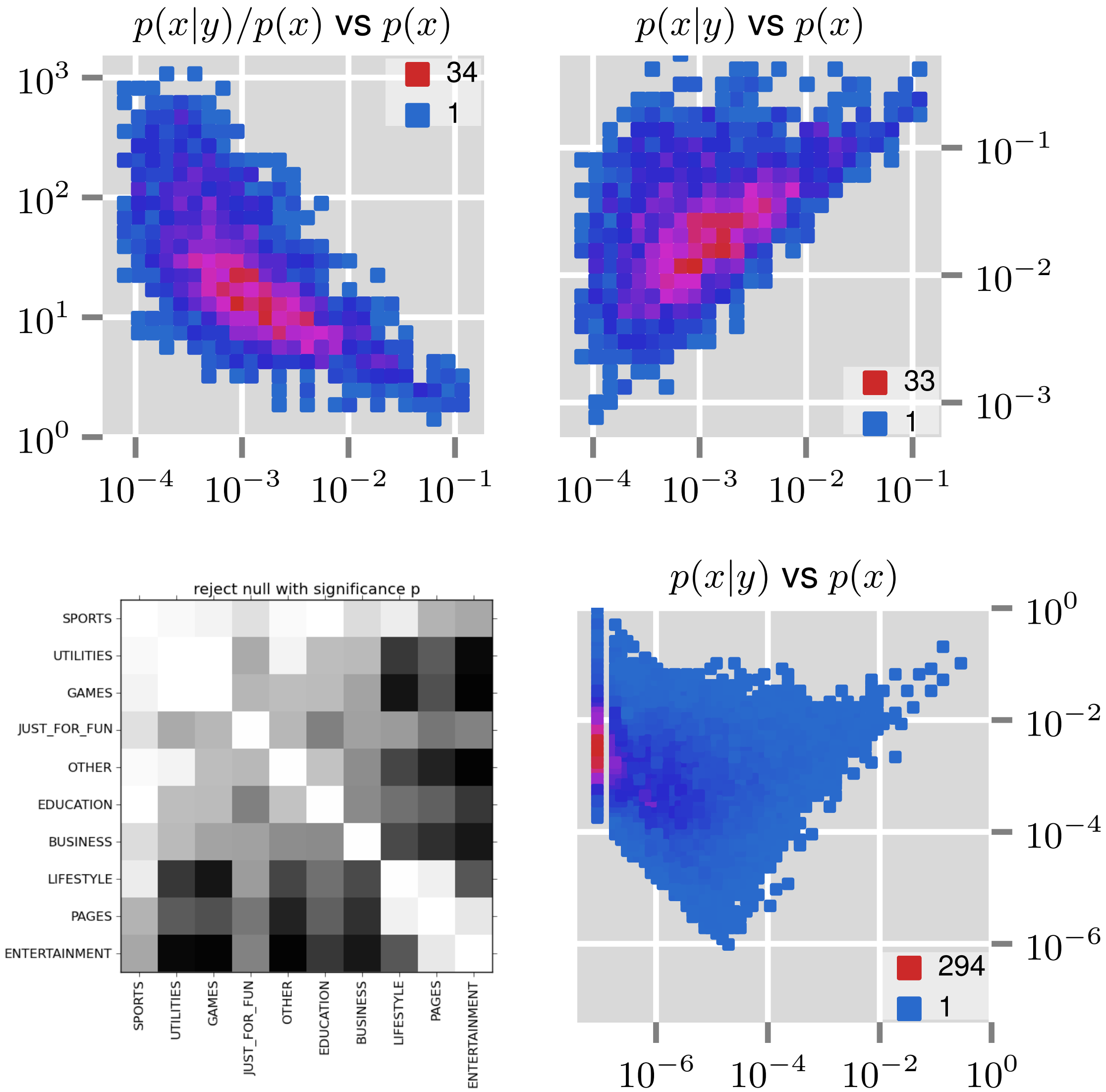


Figure 5.1: App sociality. **Top left:** Horizontal axis is app popularity, and vertical axis is the relative increase in adoption likelihood for people who have friends who also use the app. **Right panels:** Horizontal axis is app popularity, vertical axis is app sociality. The colors represent the number of apps falling within the given bin. The lower right panel uses APPS$_{RAND}$, while the other three panels use APPS$_{POP}$ (see Table 6.1 for details). The labeled colors indicate the relative frequencies of observations in each bin, such that the lowest values have been normalized to 1. **Bottom left:** Matrix indicating the *p*-values for the two sided Kolmogorov-Smirnov test comparing the distributions $p(x|y)/p(x)$ for apps within each pair of the nine listed categories. White indicates a lower *p*-value and black indicates a higher one.

Plotting apps in popularity-sociality space

An appealing feature of this pair of parameters is that it provides a natural two-dimensional view of the space of all popular apps on Facebook. We show this view in Figure 5.1 — a heat map showing the density of apps at each possible (discretized) pair of values $(p(x), p(x|y))$.

We see in Figure 5.1 that the apps fill out a wedge-shaped region in the $p(x)$-$p(x|y)$ plane, and it is informative to understand what the boundaries of the region correspond to. First, note that if the social network had no relationship to app usage, we would see the diagonally sloping line $p(x) = p(x|y)$; in the plot this corresponds to a line that lies slightly below the diagonal lower boundary of the points in the heat map. Thus, there exists a frontier in the space of apps that is almost completely asocial — those apps that lie parallel to this diagonal line — but essentially no apps actually reach the line; even the most asocial apps exhibit some social clustering. We see this in the approximately horizontal top boundary of the points in the heat map — this is a frontier in the space of apps where knowing that a person x has a friend using the app gives you a fixed probability that x uses it, independent of the app's overall popularity on Facebook. The location of this horizontal line is interesting, since it provides an essential popularity-independent value for the maximal extent of social clustering that we see on Facebook.

Note that the wedge-shaped region in a sense has to come to a point on the right-hand side, as $p(x)$ becomes very large: once an app is extremely popular, there is no way to avoid having pairs of friends using it almost by sheer force of numbers. And given the crowding of app users into the network, there is also no way for the extent of social clustering to become significantly larger than one would see by chance.

The third, lower left boundary of the wedge is a manifestation of the Facebook friend limit of 5000: if a user is friends with someone who uses the app, at least 1/5000 of their friends use it. We approach this limit in the far left hand of this Figure with apps that have two users, one with 0 other friends using the app and the other with 1 of their nearly 5000, combining to approach the lower limit of $1/(2*5000)$. The lower bound decreases as $1/(n*5000)$ as the number of users, n, increases.

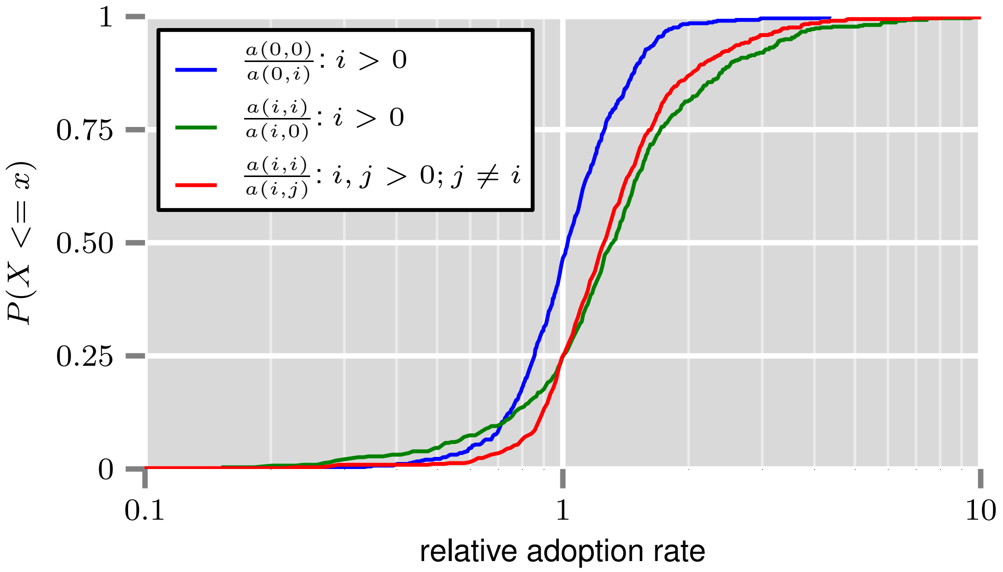


Figure 5.2: Relationship between national identity of potential app adopters, that of their current user friends, and the likelihood of their adopting the app. Blue: Relative adoption rates when a potential user and their current user friend are from the majority to when potential user is in majority and current user friend is in minority. Green: Relative rates when potential user and current user friend are in same minority to when potential user is in minority and current user friend is in majority. Red: Relative rates for when potential user and current user friend are in same minority to when they are in different minorities. The blue curve indicates that when a potential user is from the majority country, their current user friend could be from either the majority or a minority and they are still equally likely to adopt an app more often as less. In contrast, when the potential user is from a minority country, in 75% of apps they adopt more frequently when their current user friend is from the same country as them.

5.3.2 Analysis of Social Neighborhoods

We saw in the previous section that app adoption can be localized in the social network. But what is the mechanism by which the app is adopted by friends? Is homophily driving both friendships and adoption of specific apps based on interests? Or is the primary mechanism one of exposure, where having even just a single friend who has installed the app now gives an opportunity to become familiar with the app and subsequently adopt it?

To answer these questions, we observe the Facebook friendship graph at time t. For each app, we consider everyone on Facebook who has friends using the app, but who has not used the app themselves by t. Are there features of the individual and their friends that will predict whether the individual will adopt the app at some point in the future?

One-node neighborhoods

We begin with a question about homophily and its relation to app usage. Consider a Facebook user A who does not currently use the app, and suppose that has exactly one friend B who uses the app (that is, A has between 1 and 5000 friends on Facebook, but for our purposes here, exactly one of those friends is an app user). We choose some attribute on users (for example nationality, or age); we let $f(A)$ and $f(B)$ denote the value of this attribute for A and B, and we let f^* denote the modal (or median) value of the attribute across all app users.

Now the following question arises. Suppose that A is different from the typical user in this attribute, in the sense that $f(A) \neq f^*$. Is A more likely to adopt the app if the friend B is similar to A, or if B is similar to the typical app user? This is a basic question about the role of individual similarity in adoption decisions — if we're studying potential users who are outside the target demographic, is it more effective for their app-using friends to be similar to them, or similar to the target demographic?

We study this for nationality as an attribute in Figure 5.2 — given an app, we index nationalities so that the most common nationality among users of the app is labeled

0, and other nationalities are labeled by values $i > 0$. We now say that $a(i, j)$ is the adoption probability of a user A who has one friend B using the app, with $f(A) = i$ and $f(B) = j$. (Note that $f^* = 0$ according to our notation, since 0 is the most prevalent nationality in the app.) The figure shows that for a considerable majority of apps, we have $a(i,i)/a(i,0) > 1$, indicating a clear aggregate tendency for the question in the previous paragraph: a user A is more likely to adopt the app in general when A's one friend using the app is similar to A, not to the typical app user. In contrast, when $f(A) = f^*$, the ratio $a(0,0)/a(0,i)$ is balanced around 1, so there is no clear tendency in adoption probabilities between the case $f(B) = f^*$ and $f(B) \neq f^*$: for users who have the modal attribute value, the attribute value of their friend does not have a comparably strong effect.

This style of question gives us a way of analyzing individual attributes in general, and we find that attributes differ in the way this effect manifests itself. For example, when we consider age as an attribute (in place of nationality), we see (Figure 5.3) that if A has a friend B using the app, the age of B has very little correlation with A's probability of adopting. However, the age of A is related to the adoption probability — users A who are much older or younger from the median age are relatively more likely to adopt the app if they have a friend who uses the app, compared to individuals who are near the median age themselves.

Two- and three-node neighborhoods

We can also use the population of apps, and the adoption decisions that people make about them, to address a recurring recent question in the literature on on-line diffusion. Given an individual A who is not currently using an app, but who has k friends $B_1, B_2, \ldots, B_k$ using it, how does A's adoption probability depend on the pattern of connections among these k friends? Is A more likely to adopt if there are many links among these friends, or very few?

Past work has suggested that the answer can depend on the adoption decision being studied. Consider the results observed by Backstrom et al. [10] with LiveJournal

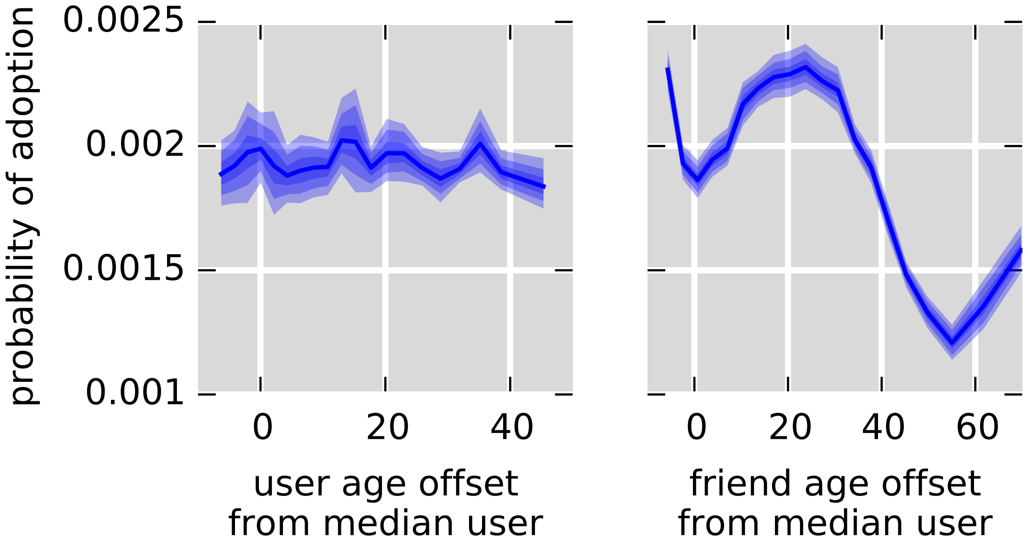


Figure 5.3: The probability that a user adopts the app given that they have one friend using the app. as a function of (left) the friend's age offset from the median and (right) the user's age offset from the median. The left plot indicates no apparent relationship between the age of the friend and that the user adopts. In contrast, the right plot illustrates that young users and users who are aged between 10 and 30 years above the median age are more likely to adopt. Users who are more than 40 years older than the median age are less likely to adopt. The probabilities were binned by age into 20 equally populated bins and the reported adoption probabilities are bootstrap estimates. The thick central line is the median bootstrap estimate of the mean, while the three outer bands indicate the 68%, 95%, and 99.7% confidence-intervals.

data, where conversion probability increases with the connectedness of one's friends, and contrast them with those observed by Ugander et al. [115] with Facebook e-mail invitations, where, for a fixed neighborhood size, one's probability of conversion strictly increased with the number of independent components. In both cases the result is non-obvious, as there is no a priori mathematical reason that the effect should be monotone with connectedness of one's neighbors.

Given the diverse answers arising in prior work, and the consequent suggestion that the result depends on the adoption decision, we consider how adoption probabilities vary with neighbor connectivity across a large sample of the most popular apps.

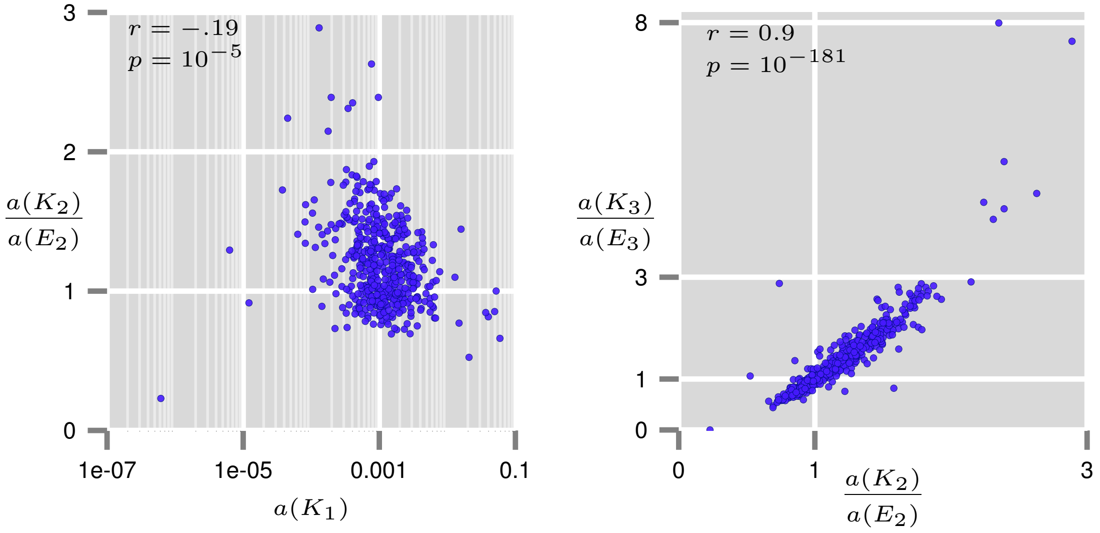


Figure 5.4: Right: the baseline rate of adoption given that a user has two friends using (horizontal) and the ratio of probability of adoption given friends are connected to probability given that friends are not connected (vertical). Apps above the line $y = 1$ exhibit the same trend as LiveJournal adoptions, and those below follow the trend observed in Facebook adoptions. **Left:** Closed to open conversion ratio for two-node neighborhoods (horizontal) and three node neighborhoods (vertical). Apps in the upper right quadrant follow the LiveJournal trend for two- and three-node neighborhoods, while apps in the lower left follow the Facebook trend. The correlation between these rates is 0.98 with $p << 0.01$, and there is a stark deficiency of apps in the diagonal quadrants.

We begin by considering the question just for two-node user neighborhoods, asking it separately for six hundred apps from

APPS$_{POP}$: given that a non-user A has exactly two friends using the app at time t, how does A's adoption probability depend on the presence or absence of an edge between A's two friends? Note again that these users may have any number of Facebook friends between 2 and 5000, but that only two of those friends can be app users. We explore this question in Figure 5.4. We find that both possibilities — higher or lower adoption probability with the presence of an edge — occur in roughly comparable proportions across the population of apps, suggesting that at the level of two nodes, both possibilities are indeed prevalent.

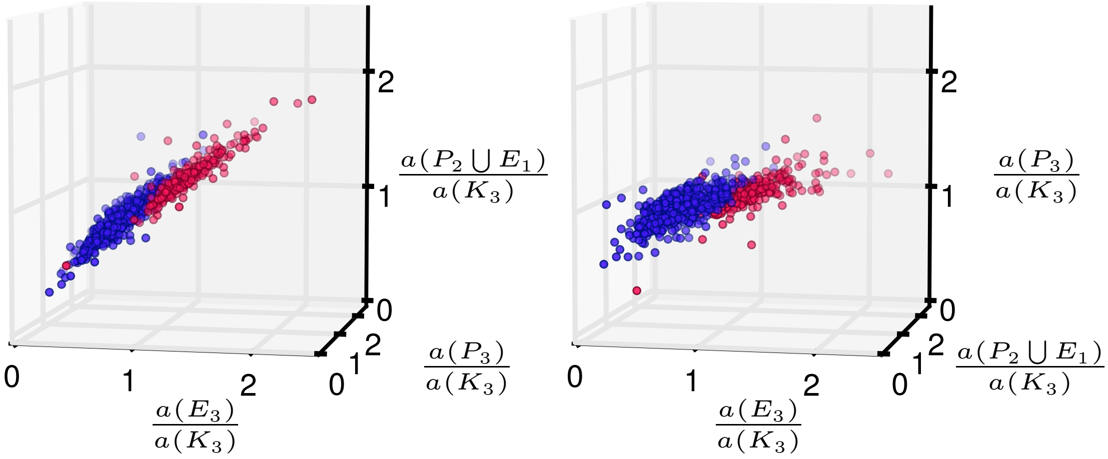


Figure 5.5: Two views of the same 3D point cloud: apps positioned according to the ratios $a(E_3)/a(K_3)$, $a(P_2 \bigcup K_1)/a(K_3)$, and $a(P_3)/a(K_3)$, and colored such that blue apps have $a(K_2)/a(E_2) \geq 1$, and red have $a(K_2)/a(E_2) < 1$. All adoption rates are reported relative to the rates for when a friend's user friends are a clique. Red apps have higher adoption rates with lower connectivity in the two-node graphs, and we see a near perfect correspondence in this trend for each possible combination of connectivity in three node graphs; this is demonstrated by the blue and red points falling naturally on either side of 1 in all three dimensions. In the left hand view the vertical extent of the cloud demonstrates the natural variation in relative adoption rates when a friend's user friends form two components (y-axis) compared to three (x-axis). In contrast, in the right hand view of this point cloud we observe more limited variation in relative adoption rates when the user friends are connected in one component. These differences are reflected in the Pearson correlation coefficients of 0.94 between $a(E_3)/a(K_3)$ and $a(P_2 \bigcup K_1)/a(K_3)$, and 0.57 between $a(E_3)/a(K_3)$ and $a(P_3)/a(K_3)$ ($p < 10^{-10}$ in both cases).

As often happens in the analysis of phenomena on small social subgraphs, we begin to see some rich structure emerge in the question when we move to three-node neighborhoods. For a graph G, we let $a(G)$ denote the adoption probability of user A when A's neighbors induce the graph G, and note that on three nodes there are four possible graphs: the complete graph K_3, the three-node path P_3, the single-edge graph $P_2 \cup E_1$, and the empty graph E_3.

We find (Figure 5.4) that the ratio $a(K_3)/a(E_3)$ covaries closely with $a(K_2)/a(E_2)$ across the set of apps — in other words, when the adoption probability of an app is

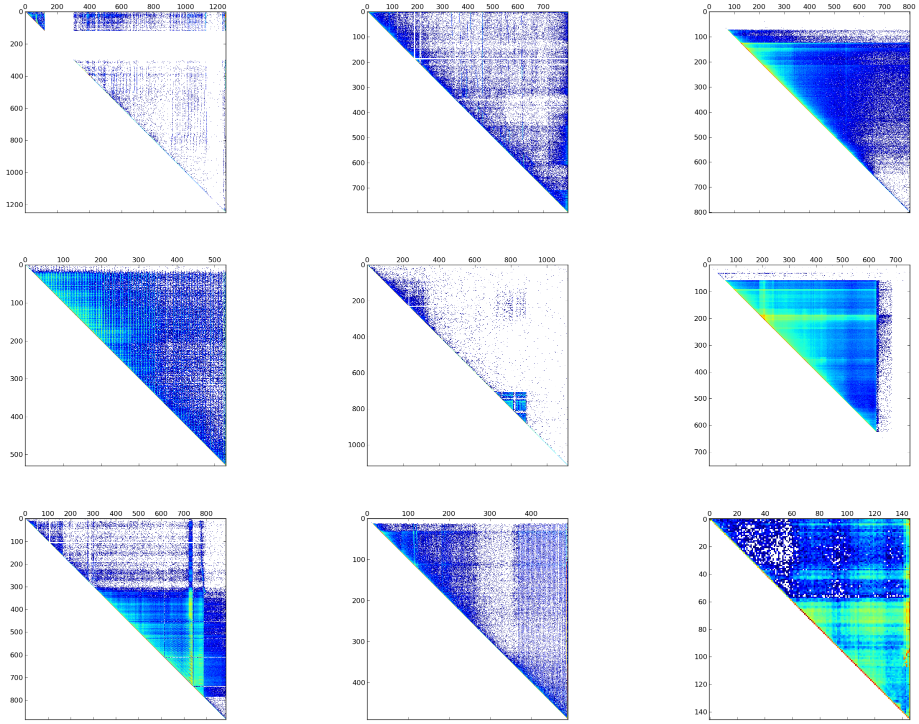


Figure 5.6: App adoption and departure dynamics. Heat maps of aggregate first and last login times for users of several examplar apps. The y-axis corresponds to the date of the first login and the x-axis to the last. The concentration is from blue (few) to yellow (many). Bright yellow and green horizontal or vertical bands correspond to periods of rapid adoption and departure, respectively. The color scale increases in density from white to blue, then yellow, then red.

higher for a connected pair of friends, it is also higher for a connected triplet of friends. This indicates how properties of two-node neighborhoods provide strong information about the properties of larger neighborhoods — and it is an empirical regularity of the adoption decisions rather than a strictly mathematical one, since the property for three nodes does not follow from the property on two nodes. We find similar regularities in the fact (Figure 5.5) that when adoption probabilities are higher for $a(E_3)$ relative to $a(K_3)$, they are also higher for the next-sparsest graph $a(P_2 \cup E_1)$. And we see comparatively much less variation between $a(P_3)$ and $a(K_3)$, which is consistent in interesting ways with the finding of Ugander et al.[115] that neighborhood topologies inducing the same number of connected components tended to lead to similar adoption probabilities.

5.4 Temporal patterns in apps

In addition to learning how the adoption of apps depends on friendship ties, we'd like to characterize the app's ability to retain those users who have adopted. These temporal features of an app's evolution hold some of the keys to its success.

To get a sense for what the retention of users looks like at a global level, we show the evolution of usage for a sample of apps in Figure 5.6. The images in this figure are heat maps in which the cell in the (i,j) entry records the density of users who first used the app at time i and last used it at time j. These maps thus show periods of heavy recruitment as horizontal bands — days when many individuals first started using the app. While there are a few vertical bands, denoting a narrow period of time when many users were last active, there is a clear concentration along the diagonal of many users departing soon after their first login, while some, located off-diagonal, remain active much longer. There are also sudden increases and decreases in density, as apps became more or less popular. With this type of global view of the diversity in usage and retention patterns, we next turn to modeling these retention patterns; our goal is to extract parameters from such a model to use as features in predicting an app's future engagement.

5.4.1 Retention model

For an app to have long term success we expect that it needs to maintain a relatively high level of user retention. We would like to have a model of retention that characterizes not only whether an individual will log into the app the very next day, but for any day subsequent to their first login.

Past work on retention modeling (e.g. [33, 60, 129]) has been focused on a particular product/activity (mostly online games), trying to predict users' continous engagement with a wide selection of features, many of them are domain-spefic and computationally expensive. To study thousands of apps and billions of users, we want to propose a model that is easy to compute and highly generalizable.

Simplest model: exponential decay

We start with the population of newly installed users, $n(0)$, and assume that at every time step each user has a constant probability of leaving, x_0. This mechanism gives rise to exponential decay:

$$\frac{dn(t)}{dt} = -x_0 n(t) \to n(t) = n(0)\exp(-x_0 t), \tag{5.1}$$

where $n(t)$ is the number of app users at time t. It turns out that this model does not yield a good fit to the data. However, the fit improves if we introduce a second parameter to the model by fitting from the second day; that is, we fit both for the decay rate, x_0, and the fraction of users that returned on day 2. With this relaxation from fitting day 1, the model becomes

$$n(t) = An(0)\exp(-x_0 t). \tag{5.2}$$

It is interesting that the exponential decay model fits the day 2 and onward trend well while not fitting day 1: it is reasonable to expect that there is a discontinuous transition in the probability of returning given an install versus an install and a return the following day. Day 1 users are dominated by those who use the app exactly once, whereas all the other days contain a signal from users who exhibited at least some level of continued interest in the app. Despite its ability to capture this distinction, this model is unsatisfactory: it entirely ignores the day 1 users, and even in the two-parameter version it under-predicts retention at long times (Figure 5.7).

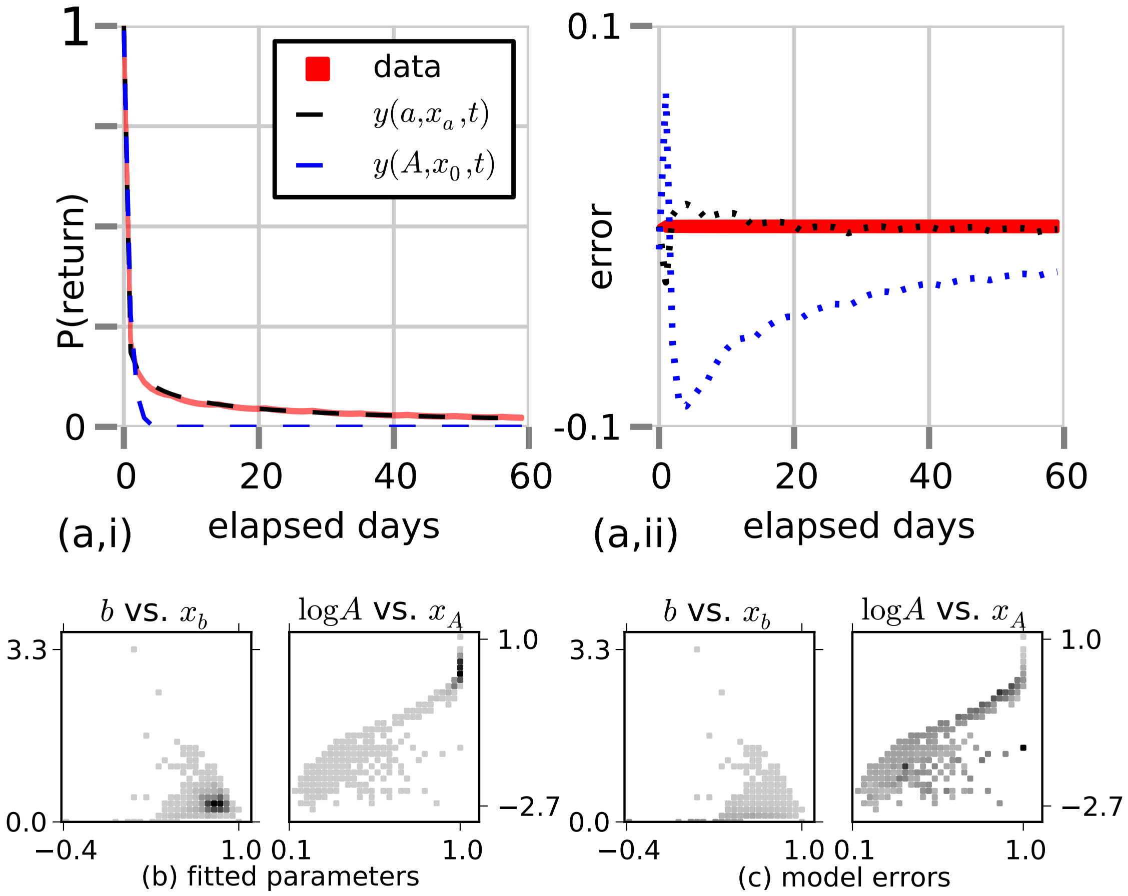


Figure 5.7: Empirical retention data, model predictions, and parameters. **(a,i):** Retention data and model predictions for an exemplar app. Error bars on the data (red solid curve) representing 99.999% confidence intervals ($4.4172\sqrt{p(1-p)/n}$). **(a,ii):** Error corresponding to fit shown in (a,i). **(b):** Distribution of fitted retention model parameters for the apps in APPS$_{POP}$. The shade represents the frequency of fitted parameter values falling into the given bin (darker being more frequent). **(c):** Mean error achieved by the model for apps with fitted parameters in the corresponding bin range. The same linear shade scale is used for both panels, with the lightest gray being 10^{-5} and black being 3 (white corresponds to no data). b and x_b denote the parameters in the time dependent model, and A and x_A those in the time indepdendent version.

Introducing simple time dependence

Instead of assuming that people have a constant probability of leaving at every time step, let us assume that their probability is a simple function of time:

$$\frac{dn(t)}{dt} = -\frac{x_a}{t^a} n(t) \to n(t) = n(0)\exp\left(-\frac{x_a t^{1-a}}{1-a}\right). \tag{5.3}$$

This model allows for the possibility that their likelihood of returning to the app could have a time dependent component, and it introduces this time dependence with the addition of only one parameter. Notice that by setting $a = 0$ we recover the traditional exponential decay model.

The parameters in this model have an interesting interpretation. Smaller values of a indicate that the app users have more momentum, that is, the app has more sticking power. The parameter x_a is still related to the familiar probability of depature: small x_a indicates that users are more likely to continue using the app.

5.4.2 Temporal analysis

After studying the temporal dynamics of individual apps through our retention model, we now look for regularities in the temporal patterns across multiple apps. We start this analysis by taking a random sample of all apps, and clustering their time series of daily active users using a k-means algorithm. All the time series are normalized by the number of active users on the app's peak day.

By varying k, we can get different sets of temporal clusters (see Figure 5.8). The two clusters generated for $k = 2$ already capture two dominant temporal trajectories of apps: one exhibits a clear rise and fall, while the other exhibits a slow but more sustainable rise. Note that the slow rise shown in the second cluster here may be misleading: as these apps keep growing and we normalize the time series by total volume, apps with a bigger user base will appear to have fewer fluctuations and slow growth/drop. When k increases, other small temporal clusters emerge, but they are not significantly different

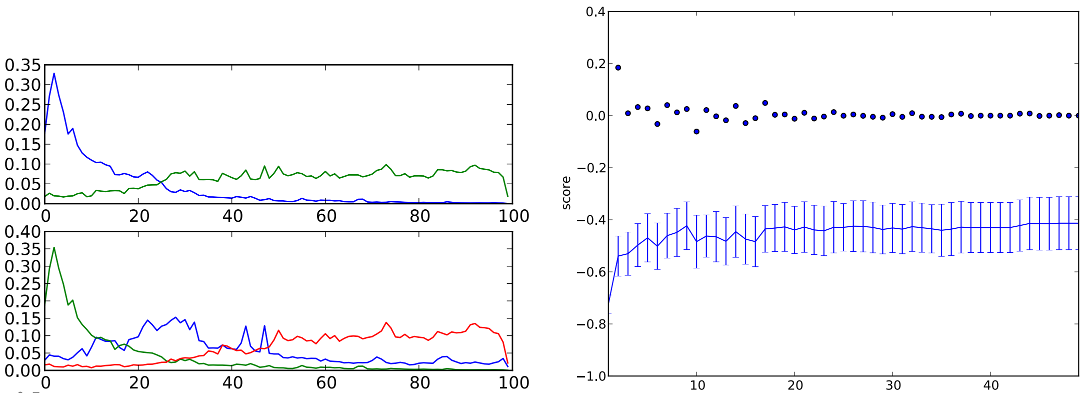


Figure 5.8: **Left:** k-means centroids of DAU for APPS$_{RAND}$, for $k = 2$ and $k = 3$. The clustering was done on the peak normalized time series of a 100-day observation window. **Right:** K-means scores (mean of distances from nearest centroid) for various values of k. Error bars represent 95% confidence intervals. Scatter plot is derivative of scores. Notice that no statistically significant improvement is gained for $k > 2$; that is, for all $k > 2$ the score of the resulting clustering is statistically significant different from that for $k = 2$. The scores were evaluated using a 75-25 train-test split, the clusters were generated with 100 restarts, and $L2$ was used as the distance metric.

from the two typical ones.

Then, for all apps that existed on June 1 2013, we compute their monthly active users (MAU) on that day and one year out. We plot those distributions against each other in Figure 5.9, where the color indicates the number of apps with the stated pairing of MAUs. The right-hand figure normalizes the columns, so that each bin column can be interpreted as the probability of ending up at the indicated MAU, given the current position. We can see that apps are most likely to stay at approximately the same MAU, but that, especially for very popular apps, there is a subpopulation that loses almost all of their users. This pattern suggests a natural underlying binary prediction problem: given that an app enjoys current success will it continue to be as popular one year later?

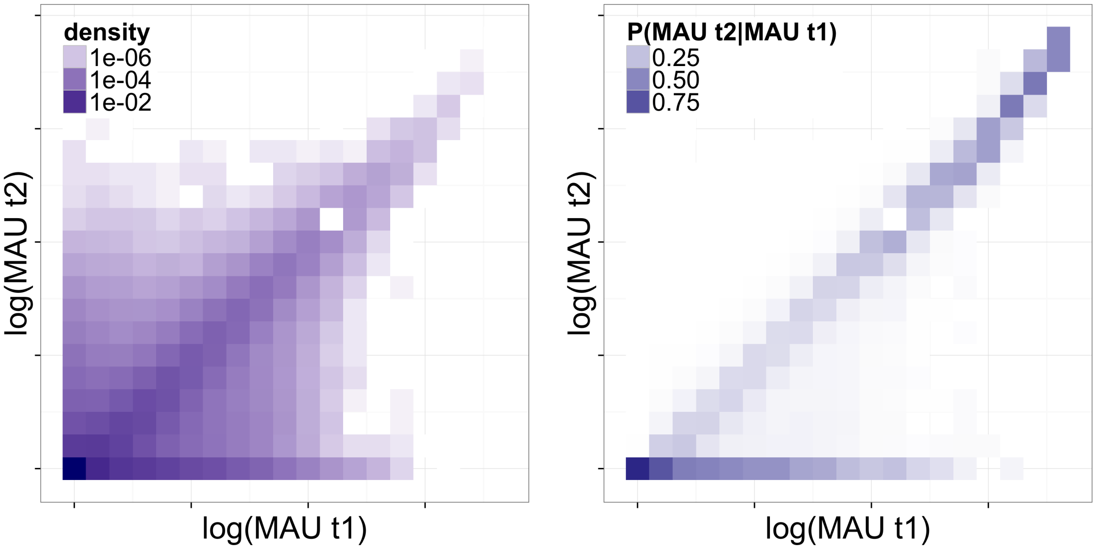


Figure 5.9: MAU of all apps in June 2013 (horizontal) and June 2014 (vertical). **Left:** Number of apps with MAU@t_1 and MAU@t_2 corresponding to the specified bins. **Right:** P(MAU@t_2— MAU@t_1) is the empirical probability of an app having y users at t_2 given that it had x users at t_1. At t_2 we observe that apps tend to either continue at the same level of popularity as they experienced at t_1 (bright diagonal) or exhibit a dramatic decrease in popularity (bright band on horizontal axis). Apps that are more populare have greater rates of continued success. However, when their popularity drops, the collapse tends to be complete.

5.5 Predicting app success

In the previous sections we have seen that apps can be described in a variety of ways. We began by exploring the relationship between an individual's social network and their likelihood of adopting an app, and in general how app usage is clustered in the social network. We also related a user's likelihood of adopting an app to their individual characteristics relative to those of the current app's users. Next we observed that, though overall patterns of adoption can be quite complex, an app's retention properties are well described by a simple model with a small set of parameters. And finally we again saw that while the fine grained activity level for any app is complicated, in the long term apps tend to either continue at the same activity level or diminish in popularity.

In each analysis we considered either hundreds or thousands of popular apps from this ecosystem, and saw that these various low dimensional features had interesting and diverse distributions across the population.

This brings us to our final set of questions: can we use an app's social, demographic, retention, and temporal features to predict whether or not it will be successful in the long term?

5.5.1 Predicting the longevity of apps

Note that we have seen empirically that the question of an app's long term success is well approximated by a binary variable (see Figures 5.8 and 5.9). In this subsection and the next, we will consider two variations on a binary prediction task. One task is straightforward: given a collection of promising apps, we want to predict which apps will have persistent success over the next year. The other task is based on a pairwise evaluation: to compare a pair of similarly popular apps and predict which one will be more successful in the future.

First, we consider the task of predicting which apps in the entire population will continue to be successful. Based on the number of active users on June 2014, we label an app as a positive example if it has over 50% of the number of active users it had in June 2013, and we label it as a negative example if has lost more than 50% of its users. This labeling turns out to provide us with a balanced class distribution, with the guess-all-positive baseline being 50%. For this binary classification task we built and evaluated the model by training random forests on apps in APPS_{POP}, where each app is represented as a vector of the features in Table 5.2.

The prediction performance results are shown in Table 5.3, and the use of all the available features leads to performance above 70% on this binary task. We find that the temporal features are the best single set of features, with the most important features being the median number of users in months 8 and 9 of the 12 month observation period (June 1 2012-June 1 2013 – see Table 6.1). The apps that would continue to be successful also had a higher weekly minimum; given that the overall popularity of the apps between classes is evenly distributed, we interpret this high weekly minimum as a

Temporal	
med /min/max $\text{DAU}_{mo.X}$	median, min, max number of daily users in month X of observation
med /min/max $\Delta\text{DAU}_{mo.X}$	median, min, max of change in daily users within month X ($\text{DAU}_X - \text{DAU}_{X-1}$)
med /min/max $\Delta^2\text{DAU}_{mo.X}$	median, min, max of second order change in daily users within month X
med /min/max DAU_{year}	median, min, max of DAU_X for $X \in 1, \ldots, 12$
med /min/max Δ DAU_{year}	median, min, max of ΔDAU_X for $X \in 1, \ldots, 12$
med /min/max Δ^2 DAU_{year}	median, min, max of $\Delta^2 DAU_X$ for $X \in 1, \ldots, 12$
Δ_{year} DAU	med DAU_{12} - med DAU_1
$\Delta_{year}\Delta$ DAU	med Δ DAU_{12} - med Δ^2 DAU_1
$\Delta_{year}\Delta^2$ DAU	med Δ^2 DAU_{12} - med Δ^2 DAU_1
*WAU, MAU, users, new users	Same statistics as listed for DAU above, considering instead weekly users, monthly users, total users, and new users
Demographic	
Country_X / $P(\text{Country}_X)$	Number, fraction of users from country X
Gender_X / $P(\text{Gender}_X)$	Number, fraction of users who stated their gender to be X
Age_X / $P(\text{Age}_X)$	Number, fraction of users who stated their age to be X
$l_{k,7}$ / $P(l_{k,7})$	Number, fraction of users who were active on Facebook for k out of 7 days
is30 / isnot30 / $P(is30)$	Number of users who are / aren't monthly active Facebook users; fraction of users who were monthly active Facebook users
Entropy(Country)	Entropy of country user distribution: $-\sum_{X \in \text{Countries}} P(\text{Country}_X) \log_2 P(\text{Country}_X)$
Entropy(Gender)	Entropy of gender user distribution: $-\sum_{X \in \text{Genders}} P(\text{Gender}_X) \log_2 P(\text{Gender}_X)$
Entropy(Age)	Entropy of age user probability distribution: $-\sum_{X \in \text{Ages}} P(\text{Age}_X) \log_2 P(\text{Age}_X)$
Entropy(l_7)	Entropy of l_7 distribution: $-\sum_{k \in 1,\ldots,7} P(l_{k,7}) \log_2 P(l_{k,7})$
Entropy(is30)	Entropy of is30 distribution: $-[P(is30)\log_2 P(is30) + (1 - P(is30))\log_2(1 - P(is30))]$
Retention	
N(t)	Number of users who returned t days after their first login
P(t)	Empirical probability of a user returning t days after their first login
a, x_a	Parameters for best fits of time dependent model: $N(t) = N(0)exp\left[\frac{-x_a t^{1-a}}{1-a}\right]$
A, x_0	Parameters Least squares parameter fits of time independent model:$N(t) = AN(0)exp[-x_0 t]$
Social	
med / max deg	median and maximum number of friends of an app user
med / max using	median and maximum number of friends of an app user who also use the app
$p(x \mid y)$	sociality: empirical probability of having adopted an app given that a friend has, i.e. mean fraction of an app user's friends who also use the app
$p(x \mid y)/p(x)$	relative change in probability of a user adopting an app given that their friend has
SIRS model	
S_0	susceptible population size, i.e. number of Facebook users who are interested at this app
α	probability of a non-user adopting the app due to non-social reasons
β	probability of a non-user adopting the app through social process
γ	probability of active user becoming inactive
ε	probability of in-active user being drawn back by active users
$pred(day_k)$	the DAU prediction at day k for k between 2013-06-01 and 2014-06-01

Table 5.2: Features used for training and testing the binary app success prediction tasks. Features were measured for all apps in APPS_{POP} (see Table 6.1), with the exception of the SIRS model features due to issues of model convergence.

signal of stability, and that this stability was a positive predictor.

Individual user attributes yielded the second highest performance, with the most important class features being activity-based ones: $l_{5,7}$ and $l_{6,7}$ ($l_{k,7}$ is the fraction of app users that were also active Facebook users for past k out of 7 days). We observe that for $k = 0, \ldots, 6$, negative examples are correlated with greater values of $l_{k,7}$, whereas for $l_{7,7}$, the trend reverses, and the positive examples with more users who are active on Facebook every day. This means that having users who were also highly active Facebook users is a positive indicator of success.

Among all the retention features, the most important one was the fitted parameter x_A, which represents the "departure probability" in the exponential decay model of users leaving an app. Not surprisingly, we find that the positive examples tend to have lower x_A than negative examples, indicating that having users who continue using the app for an extended period of time (i.e. a lower leaving probability) is correlated with the app's long-term success.

Finally, the most important structural features were sociality, i.e., average user degree, and mean/max number of friends who used the app. For the latter two we could not notice any significant differences between the two classes, but we do notice that high sociality is a negative indicator of success. This is likely due to the fact that we normalize the sociality measure ($p(x|y)$) by the popularity of the app ($p(x)$); thus those apps with very high sociality score are relatively small, and tend to be the ones situated in a very specific, niche market. Indeed, we find that if we consider the separate distributions of the numerator and denominator, we observe that $p(x|y)$ is indistinguishable for the two classes, while $p(x)$ is a positive indicator, leaving $p(x|y)/p(x)$ as a negative indicator.

SIRS model

In general the task of predicting an app's time-series trajectory is a rich and interesting problem, but the binary nature of trajectories that we observed motivated our simplification to the binary prediction task. To explore the potential inherent in modeling richer properties of the time series, we also consider a model of app usage via a set of interacting reaction diffusion processes, much like a chemical reaction. The model we use was proposed by Ribeiro et al. [98], and falls into the well-known class of SIRS models. We will briefly describe how we implemented this model, and when we return to our underlying prediction task, we will consider the predictions and parameters from this model as an additional set of features.

Ribeiro et al. [98] proposed a model describing the dynamics of a webservice of daily activity time series, derived from the classical epidemic model and comprised of a set of reaction diffusion processes. The model is specified by a set of parameters, including the estimate of the susceptible population, and the transition probabilities between different states. Ribeiro also outlines a framework for fitting these parameters given a window of time series activity levels, and then uses them to extrapolate and make a long term prediction of future activity levels. We implemented a model very similar to the one described in [98]. We fitted the model using a Monte Carlo process using time series from June 2, 2012 to May 25, 2013 (the same period from which we extract temporal features), and used the fitted model to generate predictions between May 26, 2013 and May 15, 2014.

There are two things we note about the SIRS model. First, as we try to predict the future of apps from a fixed time point, the apps we are studying can be in very different life stages. For example, some apps in our dataset had only existed for a short period of time by the observation day, and thus have very limited time series data to compute a good fit of the SIRS model. Second, some underlying assumptions in the SIRS model, such as the constant rate of user adoption through advertisement or word-of-mouth process, may not hold in reality. As a result, the model would not converge for certain apps, especially the ones that experienced large fluctuations in their lifecycles.

Nevertheless, we were able to fit over two-thirds of the apps in APPS$_{POP}$. Among them, 1100 apps had reasonable convergence and error estimates. We then used both the fitted parameters and the predicted time series as our features for this subset of 1100 apps. On that subset of 1100 apps, the relative performance of the other features sets was the same (all combined features yield the highest performance, followed by temporal, then demographic, retention, and finally social).

We find that the features from the SIRS model perform worse than the retention features but better than the social features. Thus, despite the richness of the time-series modeling made possible by the SIR framework, as a feature set it does not perform as well as other measures incorporating temporal properties, including the retention model from the previous section.

Feature set	*accuracy*	*prec:* *+;-*	*recall:* *+;-*	*top 2 features:* $\{$*among all*$\}$*;* $\{$*within class*$\}$
Baseline	0.50			
All	0.73	0.72; 0.74	0.74; 0.72	med users$_8$ med users$_9$; –
Temporal	0.71	0.72; 0.7	0.68; 0.74	$\Delta_{year}WAU$, $\min WAU_{11}$; med users$_8$ med users$_9$
Demographic	0.66	0.64; 0.68	0.70; 0.61	$l_{6,7}$, $l_{5,7}$; $l_{6,7}$, $l_{5,7}$
Retention	0.61	0.59; 0.64	0.70; 0.53	x_a, x_A; day 2 and 3 returns
Social	0.6	0.59; 0.61	0.60; 0.59	$\frac{p(x\|y)}{p(x)}$, $\langle$user degree$\rangle$; Mean and max # of friends using the app

Table 5.3: Prediction performance results for five combinations of features. Precision and recall: top and bottom rows are for positive and negative classes, respectively. Features are ranked by out-of-bag importance estimates while training the random forests. We trained the classifier using all the features, and report the most important ones in each category in the top row ("among all"), and train the classifier with only the features in each category, and report the top opens in lower row ("within class").

5.5.2 Predicting pairwise relative success

Next we formulate a separate but related prediction task, by constructing a pairwise comparison version of predicting app success. Given that two apps have approximately the same monthly active users at t_1 (MAU@t_1), and by t_2 they had diverged from each other, we want to predict at time t_1 which app is going to be more successful. We evaluate this problem with a variety of thresholds for what we considered "near-" and "long-"term predictions of MAU. This prediction task is particularly useful when investigating a set of competitive apps in the same market. Intuitively, it is difficult to tell similar apps apart at an early stage [99]. However, by looking at pairs whose outcomes at t_2 are successively farther apart, we can control for the difficulty of the task and understand when it becomes feasible to predict such divergence.

For the pairwise prediction task we begin by generating a 50-50 train-test split between apps, and represent each pair of apps as a concatenation of two feature vectors, again using the features from Table 5.2. We then introduce a subtle variation to make the setup more relevant to a real-world scenario. The features and labels used in the training stage are generated using snapshots of our datasets at t_0 and $t_1 = t_0 + 6$ months, while those used for testing are generated using snapshots at t_1 and $t_2 = t_1 + 6$ months. This simulates the practical scenario of observing the app population at t_1, learning which characteristics of apps lead to their success, and using the learned knowledge to predict the future.

Two apps are considered to start off as being "comparable" if they fall into the same decile at $t_{0/1}$ (train/test), and are considered "distinct" if they are at least k deciles apart at $t_{1/2}$ (train/test). In Figure 5.10 we see that prediction accuracy increases monotonically with k, and that the best set of features (temporal) ultimately yield 75% prediction accuracy. The other most striking feature of Figure 5.10 is that for most of the threshold window, all the features yield approximately the same performance. Each set of features, besides demographic, takes a turn at being both the top performer and the lowest. The individual feature analysis that we did was consistent with the observation that this task is not highly sensitive to the choice of features. To analyze which features could best discriminate between positive and negative examples we used the two-sided Kolmogorov-Smirnov test to compare the distributions of each feature for positive and negative examples. We find that, with the exception of a few underpopulated demographic features, the Kolmogorov-Smirnov test finds that each feature is distinguishable between the negative and positive examples with p-values extremely close to zero.

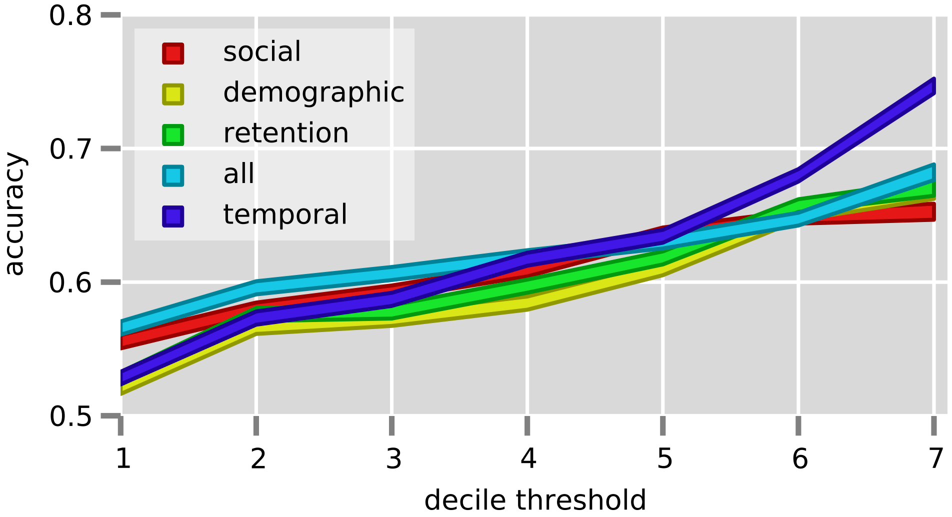


Figure 5.10: Prediction accuracy for the pairwise relative success prediction task, as a function of decile threshold, *k*.

5.6 Related Work

Sociologists and economists have long studied the problem of product adoption and retention. Early work in this domain focused on the diffusion of innovations, as people proposed a series of mathematical models to describes the adoption of new products by consumers, such as the "S-shaped" adoption curve [32] and the Bass model [15]. These models have been successful in predicting the impact of advertising, especially the effect of advertising through mass-media and billboards. Other work has focused on the diffusion of innovations and products through social ties [101]. With the rise of social media and online social networks, there has been more and more evidence that the social influence, i.e. the word-of-mouth effect, is playing a increasingly important role at driving the adoption of products and services [13, 75, 85].

To understand how products and information spread in social networks, most existing work tries to predict the volume of popularity, such as the the size of online communities [10], the number of fans of Facebook pages [111], and the usage of hashtags on Twitter [12, 102, 120]. While these work showed the correlation between the scale of diffusion and its structural and topical properties, there has been a recent line of work questioning the predictability of large viral events [12, 103]. In response, Cheng et al. [18] showed that it is possible to predict how much more a cascade will grow by observing the temporal and other features of its spread up to the present time.

Besides being a key predictor for cascade size, the temporal dynamics of cascades have been an interesting research topic [25, 76, 127]. Upon the discovery of several robust temporal classes of cascades on different platforms, most studies on the temporal dynamics focused on bursty events [117], or the peak volume [25, 76]. Indeed, the majority of popular things spread on-line enjoy very short attention span: the popularity rises and drop quickly, usually within a few hours or a day [124, 127]. The persistence of interest, although rare, is rather intriguing. Wu et al. [124] found that the longevity of URLs on Twitter can be explained by the intrinsic cultural value of the content they link to. Follow-up work showed that information with positive sentiment is more likely to persist [125]. Ducheneaut et al.. discovered that smaller and denser guilds in World of Warcraft are more likely to survive longer [35].

While many papers correlate the temporal patterns of cascades with its empirical properties, some researchers have developed theoretical models on individuals' choice of adopting and engaging with a product or activity [98, 117]. These models are useful at depicting the mechanism behind the observed temporal dynamics, however, it is unclear how generalizable they are beyond the particular product or activity studied.

Our work contributes to current research in two major aspects.

First, we study the entire lifecycle of apps over a timespan as long as 5 years . Our focus is the persistency of growth other than the peak popularity. Different from a viral YouTube video or a meme photo, successful apps needs to engage with their users repeatedly. Therefore, we spent a significant amount of work analyzing and modeling the retention of apps, and showed its importance to the long-term success of apps.

Second, we study thousands of apps at once. Previously, most papers examined the adoption and retention of a single product/activity, thus their results might not be generalizable to other domains. By studying a large selection of apps on Facebook, we are able to control for app-specific features and understand how the characteristics of an app interact with its social and temporal dynamics.

Some work similar to ours includes a recent study of the growth and longevity of online communities [58], the modeling and prediction of the temporal pattern of membership-based websites [98], and a study of mobile app adoption over a small real-life social network [93]. Our study builds on these papers in both the scope and the variety of examples examined. Also, with the rich dataset we have about apps, users, and the underlying social graph, we are able to introduce several new theoretical and analytical models, and to compare them with recent formalisms [98]. By incorporating the parameters of a fitted model as part of the feature set, we are able to extend and compare different methodologies.

5.7 Conclusion

In this paper we studied the lifecycle of apps: as they grow and thrive, and, in some cases, as they decline. We studied differences in their development, looking for clues to their future fate. First, we sought parameters with which to model the interaction between the app and the individual. We found that a simple exponential decay, even with an adjustment for attrition after the first day of use, did not accurately capture user retention. Instead, those who keep using the app over a longer time period are less and less likely to stop. Modeling retention of individuals in this way is helpful in predicting app success.

Another dimension goes beyond the individual to whether the app is adopted socially. Apps vary widely in the sociality of their adoption, and we find heterogeneity in the apps based on how their adoption probabilities depend on the connectedness of friends who use the app and the similarities in attributes between an adopter and his or her friends.

The features most predictive of an app's future dynamics are those describing its past growth trajectory. More widely adopted apps that have recently been on a growth trajectory are more likely to persist. Given a range of features, we obtain over 20% absolute improvement over random guessing when it comes to making a binary prediction as to sustained activity for an app. We also obtain strong performance when we formulate the problem as one of matching two apps of roughly equal size which take different trajectories, and trying to distinguish the two with a much higher than random accuracy.

There are a number of further aspects of the app ecosystem that would be interesting to take into account in future work. First, app adoption is driven in part by the marketing and other recruitment strategies of the app owners. Although our models incorporate the numbers of new users coming to the app over time, they do not differentiate between organic growth and advertising-driven growth. Furthermore, it is not clear whether sociality of apps might accelerate growth or decline or both. Finally, it is unclear whether some features might be early harbingers of future behavior, e.g. whether the change in retention of long-time or recently acquired users is more useful in forecasting the eventual adoption of the app. We leave these and other questions for future work.

CHAPTER 6
INTERNET COLLABORATION ON EXTREMELY DIFFICULT PROBLEMS: RESEARCH VERSUS OLYMPIAD QUESTIONS ON THE POLYMATH SITE

This chapter is written in collaboration with Chenhao Tan, Jon Kleinberg, and Lillian Lee and is being published in Proceedings of the 25th International Conference on World Wide Web in 2016.

Despite the existence of highly successful Internet collaborations on complex projects, including open-source software, little is known about how Internet collaborations work for solving "extremely" difficult problems, such as open-ended research questions. We quantitatively investigate a series of efforts known as the *Polymath* projects, which tackle mathematical research problems through open online discussion. A key analytical insight is that we can contrast the polymath projects with *mini-polymaths* — spinoffs that were conducted in the same manner as the polymaths but aimed at addressing math Olympiad questions, which, while quite difficult, are known to be feasible.

Our comparative analysis shifts between three elements of the projects: the roles and relationships of the authors, the temporal dynamics of how the projects evolved, and the linguistic properties of the discussions themselves. We find interesting differences between the two domains through each of these analyses, and present these analyses as a template to facilitate comparison between Polymath and other domains for collaboration and communication. We also develop models that have strong performance in distinguishing research-level comments based on any of our groups of features. Finally, we examine whether comments representing research breakthroughs can be recognized more effectively based on their intrinsic features, or by the (re-)actions of others, and find good predictive power in linguistic features.

6.1 Introduction

Groups interacting on the Internet have produced a wide range of important collaborative products, including encyclopedias, annotated scientific datasets, and large pieces of open-source software. These successes led the Fields Medalist Timothy Gowers to ask whether a similar style of collaboration could be used to approach open research questions. In particular, his focus was on his own domain of expertise, mathematics, and in early 2009 [45] he famously asked, "Is massively collaborative mathematics possible?"

Shortly after posing this question, he and a group of colleagues set out to test the proposition by attempting it. They began the first in a series of so-called *Polymath projects*; in each Polymath project, an open, evolving group of mathematicians communicate via a shared blog attempt to solve an open research problem in mathematics. The groups have been quite diverse in background; they have included active participation from Gowers and a second Fields Medalist, Terence Tao, along with a large set of both professional and amateur mathematicians. To date there have been nine Polymath projects; three of them have led to published papers and one to notable partial results preceding the subsequent resolution of its central question, thus demonstrating that this approach can lead to new mathematical research contributions with some regularity.

The Polymath projects have an explicitly articulated set of guidelines that strongly encourage participants to share all of their ideas via online comments in very small increments as they happen, rather than thinking off-line and waiting to contribute a larger idea in a single chunk. We can thus see, through the comments made on the site during the project, almost all the ideas, experiments, mistakes, and coordination mechanisms that participants contributed.

Attempts to think about the nature of the collaboration underpinning Polymath lead naturally to analogies in several different directions. One analogy is to the online collaborations one finds in other settings, such as Wikipedia [64] and open-source software projects [126]. A second analogy is to large decentralized collaborations that take place in "traditional" scientific research [57].

But both analogies are limited. The first does not quite fit because our existing models of collaborative work on the Internet involve domains where the task is inherently "doable": the feasibility of the task — authoring an encyclopedia article or writing an open-source computer program to match a known specification — is not in doubt, and the primary challenge is to achieve the requisite level of scale and robustness. In Polymath, on the other hand, we see people who are the best in the world at what they do struggling with a task that might be beyond them or impossible as they work on open problems in their field.

The second analogy also does not quite fit: as noted by Gowers [45], decentralized scientific collaborations have typically focused on problems that are inherently decomposable into separate pieces. With Polymath, on the other hand, we see problems that present themselves initially as a unified whole, and any decomposition needs to arise from the collaboration itself. Anyone with Internet access can participate for any period of time that they wish.

For all these reasons, Polymath provides a glimpse into a novel kind of activity — the use of Internet collaboration to undertake world-class research — in a way that is not only open but completely chronicled. In the same way that co-authorship networks have provided a glimpse into the fine-grained structure of scientific partnerships [43, 52], the contents of Polymath offer a look at the minute-by-minute communication leading to the research that these partnerships enable.

With a growing number of sites where people congregate to discuss solutions to hard problems, it is useful to also appreciate the basic similarities between Polymath and other Web-based communication and collaboration platforms. Even if the specific findings about Polymath do not generalize to all other contexts, the questions themselves can often be generalized. With this in mind, an additional goal of the paper, beyond the investigation of Polymath as a domain, is to present a template for questions that we believe can be productively asked in general about the type of data that sites like Polymath generate. We hope that this template will help facilitate direct comparisons and contrasts with future studies of collaborative Web-based problem-solving.

6.1.1 Summary of contributions

Data from Polymath 1 was analyzed in an interesting paper by Cranshaw and Kittur [26]; in their own words, they provide “an in-depth descriptive analysis of data gathered from [Polymath 1],” focusing on the role of leadership in the progress of the project, and the interaction between established members and newcomers as the projects proceeded. With the inception of eight new Polymath projects, and rich variation in their evolution and success, a new set of opportunities arises in the type of questions we can explore with Polymath data. We organize our analysis around *two central questions* regarding Polymath.

(1) Research or hard problem-solving?

At a general level, our first question is to analyze some of the distinctions between online discussion about open research questions versus online discussion about tasks where the outcome is more attainable.

To address this question, and to make the comparison as sharp as possible, we use a source of discussion data that comes from Polymath itself: the *mini-polymath projects*. Shortly after Polymath was successfully underway, Terence Tao assembled a group to solve something hard but more manageable than a research question; each mini-polymath problem is a question from a past International Mathematical Olympiad (IMO). The existence of the mini-polymaths provides us with a very natural contrast between the two types of activities. Specifically, we can understand the differences between tackling an open-ended research problem, where current techniques may be completely inadequate for finding a solution, vs. solving a problem that, while difficult, is known to be feasible, in a setting where, to a large degree, there is control for topic (in both cases, difficult mathematics) and for participants (there are dozens of people who participated in *both* Polymath and the mini-polymaths). We study and contrast the polymath and mini-polymath projects with three lenses: the roles and relationships of the authors, the temporal evolution of the projects, and the linguistic properties of the comments.

Roles of authors and leadership. First, we analyze the role of the authors, the role of leadership, and differences in patterns of conversation networks in the two domains. In particular, in the research domain we observe that there is a substantially higher concentration of activity in the hands of fewer people, indicating that there was a more distinct notion of contribution leadership in the research domain than the somewhat easier mini-polymath domain. We further observe that there is significantly more symmetry in the global conversation network than what would be initially expected, which is not the case in the mini-polymath projects.

Temporal dynamics. Second, we consider how progress in the two domains evolved over time, and observe interesting patterns both in differences and similarities between the two domains. The two types of projects differ in the temporal properties of the discussion: overall, comments come more quickly in mini-polymath projects, befitting their smaller-scale format, but, interestingly and unexpectedly, on the shortest time scales comments actually come more quickly in Polymath, indicating that the research discussions have the potential to reach the most rapid-fire rate.

Linguistic properties. Third, we study the use of language in the two domains, in both content and high-level linguistic features such as politeness, relevance, and specificity, again finding interesting differences between the two domains. Strong signals in the text distinguish comments in Polymath projects from those in mini-polymath projects. At the most naive level, using bag-of-words classification achieves an accuracy above 90%, since problem-specific terms and time differences (as expressed by words such as "primes" or "July") can be prominent in these two kinds of discussions. But surprisingly, and more importantly, restricting attention to just words that are *not* topic-focused still achieves 90% accuracy, suggesting stylistic differences in Polymath comments and mini-polymath comments. Additionally, high-level linguistic features beyond just individual words display significant differences between the two domains: research discussions in Polymath projects have higher average word distinctiveness, higher relevance to the original post for the topic, greater politeness, and greater usage of the past tense.

(2) General contribution or research highlight?

Our second question is based on a key aspect of research collaborations — they pass through "milestones" when important progress is made. Can we characterize such milestones as the collaboration unfolds? With the ability to do this, one may be able to set up mechanisms that help researchers focus on promising directions, which can potentially result in more productive research collaboration. Alternatively, a more pessimistic hypothesis is that these milestones may only be realized in retrospect. To characterize these milestones, we formulate a prediction problem that asks whether it is possible to identify comments that were marked "highlights" by participants.

The task of identifying highlights turns out to be more challenging than our first task, distinguishing Polymath comments from mini-polymath ones. Nevertheless, we still obtain prediction performance significantly above the baselines for the task. To help understand whether the challenge is inherently in the task or in the shortcomings of our prediction algorithms, we compared to the performance of applied mathematics graduate students in recognizing highlights from Polymath discussions. Algorithms using the strongest feature sets achieve comparable performance to these human judges. We also find that features based on the individual comments themselves outperform features that try to capture reactions or the run-ups to the comments in question.

6.2 Data

The Polymath and mini-polymath projects share their common roots in a gateway wiki hosted by Michael Nielsen[1]. Starting from that site, we parsed all discussion comments, and for each comment retained its text, its author's WordPress username, its timestamp (with minute-level granularity), and its permalink.

[1] http://goo.gl/LVEWbe

For portions of our analysis we use all the Polymath projects, but in other parts we focus on the most active and successful. As Table 6.1 indicates, there is a relatively wide variation in the amount of content produced as part of each Polymath project, as well as variation in their levels of success. The mini-polymaths, on the other hand, are more uniform and each solved the Olympiad problem that they focused on. When comparing Polymath to mini-polymath, we often focus on the subset of Polymath projects whose successful outcomes are analogous to the successes of the mini-polymaths; these are the Polymaths that led to publications (Polymaths 1, 4, and 8) as well as Polymath 5 which was also highly active and led to important partial results on the Erdos Discrepancy Problem (EDP).[2] Unless otherwise stated when we refer to *quantitative* results or observations about the Polymath projects, we are referring to this subset.

In addition, we collected data about which comments in the Polymath 1 project were identified as research highlights, which was recorded on a subpage of the Polymath projects wiki page.[3]

The data studied in this paper has been made publicly available online at https://bitbucket.org/isabelmette/polymath-data.

6.3 Roles and Leadership

6.3.1 Leadership and inequality in research discussions

What is the role of the top contributors in the Polymath research setting compared to mini-polymath's simpler domain? Similarly, what role do authors who contribute less frequently play in the two settings? And how does the interaction structure of the authors vary across the projects? We find striking differences between the two domains;

[2] Polymath 5 was very active (see Table 6.1) and led to partial though unpublished results, which were then cited by Terence Tao when he published his resolution of the EDP in 2015 [113].

[3] `http://goo.gl/ijbIqP`

Table 6.1: Activity summary for each of the polymath and mini-polymath projects. Focal polymath projects of the present study are highlighted in blue, other polymath projects are shown in black, and mini-polymath projects in red. *Tag*: label used in subsequent figures. *Papers*: number of papers written by the corresponding project. *See Footnote 2 regarding partial results from Polymath 5. *Active days*: number of days on which at least one comment was made. The figure shows the number of comments and distinct authors in each project.

Project (tag)	*Papers*	*# of comments*	*Active days*
Polymath 1 (p1)	2	1509	112
Polymath 4 (p4)	1	573	103
Polymath 5 (p5)	0*	2757	238
Polymath 8 (p8)	2	3975	413
Polymath 2 (p2)	0	48	10
Polymath 3 (p3)	0	553	110
Polymath 6 (p6)	0	16	4
Polymath 7 (p7)	0	531	81
Polymath 9 (p9)	0	100	28
Mini 1 (m1)	n/a	336	15
Mini 2 (m2)	n/a	120	7
Mini 3 (m3)	n/a	146	16
Mini 4 (m4)	n/a	102	10

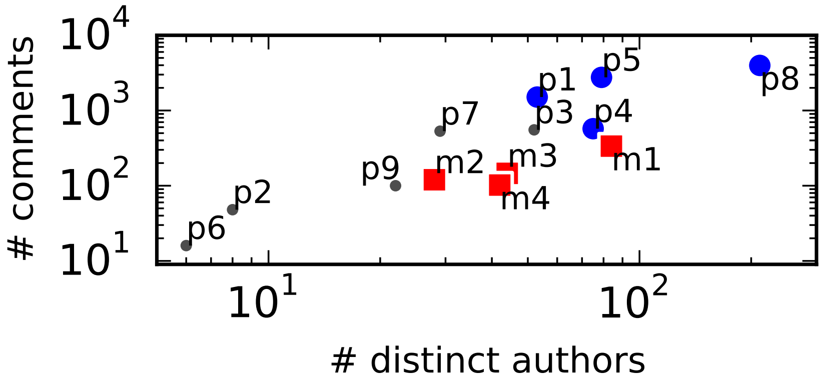


Table 6.2: Overview of leadership in the Polymath projects and mini-polymath projects.

Project	Host(s)	Top two contributors
Polymath 1	Tao, Gowers	Gowers, Tao
Polymath 4	Tao	Tao, Croot
Polymath 5	Gowers	Gowers, Edgington
Polymath 8	Tao, Morrison	Tao, Paldi
Mini 1	Tao	Bennet, Speyer
Mini 2	Tao	Bennet, Hill
Mini 3	Tao	Thomas H, Narayanan
Mini 4	Tao	Gagika, Olli

contrasts in the leadership structures are present by design, but the differences in the organic structure of participation stand out equally strongly.

There is an initial superficial difference between the Polymath and mini-polymath projects: in the Polymath projects, the leaders were also among the main contributors, while the mini-polymath projects were designed so that the leaders did not contribute extensively.[4] In a bit more detail, there is a clear definition of "the leadership" in the Polymath projects, as Tao and Gowers were both the project hosts (they collaboratively hosted Polymath 1 on their two blogs) and its two most prolific authors. Table 6.2 lists the hosts for each project alongside each project's two top contributors. In the Polymath projects the hosts are almost always among the top contributors.

Moving beyond this straightforward distinction between moderators and contributors, we explore to what extent contributions in the successful Polymath and mini-

[4] As Tao noted in setting up the mini-polymath projects, he hosted them (either via his own blog or as the moderator on the polymathprojects.org blog), but he refrained from contributing to the collaborative effort, stating, "I myself worked it out ... in order not to spoil the experiment, I would ask that those of you who have already found a solution not to give any hint of the solution here until after the collaborative effort has found its solution. ... I will not be participating in the project except as a moderator."

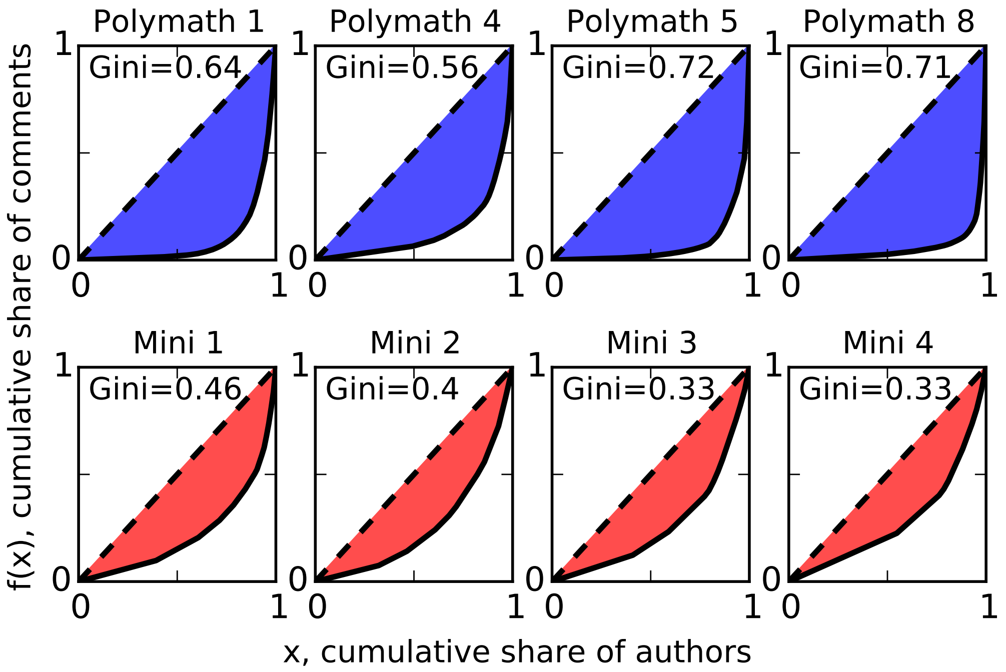


Figure 6.1: The Gini coefficient — the area between the solid and dashed lines — indicates that there is more equality in the mini-polymath author-comment distributions than in Polymath's. The vertical axis $f(x)$ is the cumulative fraction of comments that have been contributed by the corresponding cumulative share of authors x, where the authors are sorted by increasing number of comments written. Dashed line: $f(x)$ for a hypothetical uniform distribution. Solid line: observed distribution in the given project.

polymath projects were made by a small group of active authors versus shared across a larger group.

On one hand we have the hypothesis that the easier mini-polymath projects could more easily be dominated and solved by just a handful of people, while the more difficult projects would require contributions from a greater number of people. On the other hand, it may be that work in the mini-polymaths would be distributed more evenly among many people because their lower difficulty level made them accessible to a larger group, whereas in Polymath the problems are so difficult that very few people are able to make a substantial number of contributions.

We explore this question and find clear differences in the role of leadership and heterogeneity using the Gini coefficient, a well-known measure of a system's inequality, as shown in Figure 6.1. In this domain we apply the Gini coefficient to the fraction of authors who contribute a given fraction of the total number of comments in a system. The Gini coefficient is computed via the Lorenz curve, the fraction of comments $f(x)$ made by the x fraction of people who provided the least number of comments. Larger Gini coefficients indicate more inequality.

From Figure 6.1, we find that the mini-polymath projects possess a notably greater degree of commenting equality (a lower Gini coefficient) than the research projects. This means that in the research domain a larger fraction of comment contributions was made by a smaller fraction of authors. But while research discussions tend to be dominated by fewer people, do the less dominant people still make meaningful contributions? We find that the answer is yes. Recall from the introduction that a subset of the comments in Polymath 1 were labeled as "highlights" by participants. We can thus measure the Gini coefficient on two separate sub-populations defined by these labels: the highlights and the complement of the highlights. We find that the two sub-populations have nearly identical distributions, and thus to the extent that lower frequency contributors participated in Polymath 1, they were making contributions that were indeed classified as highlights in the overall success of the project.

6.3.2 Symmetry and Sticky Conversations

What does the sequence of participants in a conversation tell us about the domain? How does the reply structure of a conversation aimed at solving an extremely hard problem compare to the reply structure in an easier problem-solving domain? To investigate these questions, we pinpoint two closely related metrics: *reply symmetry* and *stickiness*.

Setup and baseline. Both metrics, reply symmetry and stickiness, are computed using the sequence of authors who comment on the project.

In particular, for each project we have the set of authors who comment, denoted $\mathscr{A} = \{a^i\}_{i=1}^n$, and the sequence S in which their m comments were made: $S = \{a_1^i, a_2^j, \dots\}$. The random baseline for these metrics will be based on a time-zone-controlled random sequence. That is, to create a random sequence S^{rand}, for position S_i^{rand}, we select a random author from the set of authors who have commented in that hour of the day, proportional to how frequently they have commented during that hour.

Definition: reply symmetry. To define reply symmetry we consider the reply matrix A: A_{ij} is the number of times author j follows author i in the sequence S. We then define symmetry in the matrix as $sym(A) = 1 - \frac{|A - A^T|_1}{|A|_1}$. The 1-norm of the matrix A, $|A|_1$, is the total number of comments, and $|A - A^T|_1$ is the number of alterations that would be made to the sequence of comments such that the reply matrix is completely symmetric.

This definition captures the extent to which people respond to the same people who respond to them, regardless of whether they respond immediately in real time, or at a later time.

Definition: stickiness. Next we define the notion of *stickiness*, which captures the local author symmetry in comment sequences. In the author sequence, we first count the number of times we observe the sequence motif *aba* — an author *a* is followed by another author *b*, who is then followed again by *a*. Similarly, the motif *abc* corresponds to comments by three distinct authors in succession, while the motif *aaa* corresponds to three comments in a row by the same author. We define *stickiness* of the interaction to be the extent to which the *aba* motif is overrepresented; it is the probability of observing the motif *aba* in the real sequence relative to the probability of observing it in a time-zone-controlled random baseline (the likelihood ratio).

Results: symmetry and stickiness in research domains. In Figure 6.2 we test the hypothesis that the amount of symmetry observed is as much as would be observed by a random, asymmetric graph. We find that in each case the bootstrapped $p-$value for the Polymath projects is less than 0.05, indicating that we can reject this hypothesis and that the symmetry we observe is more than one would expect from random fluctuations (the exceptions are Polymath projects 2 and 6, which both have fewer than 10 authors and 100 comments total, which is too little data to compute a meaningful estimate). On the other hand, for each of the mini-polymath projects the estimated $p-$value is above 0.05, indicating that the observed symmetry may be due to random variations.

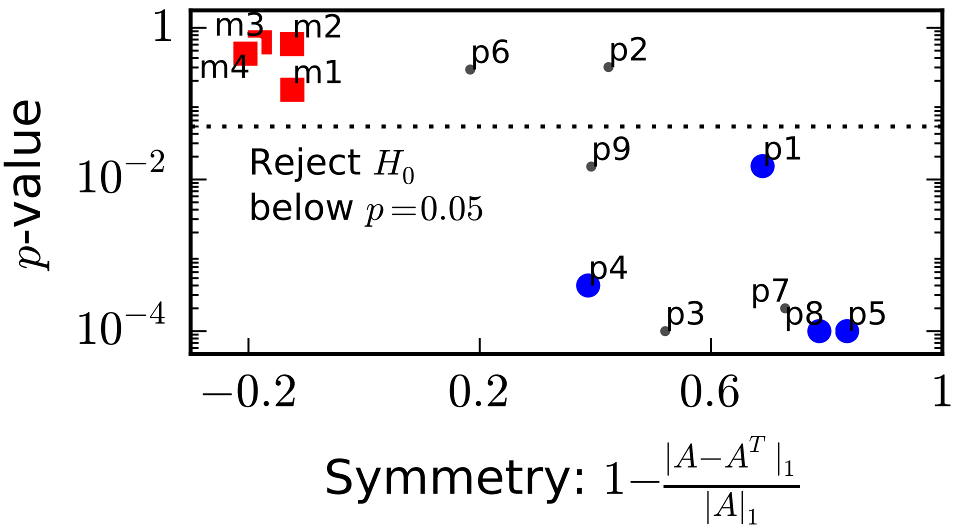


Figure 6.2: Symmetry of conversations in Polymath and mini-polymath projects. The horizontal axis is the amount of symmetry observed in comment threads: higher symmetry indicates that authors follow up comments from the same authors who follow up their comments, and the p-value on the vertical axis is the bootstrapped estimate of the level of significance at which we can reject the null hypothesis that the symmetry is due to random variations.

Similarly, in Table 6.3, we observe that in three of the four polymath projects under question there is significantly more stickiness than in the random baseline, whereas in three of four mini-polymath projects, there is less.

What we find surprising about these phenomena of increased symmetry and stickiness is not that it occurs at all, but that we observe it in the Polymath projects while not observing it to the same extent in the mini-polymath projects, which was hosted on the same platform and involved a similar group of people.

We expect that in the Polymath projects it is at least in part thanks to a norm that emerged from the collaboration: as conversation in each project developed, there were a large number of subproblems that needed to be completed (everything from running simulations, to reviewing related work, to building information sharing web-apps), and subgroups of people would work on them together. These subgroups of people would

Table 6.3: The increased likelihood of the motif in the Polymath projects and mini-polymath projects. *, **, and *** indicate that the result was significant when measured against a time-zone-controlled random baseline (as defined in the text) at the 0.05, 0.01, and 0.001 significant levels respectively. Otherwise, the number in parentheses indicates the $p-$value. nan indicates that there were no examples of this motif in the temporally-controlled random baseline. We measure stickiness based on the motif *aba* and contrast the results with *aaa*. The increased likelihood of *aaa* does not differ much between Polymath projects and mini-polymath projects.

Project	aaa	aba
Polymath 1	5.15***,	1.41 (0.25)
Polymath 4	2.34***,	1.42**
Polymath 5	3.51***,	1.54*
Polymath 8	3.65***,	1.91***,
Mini 1	3.55***,	1.5 (0.14),
Mini 2	5.14***,	0.86 (0.82)
Mini 3	1.9***,	0.68 (0.92)
Mini 4	nan***,	0.82 (0.79)

tend to communicate with each other more frequently than with other people, leading to the symmetry we have observed.

The apparent lack of stickiness in the mini-polymath projects compared to the polymath projects may indicate that the role of smaller groups discussing subproblems was less important in this easier problem domain.

6.4 Temporal level features

6.4.1 Response time dynamics

The time scale on which mini-polymath projects play out is quite different from that of the Polymath projects, with the latter taking place over the course of several months to a year and the former being concluded in a matter of days. This difference in overall time scales suggests that we consider contrasts in the responsiveness dynamics for Polymath versus mini-polymath projects: when an author posts a comment, how quickly do people follow up after them and how do those dynamics compare in the two types of collaborations? We find that the answer is subtle and depends on the temporal scale of analysis itself.

First, we define the *response time* of a comment to be the amount of time that has elapsed since the comment immediately preceding it was posted.[5] We then consider the mean response time in Polymath and mini-polymath, conditioned on those response times being less than some upper threshold. That is, for some value t, what is the mean response time of all comments whose response time is less then t? We denote this quantity by $\bar{r}^t_{main}$ and $\bar{r}^t_{mini}$ for Polymath and mini-polymath, respectively.

Given that mini-polymath projects played out much more quickly overall than Polymath projects, it would be natural to expect that response times on mini-polymath should be less than those on Polymath for all values of the threshold t; that is, one would expect a positive difference $\bar{r}^t_{main} - \bar{r}^t_{mini}$.

What we find is more subtle, in that it depends on the threshold t; we get a different answer if we condition on comments made within a few minutes of each other. In Figure 6.3 (top), we observe that when we focus on very small time scales of less than five minutes, commenting in Polymath is actually faster than in mini-polymath. This is reflected in the negative difference $\bar{r}^t_{main} - \bar{r}^t_{mini}$ for $t < 5$ minutes. And then (as expected) as we allow for comments with larger and larger response times, the mean response

[5] The blog data includes comment timestamps with one-minute granularity.

time in Polymath becomes larger than in mini-polymath. In the figure we report the mean difference, and consider the $p-$value corresponding to the significance with which we reject the null hypothesis that the means are the same, estimated using Welch's t-test (for comparing population means between populations with unequal variance). For all thresholds except 4 and 5 minutes, at which the transition between mean signs is observed, $p < 0.001$.

6.4.2 Momentum and acceleration: comment dynamics

Next we consider the question of how commenting rates evolve over time in the Polymath and mini-polymath projects. To explore this process we draw on two measures from physics for quantifying motion: acceleration and momentum. We define them formally below, but broadly speaking, *acceleration* captures whether authors are commenting on the project at a constant rate, an increasing rate, or a decreasing rate; *momentum* captures the overall rate at which the progress is advancing, considering both the rate at which authors are creating new comments, and also the amount of content that they are producing in those comments.

Definitions. Let us refer to the current "position" of the project as $x(t_i)$, where $x(t_i)$ is the number of comments that have been made up to time t_i. Then the project's instantaneous velocity and acceleration are the first and second time derivatives of $x(t)$, which can be measured using the central difference formula: $v(t_i) = x'(t_i) \approx \frac{x(t_{i+1}) - x(t_{i-1})}{t_{i+1} - t_{i-1}}$, and similarly for $a(t_i) = v'(t_i)$. We compute the average velocity with units of comments per minute, providing a summary measure of how rapidly each project progressed. The average acceleration then has units of comments per minute per minute, and tells us whether or not the speed of the project was picking up (positive acceleration) or slowing down (negative acceleration).

Finally, we introduce the notion of a comment's *momentum*: borrowing from physics, the momentum of an object is the product of its mass and its velocity. We interpret the number of characters in a comment as its mass and so compute the momentum as the product of a comment's length and its velocity. This notion of momentum

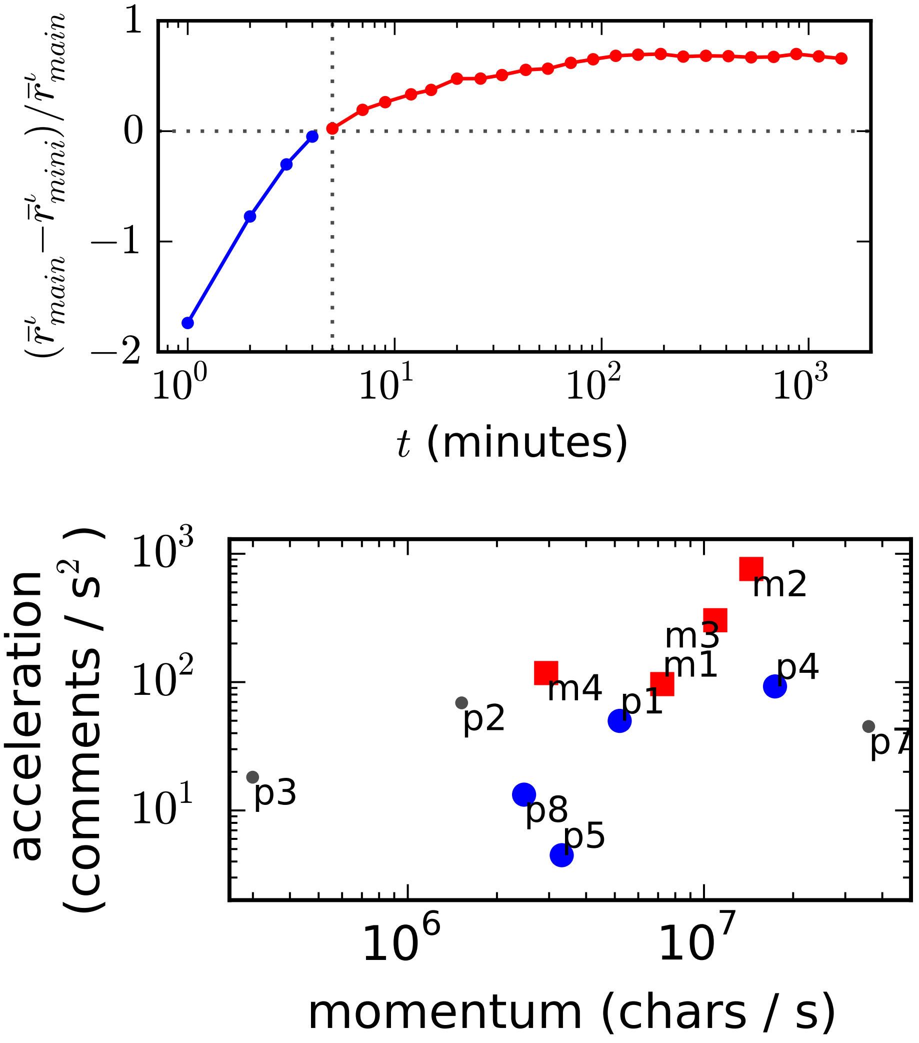


Figure 6.3: **Top figure**: (Blue vs. red indicates temporal regime where Polymath vs. mini-polymath has faster response time). Vertical axis: fractional difference in response times between the Polymath and mini-polymath, conditioned on the response time being less than what is indicated by the horizontal axis. Hence, negative values indicate that responses in Polymath are faster than in mini-polymath. **Bottom figure**: Average commenting acceleration vs. average commenting momentum. Velocity units are #comments per minute, and acceleration units are #comments per minute, per minute.

enables us to distinguish between projects with, for example, the same commenting *rate* but with different average comment *lengths*.

High-momentum projects pick up more speed. Surprisingly, in Figure 6.3 (bottom) we find that all Polymath and mini-polymath projects have a positive average acceleration. Earlier we observed that comment response times were on average faster in mini-polymath than in Polymath; we also observe that they tend to have higher acceleration

Perhaps most strikingly, in Figure 6.3 (bottom), we see that the average acceleration and momentum in this case have an approximately monotonic relationship with each other, meaning that the projects with the highest momentum were also the projects that were picking up the most speed. This monotonic relationship is not something to be expected a priori: for example, a project that started off with long, rapid comments and slowly decayed would have high average momentum and negative acceleration; but all of the examples observed here have the opposite pattern, with the higher momentum projects accelerating more rapidly.

6.5 Linguistic features

Following the plan outlined in the introduction, we continue by studying the distinctions between Polymath projects — representing research on open problems — and mini-polymath projects, which are efforts to solve Math Olympiad problems. This investigation offers the opportunity to understand the contrasts between these related but qualitatively different types of collaborative activities. In this section, we introduce the high-level linguistic features that we consider and the differences observed in how they manifest in the two domains.

Table 6.4: T-test results for high-level linguistic features. For each feature, we conduct a t-test from two independent samples, extracted from Polymath comments and mini-polymath comments respectively, where the null hypothesis is that the two kinds of comments come from the same distribution. The number of arrows in the table visually indicates the p-value magnitude: $p < 0.05$: 1 arrow, $p < 0.01$: 2 arrows, $p < 0.001$: 3 arrows, $p < 0.0001$: 4 arrows. ↑ indicates that Polymath comments have larger values; ↓ indicates that mini-polymath comments have larger values.

Feature	test results
Relevance	
similarity to original post	↑↑↑↑
similarity to current post	↑↑↑↑
Distinctiveness	
average log POS unigram prob	↑↑↑↑
average log POS bigram prob	-
average log POS trigram prob	↑
average log lexical unigram prob	-
average log lexical bigram prob	↑↑↑↑
average log lexical trigram prob	-
Politeness	
politeness [31]	↑↑↑↑
number of hedges	↑↑↑↑
fraction of words that are hedges	↓
Generality	
frac. indefinite articles	↓↓↓↓
frac. past tense	↑↑↑↑
frac. present tense	-

6.5.1 Exploring high-level linguistic features

Our set of high-level linguistic features draws on recent innovations in natural language processing that have been used for applications including the memorability of movie quotes [30], the effects of wording on message propagation [112] and the popularity of online posts [73]. We supplement these features with several more basic ones as well.

We divide the features into four groups: relevance, distinctiveness, politeness and generality. To get an initial understanding of how these features differ between Polymath and mini-polymath projects, for each one we conduct a t-test between feature values extracted from Polymath comments and mini-polymath comments (Table 6.4). We find that Polymath comments are indeed significantly different in many of these features compared to mini-polymath comments. Later in §6.6, we will see how they perform in a prediction setting in comparison to topic-based linguistic features, as well as the role- and temporal-based features discussed in §6.3 and §6.4.

We begin by describing the feature-level differences between Polymath and mini-polymath comments. For each category of differences, we summarize it first in a bold-faced sentence and then elaborate in the subsequent paragraph.

Research discussions match the original problems more closely. We first ask how much the language used in the discussion drifts away from the language used at the outset of the project to describe the problem. We do this by computing Jaccard similarity between each comment and the original post for the project. Since the discussions are segmented into *threads* of roughly 100 consecutive comments each, we also compute a related measure — the Jaccard similarity between each comment and the initial post in the thread it belongs to.

One might expect that since research discussions are open-ended, the language might drift quickly away from the description of the initial problem. In fact, we find that comments are significantly more similar to the original posts for Polymath projects, both in the original problem description and in the current post.

Research discussions have *less* distinctive language. One might expect the language in tackling hard research problems to be more "distinct" from daily language compared to that in solving problems with known solutions. We formalize distinctiveness using language model scores, defined as the average logarithm of word probabilities [30, 73, 112]. Our language model, based on frequencies of one, two, and three word sequences (unigrams, bigrams, and trigrams) of words and part-of-speech tags, is developed from the Brown corpus [71].

Perhaps somewhat surprisingly, research discussions resemble daily language more in terms of part-of-speech tag patterns. When it comes to actual words, research discussions also employ more common word patterns, although it is not statistically significant for unigrams and trigrams. The greater robustness of the part-of-speech analysis, in comparison to the word-level analysis, may reflect the fact that both projects contain a large amount of language infrequently used outside of mathematical discussions.

Research discussions are more polite. As participants are discussing harder problems for a longer period of time in Polymath projects, a natural hypothesis is that they are more polite to one another. We test this using a recently developed method for estimating the politeness of pieces of text [31], and we find that indeed there is significantly more politeness in the text of the Polymath projects.

We obtain an inconclusive comparison when we study the related phenomenon of *hedging* in the language use of the comments — a term coined by Lakoff [74] to describe the expression of uncertainty, which would be natural to have in comments discussing hard technical problems. Although Polymath comments have significantly more hedges, mini-polymath comments have a larger fraction of hedges.

Research discussions are more "specific". One hypothesis may be that we can see a difference in how general the arguments are, i.e., mini-polymath may be more specific due to the limited scope of the problem. Previous work has used the occurrences of indefinite articles and tense-related expressions to capture generality [30, 112]. Somewhat surprisingly, Polymath comments are less general, with significantly more past tense and fewer indefinite articles.

6.6 Predicting domain: research vs. hard problem solving

We now have a broad set of features characterizing the comments and can leverage them to use in our basic prediction problem. Our model uses these features to determine whether a given comment comes from a Polymath project or a mini-polymath project.

The features discussed above fall into three categories: author roles, temporal dynamics, and linguistics. We focus on the performance of each set in turn. The author roles can be further distinguished by whether they are being used anonymously (omitting author identities) or non-anonymously; the temporal by whether they are simple elapsed time differences or more nuanced dynamics metrics such as acceleration and momentum; and the linguistic properties by whether they have topic information or non-topic information.

Surprisingly, we will find that in a controlled setting, prediction using these anonymous structural and non-topical features can actually outperform topic-based and identity-based features. We also find that the dynamics metrics (drawn from physics) offer better prediction performance than the simpler, elapsed-time metrics.

Prediction setup. We set up balanced prediction tasks for distinguishing Polymath comments from mini-polymath comments. Specifically, as there are fewer comments in the mini-polymath projects, we sample a Polymath comment for each mini-polymath comment. Thus we have a pair of comments in each instance of our data, with one comment from each of Polymath and mini-polymath respectively. (We randomly order these two comments when presenting them to the algorithm.) We use two different ways of sampling pairs from the overall data.

- Random (704 pairs). For each mini-polymath comment, we randomly sample a comment from the Polymath projects.
- Controlled (203 pairs). For each mini-polymath comment, we find the Polymath comment from the same author with the minimum length difference in terms of the number of words. (We only use mini-polymath comments for which the same

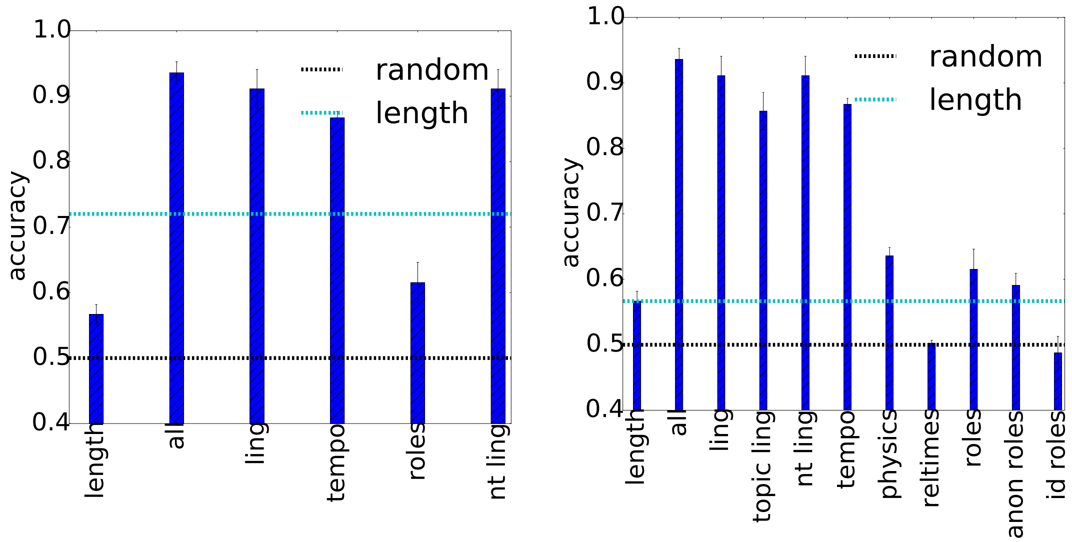
author has written at least one Polymath comment.) This constructs a much more difficult prediction task.

6.6.1 Feature definitions and motivations

We now discuss the features we use for the prediction task, drawing on the features defined above. Our plan is to compare the prediction performance using different sets of these features.

The features can be categorized as follows; the keyword in parentheses preceding each definition indicates the feature category as labeled in the performance results plots (Figures 6.4 and 6.5).

- Length. Given that comments in mini-polymath projects are generally shorter than the comments in Polymath projects, the length of a comment already provides a non-trivial baseline for prediction. Our notion of length actually includes three quantities for each comment: the number of words, the number of characters, and the number of MathJax characters as features.
- Roles
  - (id roles) Author and surrounding authors: numeric id of comment author and those authors of the ten comments leading up to it and the ten succeeding it;
  - (anon roles) Anonymous structural: same as *id roles* but with generic structural representation of the author sequence;
- Temporal
  - (reltimes) Elapsed times: hours, days, and minutes elapsed since project inception; number of comments and number of threads since project inception;
  - (physics) Dynamic properties: instantaneous velocity, acceleration, and momentum of comment, where position is defined as comment id, and mass of a comment is defined as the number of characters in it. These features are defined formally in §6.4.2.



(a) Overview with no additional controls

(b) Breakdown with author and length controls

Figure 6.4: Results for predicting Polymath comments vs. mini-polymath comments. x-axis: different feature sets, each group is defined in §6.6.1. y-axis: accuracy. Error-bars represents the standard error of the performance across 5 folds. The black line shows the performance of random guessing; the cyan line shows the performance of the length baseline. (a) and (b) respectively show the performance without any control and when we control author and length. (b) shows the more detailed performance breakdown for each of the three elements (roles: anon. vs. non-anon, timing: elapsed times vs. physics, linguistics: topic-based vs. non-topic based).

- Linguistic features. The linguistic features consist of *non-topical features* (denoted "nt-ling") listed in the first four bullet points, and *topical features* (denoted "topic ling") listed in the latter two bullet points.

  – (nt ling) High-level linguistic features, as discussed in §6.5.1: politeness, generality, specificity, hedging, fraction of *novel* words with respect to the entire preceding conversation or to a fixed-size window of previous comments.

  – (nt ling) LIWC. Linguistic Inquiry and Word Count (LIWC) includes a dictionary of words classified into different categories, along dimensions that include affective and cognitive properties [95]. We use the frequency of each LIWC category in a comment as features.

- (nt ling) Part-of-speech tags (POS). Part-of-speech tags can provide us with stylistic information for a comment. All possible part-of-speech tags are considered as features.[6].
- (nt ling) Stopwords from the NLTK[7]; most frequent 50 words from the training data; most frequent 100 words from training data.
- (topic ling) Bag-of-words (BOW). This is a very strong method typically used in natural language processing tasks. We include all the unigrams that occur at least 5 times in our training data as features. We use the tokenizer from the NLTK package after replacing urls and MathJax scripts with special tokens.
- (topic ling) Bag-of-words for the preceding and succeeding comments. The same definition as the feature above, but now for each of the five comments before the comment in question, and each of the five after.

Computational evaluation of prediction. We use 5-fold cross-validation in our computations to measure prediction performance. Since the task is balanced, we use accuracy as our evaluation metric. In the computations, for each feature set, we extract the values from each comment in a pair, and then take the differences between the first comment and the second comment in this pair. For BOW and POS based features, we normalize the feature vectors using L2-norms, while for the other features, the values are linearly scaled to $[0,1]$ based on training data. We use scikit-learn in all prediction computations.[8]

Prediction: Roles, Temporal. In Figure 6.4 we observe that using the anonymized roles (author motifs as discussed in §6.3.2) offers good performance. This positive performance may be due to the distinctions we observed above. In particular, the Polymath

[6] Througout we use the NLTK maximum entropy tagger with default parameters, which is based on the Penn Treebank Dataset (`http://www.cis.upenn.edu/~treebank/home.html`)

[7] `http://www.nltk.org/`

[8] `http://scikit-learn.org/`

projects tend to have larger and significant correlations in the reply structure of the comment threads.

We also observe that the temporal features offer significant improvements over the random baseline. As with the role features, this performance increase can potentially be understood as thanks to the substantial differences in temporal dynamics in the two projects that we discussed in §6.4.

Linguistic prediction performance: topical vs. non-topical. We make several observations about the prediction results based on linguistic-only features. First, all the feature sets improve on the length baseline for both the uncontrolled task (when we form a pair for each mini-polymath comment) and the controlled task (when we match the author and approximately match the length within each pair).

Second, the bag-of-words feature set slightly outperforms the non-topic feature set on the uncontrolled task, but when we add length and author controls, in fact the non-topic feature set significantly outperforms the bag-of-words features, achieving close to 90% accuracy. It is interesting that the non-topic feature set should achieve this, since it is not attuned to the content of the comments themselves. Moreover, the non-topic features actually give better performance on the controlled task than on the uncontrolled task, despite the fact that the controlled task was set up to limit the effectiveness of various features; meanwhile, the performance of the bag-of-words feature set in the controlled task (along with stopwords and POS) drops significantly.

As for individual categories, high-level linguistic features actually outperform all other non-topical categories despite the small number of features in this category, including commonly used LIWC features. This observation is robust across both tasks. It is worth noting that there are fewer high-level linguistic features than POS or LIWC features.

In terms of top features (Table 6.5), similarity to the original problem statement is the most prominent signal for Polymath comments, followed by part-of-speech tags including adjectives; in contrast, LIWC categories and part-of-speech tags tend to be top indicators of mini-polymath comments. Table 6.5 also shows the top word-level

Table 6.5: Top 20 features in Polymath vs. mini-polymath prediction. Features are separated by spaces. High-level linguistic features are in quotes. Other non-topical features are named by concatenating the category name and feature name; for instance, “POS-adj” means the feature “adjectives” from the part-of-speech category.

Top bag-of-word features	
Polymath	sequences “ is sequence primes prime - now values at ” in different of by 3 also latex paper x
mini-polymath	m then can points ... mine number mines point n coins proof moves comments added all any partial thread 2
Top non-topical features	
Polymath	“similarity to original post” “similarity to current post” POS-adjective POS-adverb “POS-verb (past)” POS-“ “frac. past tense” POS-preposition liwc-work POS-noun numchars liwc-adverb liwc-auxverb nummathchars liwc-preps “POS-verb (non-3rd present)” liwc-they POS-: liwc-time “average log unigram prob (lexical)”
mini-polymath	liwc-motion liwc-assent liwc-we liwc-certain liwc-cause liwc-negemo liwc-achieve “frac. indefinite articles” liwc-filler liwc-conj liwc-nonfl liwc-quant liwc-number POS-NONE “POS-adjective (superlative)” “POS-verb (base form)” POS-$ “POS-proper noun (singular)” POS-determiner POS-particle

features that emerged for the bag-of-words feature set, including topical words such as “sequence”, “prime” and “mine”[9].

[9] “Mine,” in the sense of an explosive device, occurred in one problem in IMO.

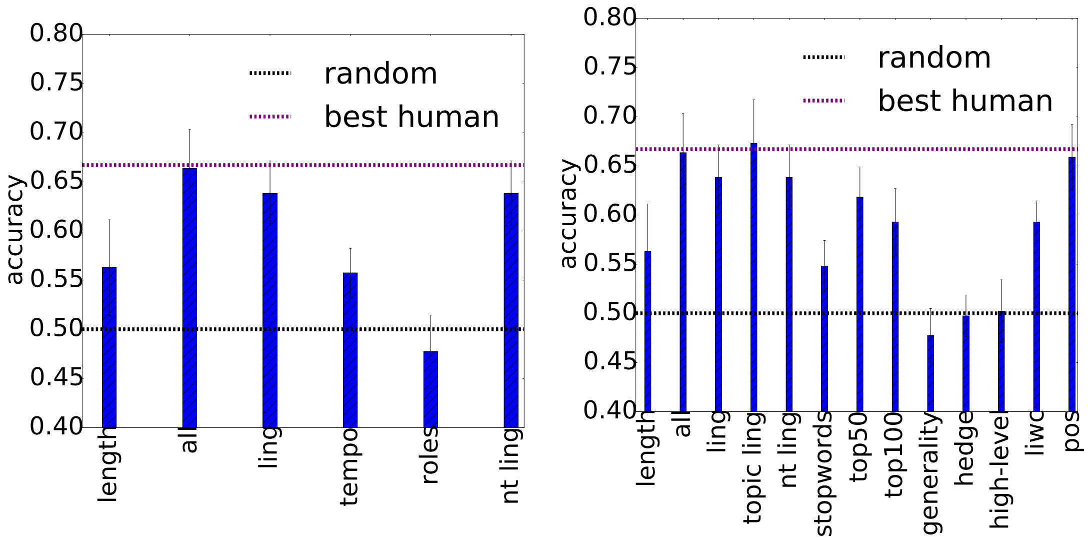
6.7 Identifying research highlights: intrinsic vs. contextual evidence

We now investigate the second main question we posed in the introduction: Are research breakthroughs identifiable in a string of comments? If they are, can one best recognize them solely from their content, a finding that could indicate that authors know the eventual importance of their statements? Or are breakthroughs best recognized by the (re-)actions of others, suggesting that it can be hard to know in the heat of the moment which results are key ones?

Polymath 1 serves as a particularly nice setting for investigating this question because, fortunately, breakthroughs have already been identified by a domain expert: Terence Tao set up a wiki timeline of Polymath 1 highlights.[10] While Cranshaw and Kittur [26] employed this highlights list to study *whether less active users* had impact, we use the list to constitute instances for the task of classifying *which comments* have impact, and to identify the most helpful intrinsic vs. extrinsic features for this task.

Prediction setup. In setting up the prediction task, we employed two paradigms: (A) classifying individual instances as being either a highlight or not, or (B) choosing one comment from a pair where it is known that exactly one was a highlight, and the other is the non-highlight written by the *same* author that is closest in length to the highlight. Due to space constraints, we only describe (B) in this paper, for three reasons. First, author- and length-controlled findings are more likely to generalize to other settings. Second, we believe that for judges (human or algorithmic) that are not domain experts, it is more reasonable to be asked to pick the more important-looking comment in a pair than to judge a single text in isolation. Third, describing (B) allows us to be more concise than describing (A), where mechanisms for handling class imbalance would need to be explained. We note, though, that (B) gives us less data to work with (since

[10] `http://goo.gl/ijbIqP`. The page's revision history reveals that Tao's intent was indeed to list "highlights", but he switched to the milder term "events" to alleviate Gowers' apparent embarrassment at one of his contributions being deemed "highlight-worthy".



(a) Overview; author and length controls

(b) Linguistic breakdown; author and length controls

Figure 6.5: Highlight-prediction results for different feature sets. Error bars: standard error for 5-fold cross-validation. Black line: random guessing. Purple line: best human performance on the author-control-only (easier) task.

we can only construct as many pairs as there are highlights), and doesn't directly map onto the application of classifying individual comments as they naturally appear.

Feature sets and cross validation. For this task, we use the same feature sets that we employed for our first task of distingishing Polymath comments from mini-polymath comments. These features are described in §6.6. Further, in all experiments, we employ the same experiment protocol as in the first task. Random guessing yields a baseline accuracy of 50%.

6.7.1 Prediction Performance

To assess the difficulty of the task for humans as a point of comparison for our algorithmic performance, we asked three Applied Mathematics graduate students to attempt the classification task on 30 author-controlled pairs in an approximately 30-minute session.[11] They got accuracies of 66.7%, 63% and 46% (agreeing 60% of the time); these results, together with post-hoc feedback from the students, indicates that our task is fairly difficult.

Prediction performance: roles, temporal. In Figure 6.5 we observe that the features based on the authors' roles in the project and those based on temporal properties do not offer additional prediction performance beyond what can be achieved by a random baseline.

Prediction performance: topical vs. non-topical linguistic. Meanwhile, the linguistic features perform 15% above the random baseline, and in fact achieve the same performance as that of the human evaluators. It is interesting to note that the text was all that was available to the human judges, just as it was precisely the words and high-level linguistic features and parts of speech that were available to the model we trained. We continue by further exploring the performance of these varying groups of linguistic features.

Figure 6.5 shows that the best-performing classifier uses bag-of-words features and yields accuracy comparable to that of our best human subject (on a slightly simpler task). The second best performing feature set is part-of-speech tags. Adding other non-topical features actual hurts the performance slightly in this task. Neither preceding comments or reactions outperforms comment-internal features. All in all, the evidence suggests that authors often write texts that eventually turn out to be highlights in a fashion that indicates they may be aware of the importance of their remarks at the time.

6.8 Related work

Shortly after Polymath 1's success Tim Gowers and Michael Nielsen wrote a retrospective opinion piece on open collaboration in Nature [46], in which they took the opportunity to share their vision for the incredible potential that the Web offers to the

[11] At the time, we had not installed the length controls, but the task is strictly easier when length is a potential clue.

future of science, as a collaborative tool that is ideal for facilitating communication and information sharing.

Michael Barany [14] wrote about Polymath from a qualitative sociological perspective, focusing on the interaction of the participants with the technological system that supported the collaboration. In particular, he considers the mutual adaptation of that technological system, the participants, and the overall collaboration as the project advanced from its uncertain beginning to a successful conclusion.

In addition to Barany's piece, Cranshaw and Kittur [26] provide a quantitative overview of Polymath 1. They find that activity tends to spur other activity, and that activity by either of the two leaders, Terence Tao and Tim Gowers, tends to spur even more activity. They observed that the numbering-threading convention was successful in allowing multiple threads to develop simultaneously, but that cross-references were limited. By constructing the comment mention-graph and cross-referencing authors' Wordpress profiles with Google Scholar accounts they were able to show that, while the top two contributors were the Fields Medalists, there were much "smaller names" close behind – indicating that Gowers's vision of the project being accessible to a broad audience was achieved at least in part.

Finally, mini-polymath has been studied by Pease and Martin [94]; they show how the approaches there follow well-studied frameworks for problem-solving.

6.9 Conclusion

Polymath is an interesting experiment in promoting Internet collaboration on a type of activity — working on open mathematical research problems — that is otherwise not really represented in large open online collaborative efforts. Using this site as a lens, we have sought to contrast Internet collaborations on open research problems with Internet collaborations on "merely" difficult problems.

Limitations. While Polymath is the most visible effort at open Internet collaboration on mathematical research problems, one should be careful about generalizing too far from a single domain. Moreover, we can ask whether there are specific aspects of Polymath that played a role in the findings. Perhaps most importantly, the participation guidelines of the main Polymaths promoted rapid, incremental posting over the arguably more typical research mode wherein one engages in longer periods of off-line reflection and independent thought. The (laudable) intent was to make the project more accessible, but it is possible that the collaboration was less natural as a result. Regardless of these concerns, of course, it is clear that several projects had successful outcomes, resulting in publications and/or important partial progress toward the stated goal.

Future Directions. Many of our findings open up promising future directions. First, the reply-time properties are interesting, with the intriguing fact that Polymath, which is significantly slower than Mini-Polymath overall, becomes faster at the shortest time scales. We would like to understand the reason for this fast pace; it is also natural to ask whether this "organically" developed fast pace is good for collaborations, or whether it is more effective to proceed more slowly at the shortest time scales. It is also interesting to ask whether we can trace any potential effects that the high-level linguistic properties have on the trajectory of the discussion or the quality of the outcome.

Finally, our second prediction task, on identifying highlights in real time, raises potential questions for the design of future iterations of Polymath-style sites. If it were possible to flag predicted highlights as they happen, is this a useful thing to make explicit for a group engaged in research? And if so, is it more productive to call attention to these predicted highlights as they happen, or at a later point? Questions in this style point to the potential opportunities for algorithms trained on this type of data to assist in guiding future discussions, when on-line groups assemble to work on hard problems together.

CHAPTER 7

CONCLUSION

APPENDIX A

CHAPTER 1 OF APPENDIX

BIBLIOGRAPHY

[1] Bruno Abrahao, Sucheta Soundarajan, John Hopcroft, and Robert Kleinberg. On the separability of structural classes of communities. In *In KDD '12*, pages 624–632. ACM, 2012.

[2] Juan A. Acebrón, L. L. Bonilla, Conrad J. Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.*, 77:137–185, Apr 2005.

[3] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[4] Elizabeth S Allman, Catherine Matias, and John A Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, pages 3099–3132, 2009.

[5] Elizabeth S Allman, Catherine Matias, and John A Rhodes. Parameter identifiability in a class of random graph mixture models. *Journal of Statistical Planning and Inference*, 141(5):1719–1736, 2011.

[6] Christophe Ambroise and Catherine Matias. New consistent and asymptotically normal parameter estimates for random-graph mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):3–35, 2012.

[7] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006.

[8] Reid Andersen and Kevin J Lang. Communities from seed sets. In *WWW*, pages 223–232, 2006.

[9] Konstantin Andreev and Harald Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.

[10] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 44–54. ACM, 2006.

[11] James P Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05001, 2008.

[12] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: Quantifying influence on twitter. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 65–74, New York, NY, USA, 2011. ACM.

[13] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM, 2012.

[14] Michael J. Barany. '[b]ut this is blog maths and we're free to make up conventions as we go along': Polymath1 and the modalities of 'massively collaborative mathematics'. In *Proc. Symp. on Wikis and Open Collaboration*, 2010.

[15] Frank Bass. A new product growth for model consumer durables. *Management Sciences*, 15(1):215–227, 1969.

[16] K. Binder and A. P. Young. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Rev. Mod. Phys.*, 58:801–976, Oct 1986.

[17] LL Bonilla, CJ Pérez Vicente, and JM Rubi. Glassy synchronization in a population of coupled oscillators. *Journal of Statistical Physics*, 70(3-4):921–937, 1993.

[18] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 925–936, New York, NY, USA, 2014. ACM.

[19] L. M. Childs and S. H. Strogatz. Stability diagram for the forced Kuramoto model. *Chaos*, 18(4):043128, December 2008.

[20] TC Choy and D Sherrington. The van hemmen model-a true spin glass? *Journal of Physics C: Solid State Physics*, 17(4):739, 1984.

[21] Fan Chung. The heat kernel as the pagerank of a graph. *PNAS*, 104(50):19735–19740, 2007.

[22] Fan Chung. A local graph partitioning algorithm using heat kernel pagerank. *Internet Mathematics*, 6(3):315–330, 2009.

[23] Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72(2):026132, 2005.

[24] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[25] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[26] Justin Cranshaw and Aniket Kittur. The Polymath project: Lessons from a successful online collaboration in mathematics. In *Proc. CHI*, 2011.

[27] H. Daido. Quasientrainment and slow relaxation in a population of oscillators with random and frustrated interactions. *Phys. Rev. Lett.*, 68:1073–1076, February 1992.

[28] Hiroaki Daido. Population dynamics of randomly interacting self-oscillators. i: Tractable models without frustration. *Prog. Theor. Phys.*, 77(3):622–634, 1987.

[29] Hiroaki Daido. Algebraic relaxation of an order parameter in randomly coupled limit-cycle oscillators. *Phys. Rev. E*, 61:2145–2147, Feb 2000.

[30] Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. You Had Me at Hello: How Phrasing Affects Memorability. In *Proc. ACL*, 2012.

[31] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A Computational Approach to Politeness with Application to Social Factors. In *Proc. ACL*, 2013.

[32] G. de Tarde and E.W.C. Parsons. *The Laws of Imitation*. H. Holt, 1903.

[33] Thomas Debeauvais, Bonnie Nardi, Diane J Schiano, Nicolas Ducheneaut, and Nicholas Yee. If you build it they might stay: Retention mechanisms in world of warcraft. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 180–187. ACM, 2011.

[34] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

[35] Nicolas Ducheneaut, Nicholas Yee, Eric Nickell, and Robert J. Moore. The life and death of online gaming communities: A look at guilds in world of warcraft. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 839–848, New York, NY, USA, 2007. ACM.

[36] Martin E. Dyer and Alan M. Frieze. The solution of some random np-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451–489, 1989.

[37] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[38] Andreas Emil Feldmann. Fast balanced partitioning is hard even on grids and trees. In *Mathematical Foundations of Computer Science 2012*, pages 372–382. Springer, 2012.

[39] Andreas Emil Feldmann and Luca Foschini. Balanced partitions of trees and applications. *Algorithmica*, 71(2):354–376, 2015.

[40] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[41] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[42] Ove Frank and Frank Harary. Cluster inference by using transitivity indices in empirical graphs. *Journal of the American Statistical Association*, 77(380):835–840, 1982.

[43] Wolfgang Glänzel and András Schubert. Analysing scientific networks through co-authorship. In *Handbook of Quantitative Science and Technology Research*, pages 257–276. Springer, 2005.

[44] David Gleich and Michael Mahoney. Anti-differentiating approximation algorithms: A case study with min-cuts, spectral, and flow. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1018–1025, 2014.

[45] Timothy Gowers. Is massively collaborative mathematics possible?, January 2009.

[46] Timothy Gowers and Michael Nielsen. Massively collaborative mathematics. *Nature*, 461(7266):879–881, 2009.

[47] Aaron M Hagerstrom, Thomas E Murphy, Rajarshi Roy, Philipp Hövel, Iryna Omelchenko, and Eckehard Schöll. Experimental observation of chimeras in coupled-map lattices. *Nature Physics*, 8(9):658–661, 2012.

[48] Taher H Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526. ACM, 2002.

[49] Bruce Hendrickson and Tamara G Kolda. Graph partitioning models for parallel computing. *Parallel computing*, 26(12):1519–1534, 2000.

[50] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[51] Hyunsuk Hong and Steven H. Strogatz. Kuramoto model of coupled oscillators with positive and negative coupling parameters: An example of conformist and contrarian oscillators. *Phys. Rev. Lett.*, 106:054102, Feb 2011.

[52] Daniel B. Horn, Thomas A. Finholt, Jeremy P. Birnholtz, Dheeraj Motwani, and Swapnaa Jayaraman. Six Degrees of Jonathan Grudin: A Social Network Analysis of the Evolution and Impact of CSCW Research. In *Proc. CSCW*, 2004.

[53] Charles H Hubbell. An input-output approach to clique identification. *Sociometry*, 1965.

[54] D. Iatsenko, S. Petkoski, P. V. E. McClintock, and A. Stefanovska. Stationary and Traveling Wave States of the Kuramoto Model with an Arbitrary Distribution of Frequencies and Coupling Strengths. *Phys. Rev. Lett.*, 110(6):064101, February 2013.

[55] Dmytro Iatsenko, Peter VE McClintock, and Aneta Stefanovska. Oscillator glass in the generalized kuramoto model: synchronous disorder and two-step relaxation. *arXiv preprint arXiv:1303.4453*, 2013.

[56] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW*, pages 271–279. ACM, 2003.

[57] Benjamin F. Jones, Stefan Wuchty, and Brian Uzzi. Multi-university research teams: Shifting impact, geography, and stratification in science. *Science*, 322(5905):1259–1262, 2008.

[58] Sanjay Ram Kairam, Dan J. Wang, and Jure Leskovec. The life and death of online groups: Predicting group growth and longevity. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 673–682, New York, NY, USA, 2012. ACM.

[59] Sepandar Kamvar, Taher Haveliwala, Christopher Manning, and Gene Golub. Exploiting the block structure of the web for computing pagerank. *Stanford University Technical Report*, 2003.

[60] Marcel Karnstedt, Matthew Rowe, Jeffrey Chan, Harith Alani, and Conor Hayes. The effect of user features on churn in social networks. In *Proceedings of the 3rd International Web Science Conference*, page 23. ACM, 2011.

[61] George Karypis and Vipin Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report, University of Minnesota, 1995.

[62] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[63] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[64] Aniket Kittur and Robert E. Kraut. Harnessing the wisdom of crowds in wikipedia: Quality through coordination. In *Proc. CSCW*, 2008.

[65] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, pages 668–677. Society for Industrial and Applied Mathematics, 1998.

[66] Kyle Kloster and David F Gleich. Heat kernel based community detection. In *KDD*, pages 1386–1395. ACM, 2014.

[67] Isabel M Kloumann and Jon M Kleinberg. Community membership identification from small seed sets. In *KDD*, pages 1366–1375. ACM, 2014.

[68] M. Komarov and A. Pikovsky. Effects of nonresonant interaction in ensembles of phase oscillators. *Phys. Rev. E* , 84(1):016210, July 2011.

[69] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *PNAS*, 110(52):20935–20940, 2013.

[70] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence*. Springer, Berlin, 1984.

[71] Henry Kučera and Nelson Francis. *Computational analysis of present-day American English*. Brown University Press, 1967.

[72] Carlo R. Laing. The dynamics of chimera states in heterogeneous kuramoto networks. *Physica D*, 238(16):1569 – 1588, 2009.

[73] Himabindu Lakkaraju, Julian McAuley, and Jure Leskovec. What's in a Name? Understanding the Interplay between Titles, Content, and Communities in Social Media. In *Proc. ICWSM*, 2013.

[74] George Lakoff. Hedges: a study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2, 1975.

[75] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. In *16th ACM Conference on Economics and Computation*, pages 228–237, New York, NY, USA, 2006. ACM Press.

[76] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM.

[77] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *In WWW '10*, pages 631–640. ACM, 2010. data source.

[78] Chen Liu, David R Weaver, Steven H Strogatz, and Steven M Reppert. Cellular construction of a circadian clock: period determination in the suprachiasmatic nuclei. *Cell*, 91(6):855–860, 1997.

[79] Feng Luo, James Z Wang, and Eric Promislow. Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6(4):387–400, 2006.

[80] E. A. Martens, E. Barreto, S. H. Strogatz, E. Ott, P. So, and T. M. Antonsen. Exact results for the kuramoto model with a bimodal frequency distribution. *Phys. Rev. E*, 79:026204, Feb 2009.

[81] Andrew Mehler and Steven Skiena. Expanding network communities from representative examples. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(2):7, 2009.

[82] Alan Mislove, Bimal Viswanath, Krishna P Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *In WSDM '10*, pages 251–260. ACM, 2010.

[83] E. Montbrió and D. Pazó. Collective synchronization in the presence of reactive coupling and shear diversity. *Phys. Rev. E* , 84(4):046206, October 2011.

[84] Elchanan Mossel, Joe Neeman, and Allan Sly. Belief propagation, robust reconstruction, and optimal recovery of block models. *COLT*, 2014.

[85] Lev Muchnik, Sinan Aral, and Sean J Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.

[86] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[87] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[88] Jukka-Pekka Onnela and Felix Reed-Tsochas. Spontaneous emergence of social influence in online systems. *Proceedings of the National Academy of Sciences*, 107(43):18375–18380, 2010.

[89] E. Ott and T. M. Antonsen. Long time evolution of phase oscillator systems. *Chaos*, 19(2):023117, June 2009.

[90] Ed Ott and Thomas M Antonsen. Low dimensional behavior of large systems of globally coupled oscillators. *Chaos*, 18(3):037113, 2008.

[91] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. *Stanford InfoLab Technical Report*, 1999.

[92] Gabriel H. Paissan and Damin H. Zanette. Synchronization of phase oscillators with heterogeneous coupling: A solvable case. *Physica D*, 237(6):818 – 828, 2008.

[93] Wei Pan, Nadav Aharony, and Alex Pentland. Composite social network for predicting mobile apps installation. *CoRR*, abs/1106.0359, 2011.

[94] Alison Pease and Ursula Martin. Seventy four minutes of mathematics: An analysis of the third Mini-Polymath project. In *Proc. Artificial Intell. Simulation of Behavior*, 2012.

[95] James W Pennebaker, Martha E. Francis, and Roger J. Booth. *Linguistic inquiry and word count: LIWC 2007*, 2007.

[96] A. Pikovsky and M. Rosenblum. Partially Integrable Dynamics of Hierarchical Populations of Coupled Oscillators. *Phys. Rev. Lett.*, 101(26):264103, December 2008.

[97] Arkady Pikovsky, Michael Rosenblum, and Jürgen Kurths. *Synchronization*. Cambridge University Press, 2003.

[98] Bruno Ribeiro. Modeling and predicting the growth and death of membership-based websites. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 653–664, New York, NY, USA, 2014. ACM.

[99] Bruno Ribeiro and Christos Faloutsos. Modeling website popularity competition in the attention-activity marketplace. *arXiv preprint arXiv:1403.0600*, 2014.

[100] Jason Riedy, David A Bader, Karl Jiang, Pushkar Pande, and Richa Sharma. Detecting communities from given seeds in social networks. Technical report, Georgia Institute of Technology, 2011.

[101] Everett M. Rogers. *Diffusion of Innovations, 5th Edition*. Free Press, August 2003.

[102] Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 695–704, New York, NY, USA, 2011. ACM.

[103] M.J. Salganik, P.S. Dodds, and D.J. Watts. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. *Science*, 311(5762):854–856, 2006.

[104] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35:1792–1796, Dec 1975.

[105] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

[106] Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *In STOC '04*, pages 81–90. ACM, 2004.

[107] J. C. Stiller and G. Radons. Dynamics of nonlinear oscillators with random interactions. *Phys. Rev. E*, 58:1789–1799, Aug 1998.

[108] J. C. Stiller and G. Radons. Self-averaging of an order parameter in randomly coupled limit-cycle oscillators. *Phys. Rev. E*, 61:2148–2149, Feb 2000.

[109] Steven H Strogatz. From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1):1–20, 2000.

[110] Steven H Strogatz. *Sync*. Hyperion, 2003.

[111] Eric Sun, Itamar Rosenn, Cameron Marlow, and Thomas Lento. Gesundheit! Modeling Contagion through Facebook News Feed. In *International AAAI Conference on Weblogs and Social Media*, 2009.

[112] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter. In *Proc. ACL*, 2014.

[113] Terence Tao. The erdos discrepancy problem. *arXiv preprint arXiv:1509.05363*, 2015.

[114] Mark R Tinsley, Simbarashe Nkomo, and Kenneth Showalter. Chimera and phase-cluster states in populations of coupled chemical oscillators. *Nature Physics*, 8(9):662–665, 2012.

[115] Johan Ugander, Lars Backstrom, Cameron Marlow, and Jon Kleinberg. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences*, 109(16):5962–5966, 2012.

[116] Jan L van Hemmen. Classical spin-glass model. *Phys. Rev. Lett.*, 49:409–412, 1982.

[117] Alexei Vázquez, João Gama Oliveira, Zoltán Dezsö, Kwang-Il Goh, Imre Kondor, and Albert-László Barabási. Modeling bursts and heavy tails in human dynamics. *Phys. Rev. E*, 73:036127, Mar 2006.

[118] Thomas J Walker. Acoustic synchrony: two mechanisms in the snowy tree cricket. *Science*, 166(3907):891–894, 1969.

[119] Ingmar Weber, Venkata R Kiran Garimella, and Alaa Batayneh. Secular vs. islamist polarization in egypt on twitter. In *In ASONAM '13*, pages 290–297. ACM, 2013.

[120] L. Weng, F. Menczer, and Y.-Y. Ahn. Virality prediction and community structure in social networks. *Sci. Rep.*, 3(2522), 2013.

[121] Joyce Jiyoung Whang, David F Gleich, and Inderjit S Dhillon. Overlapping community detection using seed set expansion. In *In CIKM '13*, pages 2099–2108. ACM, 2013.

[122] Arthur T. Winfree. Biological rhythms and the behavior of populations of coupled oscillators. *J. Theor. Biol.*, 16(1):15 – 42, 1967.

[123] Baoning Wu and Kumar Chellapilla. Extracting link spam using biased random walks from spam seed sets. In *In AIRWeb '07*, pages 37–44. ACM, 2007.

[124] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 705–714, New York, NY, USA, 2011. ACM.

[125] Shaomei Wu, Chenhao Tan, Jon Kleinberg, and Michael Macy. Does bad news go away faster. In *In In Proceedings of the International Conference on Weblogs and Social (ICWSM*, 2011.

[126] Yutaka Yamauchi, Makoto Yokozawa, Takeshi Shinohara, and Toru Ishida. Collaboration with lean media: How open-source software succeeds. In *Proc. CSCW*, 2000.

[127] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 177–186, New York, NY, USA, 2011. ACM.

[128] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 3. ACM, 2012.

[129] Jiang Yang, Xiao Wei, Mark S Ackerman, and Lada A Adamic. Activity lifespan: An analysis of user survival patterns in online knowledge sharing communities. In *ICWSM*, 2010.