1-1-2020

Data-driven Personalized Applications in Networks

Chuankai An
Dartmouth College

Data-driven Personalized Applications in Networks

Dartmouth Computer Science Technical Report TR2020-875

by

Chuankai An
Dartmouth College
Hanover, New Hampshire

January 2020

Abstract

A network models relationships. For a network that either encodes or supports internal information sharing activities, a better understanding of the network may enable data-driven applications (e.g., social network based recommendation), and boost both descriptive and predictive modeling of information flow in itself.

In a multi-faceted manner, we propose in this thesis to contribute to several challenges that arise in the development of personalized applications in the general area of information and networks: 1) articulation of new patterns (and associated metrics) for individual user behavior and network structure; 2) exploitation of new forms of feature vector representations derived from large datasets integrating users and network structure; 3) modeling the space of information flow with network science models and in particular, the prediction of direction, outlier, and outcome for information flow; 4) improving the transparency of a network-based recommender system to enable exploration of the underlying information space. The proposed methodologies combine machine learning models, network analysis and statistical analysis, which can successfully address open problems in the field. They are validated on a range of real data and show practical significance in providing widely applicable models and displaying increased accuracy over useful baselines.

Acknowledgments

First of all, I would like to thank my advisor Prof. Dan Rockmore, for his insightful guidance and continuous encouragement. Without his firm support and patience, I would not have the freedom and opportunity to pursue my interests, including this thesis. It is my great fortune to work with such an excellent advisor in my PhD study.

Now I want to thank the other committee members. Prof. James O'Malley coadvised me on several amazing and projects. His domain knowledge makes our projects much more impactful. Prof. Soroush Vosoughi and Prof. Cristopher Moore kindly gave valuable suggestions on this thesis. Thank you all for serving on my thesis committee.

I would also like to thank my other coauthors, Corey Stock, Reed Harder, Alfredo Velasco, Michael Evans. Additionally, a number of people helped me a lot during my PhD study in various ways. Dr. Chen Fang and Dr. Ye Xu, two former students in Dan's group, shared their successful career experience in academia and industry with me. During my summer internships, my mentors Pavel Kalinin, Hongzhao Huang and other talented colleagues supported my practice of "big data" projects. At the beginning of my PhD study, I also gained experience from the DartNets Lab.

I would like to thank my parents, Fengshuang and Jie. Their selfless love keeps me moving forward towards being the first PhD holder in my family. Though Hanover is far away from my hometown Jinan, Peilin, Zhao, Janica, Yinan, Jingxuan, Yue, Didi, Xinqi, Mengjia, Mengling, Mubing, Rui, Keith, Kizito and other friends embraced me with their warm hearts. Thank you for our precious friendship and making the years at Dartmouth a great memory.

At last, let me thank myself for being a hero: to see the world as it is and to love it. I will never forget why I started, and my mission can be accomplished.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Networks and network analysis have proved to be an extraordinarily useful mathematical paradigm for the understanding of interacting and related objects and the dynamics occurring between agents as well as the dynamics that take place by virtue of their connection. In this thesis, we consider both network dynamics and information flow in a network relevant to the enhanced utility of network structures in a variety of applications.

A network articulates explicit structures describing the connections among a group of entities, such as a social network of people built with their interactions. The amount of data associated within a network, in terms of the number of nodes (entities or agents) and their connections as well as the metadata attached to them, is growing explosively in different domains. Particular examples include social media platforms and activity records of professional collaboration. In such cases, big data acts as a useful source for building network-based applications for the benefit of the whole community, however different user groups (down to the scale of the individual, and known either directly in the metadata or discovered through some aspect of commonality) often require more accurate and even personalized service.

In a network, the interactions between nodes may transfer a kind of information, such as the diffusion of news in a social network and the sharing of knowledge and skills on an employee's career path in a network of companies from one job to another. If we group the activities of information sharing by the ID of a piece of information or the "walker" who/which jumps from one node to another in the network, we will get multiple parallel tracks of information flow in the network. A better understanding of information flow will provide explanations about the dynamics in a network, and suggest new powerful personalized predictive applications.

Given the diverse and dynamic nature of a network and the information flow inside, we face multiple challenges in personalizing applications. Before building the model in such an application, we would like to get some insights about the walkers (i.e., users) who/which carry a kind of information as they traverse the underlying network (e.g., a social network). A particular application area of interest for our work is healthcare where data can be used to create networks of patients and providers, either separately or together. Past work [Barnett et al., 2012a, Mandl et al., 2014] did not apply network science models to large-scale patient-physician visiting records. In our work we make use of the collection of derived physician interactions through the *a patient referral network* defined by us (and others). Here a patient plays the role of a walker who transfers her treatment history to multiple nodes in the network of physicians. Among the opportunities that the non-network-based approach missed – and which we uncovered in our work – is the representation of a sequence of visited physicians by the same patient as a network walk, which corresponds to the track of information flow in a generalized network, the statistics of which can produce important metrics of physician characterization.

Another area of interest for us is information diffusion in a network. We find that most previous work [Bourigault et al., 2016, Kempe et al., 2003, Yang and Counts, 2010] about information diffusion in a social network (e.g., news on Twitter) explains only the paths of observed information flow, without the prediction of future direction. In contrast to the explosive "multi-track" (a node could pass a piece of information to multiple nodes) news information flow in an directed acyclic network where the path of information diffusion does not contain duplicate nodes, the contexts of patient referrals, travelers visiting places, and career paths represent a class of "single-track" information flow. Single-track information flow could be described as a "walk" in the network that may contain duplicate nodes. In the single-track scenario where duplicate nodes on the track are allowed, for the purpose of personalized prediction, we need to design a new model to predict the next node that will receive the information after a time point of observation. Several applications related to information walk requires a predictive model as well, such as predicting the future outcome of a given event (e.g., treatment outcome of a patient) related to the information walk.

Finally, many popular websites/mobile applications (e.g., Yelp and Rotten Tomatoes) produce recommendations based on either inferred or explicit social networks. The inference is influenced by and evolves according to many hidden (at least to the user) variables. The track of browsing on such platforms is also an information walk in the network of candidate items. A final piece of this dissertation investigates the idea that greater transparency and interpretability in the recommendation engine would be of interest and excite more active participation in the recommendation platform.

To address the above challenges in the context of several datasets and applications, we investigate four general research problems to improve the performance, robustness, and interpretability of network-based personalized applications. Our first goal is to better understand the behavior of users and structural patterns of a network. Of particular interest is the interaction between generative models and network analysis for explaining user behavior and network structures. Second, is a goal of feature engineering with novel features and the incorporation and integration of diverse data into a feature vector. Open public databases and network science models present opportunities for new relevant material. Third, we investigate the personalized applications (e.g., the next visited node prediction) of information flow in a network. As a part of this, we introduce the notion of an *information walk* in a network and investigate its realization in various explicit contexts. Based on the network structure patterns and the observed information flow, a preference score in the Bayesian Personalized Ranking framework might be able to predict the next node that receives the information. Finally, we explore a framework to improve the transparency of a personalized application, such as a recommender system. When users are walking in the vast network space of candidate items, we hope that visualization of a user's feature vector space contributes to transparent network navigation in a recommender system. In addition to a traditional recommendation performance evaluation, we implement a user study to quantify the degree of improved transparency in user experience.

Personalized applications about information flow in a network cover a wide range of topics with many related research problems. We organize the research issues into the following chapters of this thesis:

Chapter 2 introduces some necessary background knowledge for the thesis. It briefly reviews three research topics, including network analysis, predictive models for information diffusion and efforts on transparent data mining made by other researchers.

Starting with Chapter 3, we introduce several original contributions. Chapter 3 describes smartphone usage behaviors with a generative model. We also investigate how to apply network science to unstructured raw datasets and detect significant patterns in a proposed physician collaboration network derived from patient-physician visiting records. The methods are applicable to diverse contexts. In general, user behavior models and structural patterns of a network describe the “walker” and where she “walks”, respectively.

Chapter 4 shows two examples of feature engineering for data-driven projects. We explore the use of new geographical features from a public database for user preference prediction and build a feature vector of the chronological visiting records during a patient’s treatment. This Chapter targets both the “physical walk” in local business units and the “walk” in a referral network. Those features enable a general framework applicable to diverse contexts for recommendation and information walk prediction.

Chapter 5 digs into three predictive tasks for an information walk on the context of referral network, including the sequence of referrals in the physician collaboration network. For the problem of future direction (i.e., the next visited node) prediction, we translate it to a problem of ranking over all candidate nodes and learn latent parameters in a novel preference score for the ranking. Second, we describe the “space” of all information walks in an articulate and rigorous way to detect the possible outliers of information walk. Third, we apply machine learning models to predict the final result (e.g., treatment outcome of a patient) of an event along with an information walk.

Chapter 6 presents a general transparent framework for users, with the goal of giving users a better understanding of why they find the current information on the screen. To improve the user experience in network navigation, we design a transparent recommender system with user-controlled settings and dynamic visualization of network space. This is in contrast to some online tools (e.g., “people you may know”, “movies you may like”) that directly display a list of suggested items. An initial user study of our proof-of-concept Wikipedia pages recommender shows positive feedback (e.g., more transparency and good recommendation performance) for such enhancements.

The four Chapters (from Chapter 3 to Chapter 6) of new work follow the chronological order in the practice of building a data-driven project with network and information walk. First of all, we need to understand the context in the raw dataset. Second, we should implement feature engineering for more meaningful information. Third, the prediction of a target variable may need a framework of machine learning or statistical analysis. Last but not least, it will be better to explain the logic of the algorithm with diverse visualization of the user's feature vector space and more interactions with users. Our models/methods in those Chapters could work as independent modules for different contexts (e.g., smartphone users, referral network, Yelp, Wiki, etc.), but we can also combine them together as a complete data-driven project about information walk, or other desired target.

Finally, Chapter 7 summarizes the contributions of this thesis and discusses possible future directions of work, including several research problems proposed in this thesis.

Chapter 2

Background

Chapter 2 presents a very brief introduction to the background knowledge and related baseline works. Given the breadth of the four themes in the thesis (**patterns of user behaviors and network structure, feature engineering and entity representation, predictive models for information flow, transparent network applications**), the coverage of background knowledge below cannot be complete. Our research builds on the background models to address new problems in different contexts.

Section 2.1

Descriptive Network Models

Network science underlies each of the four aspects of the problems in the thesis, since we usually either build a network with unstructured raw data or propose a network model to address a problem.

In general, network science produces structural measures (e.g., clustering coefficient, diameter), node position measures (e.g., PageRank, eigenvector centrality) and edge weights measures (e.g., assortativity) as summary measures of a social network. Several network models (e.g., core-periphery [Borgatti and Everett, 2000], “small-world” [Amaral et al., 2000]) describe general patterns of nodes connections. Recent books [Barabási et al., 2016, Serrat, 2017] survey network analysis with a complete list of powerful methods. In addition to traditional network analysis methods, we can introduce new desired features and models depending on the goal of a project. There are also connections to the so-called “multilayer networks” [Kivelä et al., 2014] and hypernetworks [Ghoshal et al., 2009, Zlatić et al., 2009] used to model complex types of relationships (with different kinds of edges) in a set of entities.

Enlightened by the above works, we implement network analysis on several network datasets to validate significant patterns and derive new structural features. Additionally, we propose a new model, a high-level network of information flow. Moreover, we apply the idea of network analysis to a recommender system. As one of our projects in the thesis, we build a network of candidate items based on their connections with the context of Wikipedia pages, and visualize the network with several algorithmic parameters set by users. We hope the network visualization could remind users where they are in the vast space of candidate items and why the recommendation algorithm returns such a list of items.

Section 2.2

Information Flow Prediction

Our proposed predictive models target the single-track information flow. This is different from the hot topic of multi-track information diffusion which talks about a kind of explosive information sharing where a node may pass a piece of information to multiple successors rather than a node-by-node single track.

In this context, the focus of our work is related to but different from the well-investigated problem of *link prediction*. Considering network dynamics and the diffusion of information on networks, we define an *information flow* as a sequence of nodes in a network (supporting some kind of information flow or sharing dynamic) that successively receive and pass a kind of information. If available, the observed explicit node-to-node path of an information flow will provide additional data for the prediction of information flow in the future, such as the next node (i.e., “direction”) of the information flow. We make use of the baseline methods from the above related works to verify the efficacy of our proposed information flow prediction model.

Link prediction [Liben-Nowell and Kleinberg, 2007] refers to the task of predicting the next most likely links to be produced in an evolving network based on the current snapshot of the network. This is one of the most popular predictive methods for a social network that adapts to information diffusion. A typical supervised learning framework [Martínez et al., 2017] requires a target label and the corresponding feature vector for model training. Probabilistic generative models [Kashima and Abe, 2006] exploit a joint distribution of links along with related node features, while discriminative methods [Yu et al., 2007] directly model links using related features as the input for classifiers. Current works [Bourigault et al., 2016, Saito et al., 2008] mainly aim to explain the observed track of information diffusion on a social media platform with possible hidden connections between nodes.

Another related problem is finding *missing links* [Nakagawa and Shaw, 2004]. Traditional link prediction models (e.g., [Adamic and Adar, 2003] and [Liben-Nowell and Kleinberg, 2007]) usually rely only on a form of node similarity derived from network topology and generally ignore the whole (information) walk. Many past works target the problem of multitrack spreading or broadcasting in *directed acyclic graphs* (DAGs), while our proposed information walk model allows the existence of a loop. Representative works include the Independent Cascade (IC) model ([Kimura and Saito, 2006] and [Bourigault et al., 2014]), the Linear Threshold (LT) model [He et al., 2012], and probabilistic methods [Gomez-Rodriguez et al., 2011, Myers and

Leskovec, 2010]. In addition, past works do not consider observed information walks as a part of their key inputs. In contrast, we incorporate information walks using summary measures of network features in the corresponding network. Diffusion models are clearly different, and they have been introduced to the research of epidemics [Raj et al., 2012].

The idea of network navigation (e.g., [Leibon and Rockmore, 2013]) is related to a class of state transition method for an accurate recommendation. Recently, a Transition-based Factorization Machine model (TFM) (see [He et al., 2017] and [Pasricha and McAuley, 2018]) was used to predict the next state in an abstract space of items for users. In contrast to the TFM model, our proposed preference score model considers network science measures and shows the benefits of incorporating them with other metadata features.

In specific domains, several applications (most notably online shopping or search) try to predict a visit to a next "item". The general BPR model [Rendle et al., 2009] has been introduced to online shopping [Rendle et al., 2010] to serve users with personalized goods recommendations in the context of user activity logs. A common problem has been to predict the next place of work of a given employee in a labor pool using LSTM [Li et al., 2017] or a "gravity law" based approach [James et al., 2018]. In medical research, Choi [Choi et al., 2016] applied deep learning to estimate the next medication code in a course of treatment by combining codes of medical treatment and physician visiting records to obtain a comprehensive feature representation.

Section 2.3

Transparency in Applications

Section 2.2 introduces several examples of good predictive models, which will greatly improve user experience. The same is true of transparency: if a user has a better understanding of the hidden algorithm in a recommender system (or other applications), as well as some agency in changing the algorithmic parameters, arguably the user can improve her experience.

We exploit the context of Wiki browsing to implement our idea of transparency in Chapter 6. Here we briefly review several works about better user experience on Wiki. Several projects have focused on Wikipedia navigation, as relates to efficient browsing. Lamprecht [Lamprecht et al., 2015, Lamprecht et al., 2017] discussed the influences of Wikipedia navigation policies and the structure of Wikipedia pages network. Odor [Odor et al., 2018] presented the evolution of Wiki hyperlink networks to aid navigation and understanding. Another automatic tool [Sáez and Hogan, 2018] aimed to generate info-boxes for Wikipedia pages from a Wikipedia knowledge graph. In a different direction, Leibon et al. [Leibon and Rockmore, 2013] show how the Wikipedia pages around a given topic – e.g., mathematics – can support a metric and thus structure of a hyperbolic geometry and with that, enables the construction of geodesics (paths in the Wikipedia space) that optimally guide a user's viewing experience (the use case of the paper is the MathWikipedia). We also propose our new model of information flows network in Chapter 5. Figshare [Wikipedia, 2016] provided content-based embeddings learned from Wiki corpus as the navigation vectors on a 2D plane. Cartograph [Sen et al., 2017] enables the presentation of a vast map of Wikipedia pages with the embeddings learned from neural networks. The last of these differs from our proposed navigation schema would allow users to change the visualization and any underlying metric supporting the visualization. Our proposed framework in Chapter 6 also brings more transparency via user-controlled visualization.

Also related is work on semantic annotation and some applications of collaborative filtering applied to Wikipedia data. IkeWiki [Schaffert, 2006] and SweetWiki [Buffa and Gandon, 2006] made the inherent structure of a Wikipedia page accessible to users and computing machines via annotations derived from semantic methods (e.g., RDF and conceptual graphs). A visual analytics framework [De Sabbata et al., 2015] illustrated how editors could work together for a public visualization of Wikipedia data.

Researchers also have been working on diverse kinds of Wiki tools to improve knowledge transfer and user experience. Harder et al. [Harder et al., 2017] designed a new measure to model and display the degree of "verifiability" of a Wikipedia page and implemented a demo in a Chrome browser extension. A visual article development tool [Flöck et al., 2015] explored editor interaction history to deal with disagreement. Balaraman [Balaraman et al., 2018] proposed a new metric to describe the relative completeness of Wikipedia data. Gundala [Gundala and Spezzano, 2018] reported the initial progress about predicting hyperlinks between pairs of non-connected pages that are helpful for search navigation. WikiTrails [Reinhold, 2006] provides a tracking system of visited Wikipedia pages to facilitate the understanding of Wikipedia content structure. Omnipedia [Bao et al., 2012] visualized multi-language editions of the same Wikipedia page via colorful circles in different sizes based on an article alignment algorithm, but it ignored network analysis. SuggestBot [Cosley et al., 2007] proposed a link recommendation framework to match people with suitable editing tasks on Wikipedia.

Lastly, there is now a growing body of experimentation with digital interfaces for searching and exploring traditional information materials, specifically for the interaction with libraries. An interesting example of this is the Harvard Stacklife project[1], which aims to bring back to online library search the missing – and bemoaned – loss of the serendipity of browsing the stacks that occurs when going to retrieve a

[1] http://stacklife.harvard.edu

book of interest. It is in the spirit of achieving such exploratory serendipity that we present the work in Chapter 6. Another interactive graph [Leibon et al., 2018] allows users to set up the weights of link structure measures and textual similarity for a 2D map of legal documents, from which researchers explore how new opinions influence the search behavior of judges and litigants and thus affect the law.

Chapter 3

Understanding the Datasets

Before building a data-driven personalized application, it is necessary to analyze the dataset we have and try to find some meaningful pattern in the dataset if there is. Therefore, we explore both individual user behavior models and global network structural patterns on real datasets. This Chapter provides general methods of understanding user behaviors and structural patterns in a network, which are of great value to a data-driven project related to a generalized network (i.e., the underlying metadata of a "society"). The significant patterns in our dataset may suggest a new target for the application. Taking information flow as an example, it is necessary to understand both individual users and the whole network before digging into a detailed task.

To model the communications between nodes (e.g., users) in a social network, it is beneficial to understand the user behavioral patterns. The daily routine of mobile phone usage is a good example. Our work [An and Rockmore, 2016b] focused on the use of a Hierarchical Generative Model to explain and predict phone usage behaviors. Our user behavior model describes three important kinds of phone usages (messages, phone calls and cellular data) with three layers: (1) the state of user-phone interaction, (2) occurrence times of an activity and (3) the duration of the activity in each occurrence. We find the prediction error of the generative model to be the smallest in comparison with several baseline methods. Since many users stay in touch with others via mobile apps, the results suggest a new way of modeling user behavior and provide a better understanding of users. Depending on the app, connections via the app may create a network of users. Beyond the context of mobile phone usage activity, the proposed hierarchical model could serve as a template for other ways of communication between users in a social network when there is some time series related pattern (e.g., seasonal or weekly change in the size of information flow).

A well-organized social network may contribute to a working society, depending on its ability for information and resources to flow efficiently within itself. Therefore, the application of network science matters in terms of mining structural patterns of a network, especially when we are going to build a data-driven application based on social network data. As an example, our paper [An et al., 2018a] analyzes the U.S. Patient Referral Network and various subnetworks in 2009-2015. In these networks, two physicians are linked if a patient encounters both of them within a specified time interval. We find power law distributions as well as a core-periphery structure in most of the state-level networks. We also discover the so-called small-world structure and the "gravity law" that often exists in some large-scale economic networks. Some physicians play the role of hubs for interstate referrals. The patterns in the referral network illustrate the potential for using network analysis to provide new insights into the healthcare system. The network models applied in the paper [An et al., 2018a] could be extended to a wider range of contexts for more significant patterns.

Section 3.1

Phone Usage Behavior

The work in this Section has already appeared in the refereed publication [An and Rockmore, 2016b].

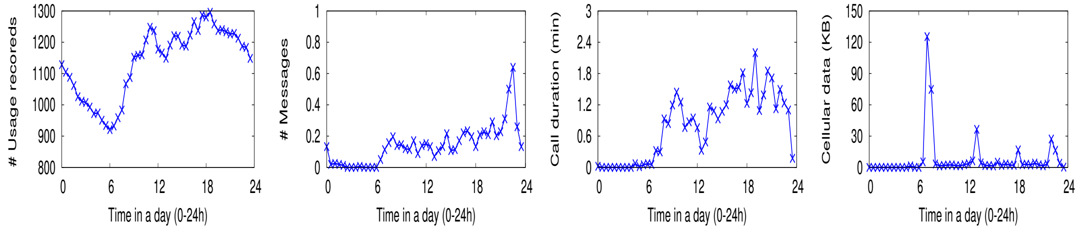
Smartphone applications can record data from diverse sensors and components in the device, such as an accelerometer or Bluetooth. The popularity of smartphone usage makes it possible to collect large amounts of sensory data, from which it is possible to predict user behavior. Examples include the prediction of mobile application usage [Shin et al., 2012] and daily geographic routines between different locations [Farrahi and Gatica-Perez, 2010]. Some usage records, such as phone calls, alarms, and GPS are clearly related to human behavior. Those behaviors are in turn correlated with a person's daily routine. Communication behaviors (e.g., sent messages, phone calls, cellular data) are of particular interest since they are related to the business model of the data carriers and service providers. With an in-depth knowledge of user behavior, or a good predictive algorithm for such behavior, service providers can offer plans personalized for the usage pattern. For example, a better pop-up message service of a mobile application would not disturb certain ongoing events. The question we ask here is, can we extract and predict the patterns making up a daily routine from a large number of phone records?

Traditional methods for solving "prediction problems" (e.g., linear regression) treat categorical (but still numerical) features as numerical values without an explanation of the result. In this work, we take for granted that daily routine is the basic and intrinsic foundation of behavior prediction. The traditional prediction methods do not organize all features in the natural way as they are generated in daily life. For example, binary output or values in $[0, 1]$ of logistic regression might not reflect the accurate value of usage behavior in a wide range (even as the amount of data increases dramatically).

Current work on mobile device usage mining does not give a direct route to predicting user behaviors with an understanding of daily routine. In the literature, there is primarily a focus on the prediction of the next event in the near future, such as the next used application or next geographic position or route [Farrahi and Gatica-Perez, 2010, Liao et al., 2013]. Some methods [Xu et al., 2013] require external information, such as a large community of people to find similar user profiles. In our method, the prediction for a user does not rely on outside datasets so that the lack of similar user groups will not weaken the result.

Given the history user behavior records, we would like to predict the total amount of some phone usage behaviors (e.g., the number of messages) in a period of time (e.g., 30 days). Our method addresses the phone usage prediction problem with a hierarchical generative model of three levels. The first one is state transition. Human circadian rhythm affects the frequency of activity on phones. We divide a day into smaller time slots and classify them into different states, such as sleep, passive and active. The second parameter is the number of occurrences for some activity in each time slot. The third is the duration of each occurrence. Once we learn the necessary parameters in those three levels, a generative model will simulate the user's behaviors on a phone in order to make a prediction. We apply our generative model to a dataset from Android Device Analyzer [Wagner et al., 2014a]. The results show that our generative model performs well with the smallest error among several methods. Briefly, the contributions of this model are:

- A hierarchical generative model to predict phone usage behaviors, which extends the current focus on event intervals to event duration.



(a) All usages. (b) Messages. (c) Phone call. (d) Cellular data.

Figure 3.1: The behavioral pattern of a user at time slots in a day.

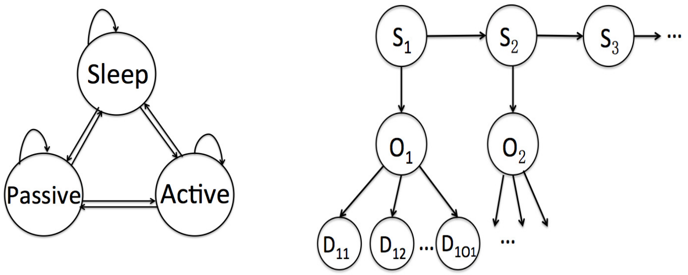


Figure 3.2: Generative Model includes three levels, which are S states of time slots, O times of occurrences, duration D of one occurrence. The state changes in time order through sleep, passive, and active.

- Demonstration of the effectiveness of the generative model in large practical datasets of sensory data (given enough states in the Markov model to describe state-to-state transition).
- Enabling better personalized mobile service based on user behavior.
- Exploration of the best setting of parameters in the generative model with control experiments.

3.1.1. Problem Definition

Consider a set of users $u_1, u_2, ..., u_n$, with time-stamp sensory records on phones. The records will reflect the change of device setting or the user-phone interaction. Each record can be described as $R = (D, T, U, A)$. D is the day when the usage happens. T is the time of the day. U represents the usage and system settings of a certain sensor or component, such as messages, screen locks, network connections. A means the attributes and values of U. The paper [Wagner et al., 2014b] introduces possible values of U and corresponding A. One sample record of alarm volume is "2012-01-21T08:16:50.533+1300;audio—volume—alarm;7".

Our goal is to predict the phone usage in the future t days based on the given records of $u_1, u_2, ..., u_n$. Though we can predict other sensory data similarly, we focus on three kinds of behaviors: the number of sent messages M, the total duration of phone call (call in and out) C and the size of cellular data (rx and tx) D. They occupy the main part of bills on phone communication and represent daily usage on the phone. They reflect a user's connectivity in a social network. To predict phone usages using a generative model, we should define and learn states of user-phone interaction, then explore the distribution of messages M, phone call C and cellular data D in each state based on some pattern.

3.1.2. Behavior Prediction

Patterns in Phone Usage Behavior. Dividing a day into 48 even time slots, we count the average number of event occurrences over many days. Figure 3.1 shows the number of records, the number of received and sent messages, the total phone call duration, and the size of cellular data. The user interacts with phones more actively in some time slots. For example, Figure 3.1(a) shows that the device will collect more records from 9am to 10pm. At night, especially from midnight to 7am, the number of records decreases sharply. When the user is sleeping, the device might collect records about itself rather than the user (e.g., records about networking). The different and uneven distributions for all users suggest a differentiation of states. The transition of phone-usage states in a day becomes the first layer in the following generative model. We define three states since in Figure 3.1(b), the number of messages can be zero, small or large (and bursty), which correspond to the sleep, passive, active states in our model. One could imagine a finer distinction with more states.

Algorithm 1 Generate phone usage behaviors as prediction.

```
Input: $N$ length of prediction days, $Slot\_num$ number of slots in a day, $S_0$ is the initial state of the first slot in the first day, $T$ denotes the transition matrix of three states (sleep, passive, active), $\Psi_O$ is the distribution of occurrence times in a certain state, and $\Psi_D$ is the distribution of duration in a certain state.
Output: $O_{ij}$ times of occurrence in the $j$th slot of the $i$th day, $D_{ijk}$ the $k$th duration of the usage behavior in the $j$th slot of the $i$th day.
  $i \leftarrow 0$
  while $i < N$ do
    $i \leftarrow i + 1$
    $j \leftarrow 0$
    while $j < Slot\_num$ do
      $j \leftarrow j + 1$
      if $S_j \neq$ sleep then
        $O_{ij} \leftarrow$ RAND($\Psi_O(S_j)$)
        for each $k$th occurrence in $O_{ij}$ do
          $D_{ijk} \leftarrow$ RAND($\Psi_D(S_j)$)
        end for
      end if
      $state\_seed \leftarrow$ RAND()
      ($T\_sleep$, $T\_passive$) $\leftarrow$ transition($T$, $S_j$)
      $S_{next} \leftarrow$ decide($T\_sleep$, $T\_passive$, $state\_seed$)
    end while
  end while
```

Hierarchical Generative Model. Figure 3.2 illustrates the process of generating a kind of event (e.g., phone call). A day is divided into several time slots. The state of phone usage S_i of the ith time slot can be described with one of the three states, which are sleep, passive, and active. Given the state of phone usage, the model generates the times of occurrence O_i of the event in each slot based on a random seed of a probability distribution learned from training set. Then the model generates the duration D_{ij} in ith time slot for jth occurrence of the event. After the generation in a time slot, a transition matrix between usage states will lead the process to the next time slot with the usage state S_2. The model can generate a prediction for any length of time in the future. Different values of parameters give a range of diversity in user behaviors.

Algorithm 1 shows how to generate the usage records for a given behavior. Some instantaneous behaviors such as messages only need to generate occurrence times. A transition matrix of states determines the current state with a uniform random variable in the interval $[0, 1]$. In each state S, assume the time of occurrences is in the distribution of $\Psi_O(S)$. We can generate a random value with $\Psi_O(S)$ as the prediction. Finally, for each occurrence, with the distribution of duration $\Psi_D(S)$ under a certain state, a generated random value represents the duration for each occurrence in a similar way. We can sum up the results of duration for all days in the future to get the prediction results. The model simulates user behaviors in the three layers, where parameters in statistical distributions determine the expectation of output prediction. Section 3.1.2 will introduce how to learn the input transition matrix and parameters of several distributions.

State transition. The distribution of usage behaviors in Figure 3.1 suggests three states of user-phone interaction. "Sleep" means that the user does not use the phone but it remains on. "Passive" and "Active" correspond to the normal and peak periods of occurrence times and duration.

Algorithm 2 Classify time slots into three states.

```
Input: Set of records in all slots $TS$= $\{TS_1, TS_2...TS_n\}$, State = {Sleep, Passive, Active},
  Set of behaviors to predict $U$ = {messages, phone call, cellular data}
Output: $TS_i.state$ state of each slot.
  for each time slot $TS_i$ in $TS$ do
    $f_i$ ← count_feature($TS_i$)
  end for
  for each feature $f_i$ do
    if no occurrence of any event in $U$ then
      $TS_i.state$ ← sleep
    end if
  end for
  Undefined ← {$TS_i$ — $TS_i.state \neq$ sleep}
  k-means-cluster(Undefined)
```

Given all usage records in the training set, Algorithm 2 shows the way to identify states for all time slots. For all usage records in each time slot, we count the user-phone usage related behaviors to build a feature vector. The elements in the feature vector are in Table 3.1. If during a slot there is no occurrence of any event which we aim to predict, the state is "Sleep". Then for the other time slots, apply the k-means algorithm to cluster them into two classes, "Passive" and "Active". The distance metric used in k-means is Manhattan Distance.

Once we know the states of all time slots in the training set, we can compute the transition probabilities based on the frequency of change between two neighbors. A probability will be set as (tiny) α if it is zero to avoid endless self-looping in some state. We treat a day as four time periods, night (0am-6am), morning (6am-noon), afternoon (noon-6pm), and evening (6pm-0am). For each time period, a 3×3 transition matrix describes the threshold of transition between any two states, so a uniform random seed can determine the next state. An even distribution among three states can generate the initial state S_0 in Algorithm 1.

Occurrence times distribution. A series of occurrence times $O_1, O_2, ..., O_n$ in all time slots within the same usage state suggests a distribution. Theoretically, it matches the definition of Poisson distribution as Equation 3.1. By solving the MLE problem with likelihood function as Equation 3.1, we find the Poisson parameter λ for each state, which works as the Ψ_O in Algorithm 1.

$$P(X = O) = \frac{\lambda^O e^{-\lambda}}{O!} \quad L(\lambda) = \prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{O_i}}{O_i!} \tag{3.1}$$

$$P(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - u)^2}{2\sigma^2}} \tag{3.2}$$

Duration distribution. Duration of an occurrence for an event corresponds to the amount of resources it gets, such as data flow in a period of connection to a cellular network. According to the definition of Poisson distribution, the intervals of consecutive events should obey an exponential distribution. However, we target the duration of some event rather than the gap between two neighboring events. Therefore, we do not model the length of an event with exponential distribution. Since the duration should always be positive and the dataset shows that people tend to have short conversions, we choose a log-normal distribution to describe the distribution of durations. Given the probability density function Equation (3.2) of the log-normal distribution and the observed duration list $D_1, D_2, ..., D_n$, by MLE method we will know μ and σ, which determine the Ψ_D in Algorithm 1. If the durations in the training set do not obey a log-normal distribution, we limit the errors by simple methods such as the lastest value before the timestamp of observation. Moreover, to avoid prediction of an infinite duration we set limits on the maximum duration in each occurrence and for the sum of duration each day.

3.1.3. Experiment Results

Datasets. The original datasets are recorded by the application called Device Analyzer [Wagner et al., 2014a, Wagner et al., 2014b], which contains records of more than $10,000$ users. We filter the datasets with several thresholds, such as the length of period when the app is recording data, the average number of records related to messages, phone calls and networking per month. Usage records over two months make it possible for predictions month to month. We omit the datasets in which some records miss the accurate time and date by formatting check. We end up with 107 users whose records are complete and correct for a period of time. The records reflect the attributes of start/shut-down, power, air mode, audio, CPU, video, image, memory card, phone, screen, time, messages, wifi, networking, Bluetooth, root, contacts, location, alarm and other sensors. The shortest length of records is 59 days and the longest is 632 days. The total number of all kinds of usage records (e.g., sensory data, device settings, and communication) varies from $587,817$ to $22,906,385$ for a user.

We select those attributes which are caused by the user or change the user's behavior to build feature lists, then classify all time slots with the values in each feature list as Table 3.1. More kinds of sensory data can be added to the list if the sensors are deployed on all users' devices. Since we cannot equip all devices with more sensitive and advanced sensors, the list only includes features that almost all devices can record. We should also realize that the other sensors, such as accelerometer and GPS module might be helpful to reflect user's behaviors.

Table 3.1: Elements in a feature list.

1	number of apps start by a user	2	sum of rx data by cellular network
3	sum of tx data by cellular network	4	times of cellular network connection
5	number of phone calls	6	total length of phone call
7	length of the time when the screen is on	8	number of screen switch (on/off)
9	number of devices found by Bluetooth	10	number of received messages
11	number of sent messages	12	number of bell rings

Performance. We compare the average prediction errors among 107 users of their sent messages, phone call duration, size of cellular data with several baselines. The datasets are divided into training sets and test sets. The size of test set for all users is 30 days, which means we predict the usage behaviors for about a month in the future. Each day is divided into 48 time slots, which means a time slot lasts half an hour. The experiments are running offline in a server, so we do not focus on time complexity. All experiments about generative models are executed multiple times.

(1) *Naive.* Treat the latest records in training set as the prediction.

(2) *Average.* Average the value in previous months as the prediction.

(3) *Drift method.* Predict with the first and the latest observations.

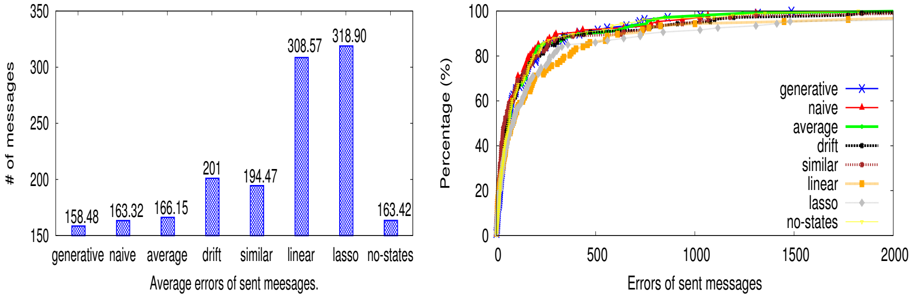


Figure 3.3: Errors and CDF curves of #sent messages.

(4) *Similar One.* Choose the value of the past month which has the smallest feature list distance to the test case.

(5) *Simple Linear Regression.* Only consider the values of target usage behaviors.

(6) *Lasso Algorithm.* Treat the feature list as the input.

(7) *No-States* (CNPP). Run the generative model with only two states.

In Figure 3.3, the average errors of 107 users in the prediction of number of sent messages (in the future 30 days) vary from 158.48 to 318.90. The generative model performs the best while the Lasso method is the worst. The naive method, average method and no-states method have almost the same errors about 165 messages/month. The generative model is slightly better. This illustrates the utility of the generative model and the distinguished user-phone interaction states. The generative model can avoid the rare peak in message usage, so it is better than naive or average. The three states facilitate the model to describe the distribution of occurrences and durations more accurately. In terms of the other methods, they ignore the inner relationship between user states and features and the target message behavior, so they have much larger error than the generative model. On the right, CDF curves show the distribution of errors among all users. The generative model stays with several other methods in the beginning part and comes to the top when the horizontal variable reaches 500. For the generative model, more than 65% of

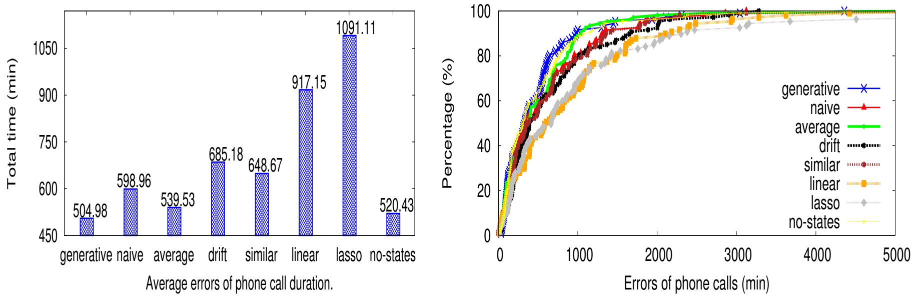


Figure 3.4: Errors and CDF curves of phone call duration.

users have errors less than 100 messages/month, about 80% errors are less than 200 messages/month, it has the smallest maximum errors among all methods. All users in the dataset frequently use their devices, some of them may change their routine or have unexpected burst, so all models face instances of large errors. However, for the majority of users, the generative model performs well and avoids more huge error cases than other methods.

In Figure 3.4, the average errors of 107 users' phone call durations in the future 30 days range from 504.9 to 1091.11 (min). The generative model has the smallest error. The second and third best methods are "no-states" and "average". The generative model and its no-states version allocate a tiny possibility for the rarely observed data so they are not sensitive to long-time phone call communication. Thus they work better with the average method than the others. In the right hand figure of CDF curves, the generative model is at the top among all methods with more than 60% of users' estimations having errors less than 500 min/month and 80% of users having errors less than 640 min/month. Though the log-normal distribution that we choose to describe duration distribution needs to be improved, the generative model still performs better than other methods. It proves the advantages of the generative model, including a seemingly deeper identification of daily routine and robustness with seldom usage patterns.

In Figure 3.5, the predicted average of 107 users' cellular data size in the future 30 days changes from 1605 to 2105 (MB). The generative model still has the smallest

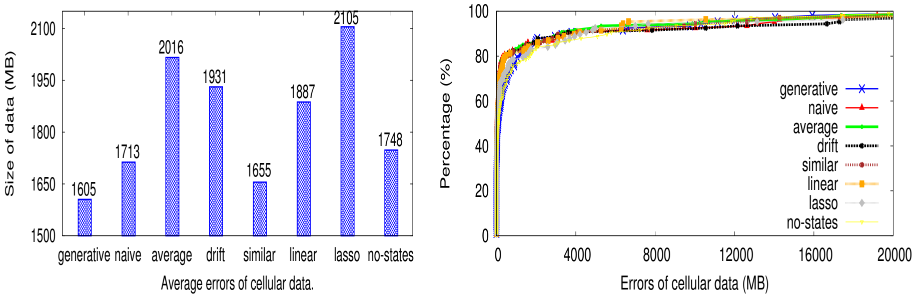


Figure 3.5: Errors and CDF curves of cellular data size.

error. No-states, naive, similar methods are comparatively better than others. For the generative model, though more than 60% cases have errors less than 500 MB/month and about 80% cases have errors less than 650 MB/month, about 9% of cases show errors more than 1000MB. The large errors in data flow prediction exist in all methods, because some users will receive or send huge data packets in the future without a similar history, and the size of data flow is unlimited. As a result, the generative model stays on the top in the beginning, but it mixes with other curves at the end due to the rare cases with huge data errors. Though the generative model is not the best in the beginning, with less huge-error cases, it still beats the other methods in terms of average error. Moreover, the log-normal distribution might not fit well with the rare unexpected large dataflow since it estimates the large data flow with a small probability. Cellular data flow usually happens in an environment of movement or a place without Wi-Fi. The generative model lacks more detail about user routine, which limits the performance of cellular data prediction. However, in total the generative model is better than the other methods due to the hierarchical framework.

Sent messages, phone call duration and cellular data size represent three types of usage behaviors respectively. They are no-duration behavior such as sent messages, occurrence with a limited duration such as phone calls, occurrence with an unlimited duration such as cellular data. For the third type of usage behavior, we should find more ways to limit the prediction and model the unexpected burst.

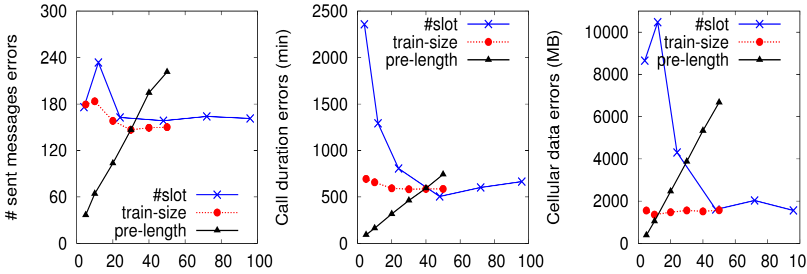


Figure 3.6: Effects of three parameters on three usage behaviors. Left: number of messages. Middle: call duration. Right: cellular data. For each behavior, we tune three parameters: number of time slots in a day, length of training set and length of prediction in the future.

Effect of Factors. Figure 3.6 illustrates the effect of three parameters: number of time slots in a day, length of training set and length of prediction in the future. Since they have a similar range between 0 and 100, we combine them in the horizontal axis. The vertical axis shows errors of certain usage behavior. To get a different length of a slot (e.g., 48 slots of 30 mins in a day), we adjust the number of time slots with different values from 4 to 96. We fix 30 days as the size of test set and the rest of the records are training sets. The size of training sets for each user varies from 5 to 50 days. We fix records of 30 days as the test sets with 48 time slots in a day. To see the effect of prediction length, we put aside the records in the lastest 50 days as the test sets, and run the generative model using 48 time slots, then compare the errors in the first 5, 10, ..., 50 days of test sets. When the number of time slots increases, the errors of three usage behaviors decrease at first and then fluctuates after 48. If we divide a day into a few slots, the differences between sleep, passive and active states diminish. In contrast, too many time slots will result in a sparse feature list. Like time slots, when the training sets are small, the generative model lacks enough records. When the training sets become large, perhaps the previous usage pattern is not inherited by the test sets. So medium-size training sets perform well. The errors are approximately proportional to the increase of prediction length from 5 to 50, since the model accumulates errors day by day.

3.1.4. Conclusion

In this project, we illustrate daily patterns in mobile usage and sensory records and build a hierarchical generative model with multiple user-phone interaction states. The model predicts three usage behaviors with acceptable errors for the majority of users. Given the simple distributions of occurrence and duration, our improved accuracy over various baselines demonstrates the value of our revised model on large real datasets. We explore the effects of several parameters in the prediction process to guide a suitable choice of timeslots as well as the size of training and test sets. The prediction of phone usage behavior by a generative model may be useful for personalized service. The model could explain diverse user behaviors with suitable distributions in different layers, so it is not limited to the context of mobile usage records.

There are several natural avenues for future work. Our model is straightforward and it would be of interest to integrate more advanced inference methods (e.g., variational inference). Latent features for usage behaviors should be explored, including the possibility of using more meaningful sensors (e.g., GPS) even though their records are not complete. Moreover, the detection of unexpected huge bursts in some usage behaviors remains a problem. The use of an additional layer or branch in the generative model could be helpful in this regard. The running time of the application which collects the sensory data is limited by energy, so the prediction with sparse data is a practical issue. A suitable sampling method may be the critical point for this.

Section 3.2

An Example of Network Analysis

The work in this Section has already appeared in the refereed publication [An et al., 2018a]. It shows an example of network analysis before we build a network-related application.

A well-designed healthcare system is a key component of a working society, and the ability for information and resources to flow efficiently in such a system is crucial to its efficacy. Referrals are one of the most common and important forms of primary-specialty care communication. The existence of a shared patient relationship between physicians likely means there are professional, information-sharing relationships [Barnett et al., 2011, Uddin et al., 2013]. Physicians decide to refer patients to other physicians in other hospitals for a multitude of reasons ranging from the need for specialization to addressing problems of overcrowding. A physician's decision to refer (or not refer) a patient is important in determining the cost and quality of care [Barnett et al., 2012b].

The referral of a patient by physician A to physician B is naturally represented as a directed edge from a network node labeled A to a node labeled B, forming a *directed network* (possibly weighted by the number of such referrals) [Barnett et al., 2011]. Here we analyze the structure of a patient referral network and in this context introduce a number of novel concepts from the network science and social networks fields. Drawing together methods from both of these two growing but surprisingly distinct fields is an important and novel feature of this work. We hope it will catalyze their use in healthcare related networks.

Prior studies provide little guidance about the network structure of effective healthcare collaboration. I.e., they do not state clearly what types of structures may be more conducive for the administration of effective healthcare. Nor have they prescribed how individual healthcare professionals should develop relationships over time for better outcomes [Uddin et al., 2013].

Another overlooked issue is the impact of artificially imposed boundary definitions on healthcare networks. Several works [Lee et al., 2011, Barnett et al., 2012a] target hospital level patient referral networks, but their findings are not validated in a larger network (e.g., a state level). We are uniquely positioned to investigate boundary effects given that our data covers the complete US referral network. This allows us to assess the degree that analyses of geographically-defined sub-networks (e.g., state networks) and derived structural assessments are sensitive to the definition of the boundary and thus may distort the relationship of the network definition to important healthcare related variables.

We analyze the structure of patient referral networks at both national and state levels. We evaluate both macro (global) and micro (local configuration or actor specific) network features, describe the network in static and dynamic terms, and test against and for simple generative models such as the random network, the small-world, and power-law network, while also measuring the degree to which structural phenomena such as high core-periphery tendency are evident in the referral network.

3.2.1. Materials and Methodology

Table 3.2: Dataset size by year.

Year	2009	2010	2011	2012	2013	2014	2015
#Records	50.3M	52.2M	54.0M	54.9M	55.1M	55.8M	34.9M
#Physicians	890k	922k	956k	988k	1.02M	1.04M	961k

Data. We used the CMS patient referral data set [CMS,] to form a physician (a subset of those physicians who accept medicaid in U.S.) network of the U.S. healthcare system. Datasets are available for the years 2009–2015, measuring the number of patients encountered by one physician and then the other physician within 30-, 60-, 90-, and 180-day interval per year, so the referrals are derived with a threshold (e.g., 30-day) over a given year. Sharing (referral) occurs when the same patient is recorded as having been treated by two different physicians in a given time period. The dates of treatment are timestamps. In this project, we choose the 30-day interval referral dataset, because it judges the existence of direct referrals between two physicians with the most stringent criteria. The temporal proximity defines a "referral". The referral dataset includes two parts, the IDs of the two physicians in a referral and the attributes of the physicians. Physicians are listed according to National Provider Identification (NPI) number. (There are $4,332,951$ physicians in the NPI dataset.) In addition, the National Bureau of Economic Research "National Provider Identification number by state" data was used to attribute each physician in each year to a state based on their NPI. Some physicians are registered in several states. We label a physician according to the state in which the physician makes the most referrals.

Table 3.2 shows the number of 30-day-interval referral records for the years 2009-2015. Notice that there are fewer referrals in 2015 due to the fact that data was only obtained for 7 months of the year (the end-date of the data is 10/1/2015 and so the last date for a first visit under which a full 60-days is available for a second visit is 7/31/2015). Accordingly, we expect a reduced average number of referrals between two physicians in 2015 compared to the earlier years.

Networks of interest. We form three kinds of networks (over a given time period): (1) The *National Patient Referral Network* includes all physicians in the US who have either made or received referrals over the period; (2) The (50) *State Patient Referral Networks* wherein for state S, the node set is all physicians who are either labeled as physicians in state S or have either made referrals to or received referrals from physicians labeled with state S over the period; The (50) *Intrastate Patient Referral Networks* is a subnetwork of the State Patient Referral Networks and requires that both physicians in a referral be labeled as in state S. The node set for state S is all physicians with NPI numbers in state S who have either made or received referrals over the given period. In network terminology the State Patient Referral Network would be called the subnetwork *induced* by the Intrastate Patient Referral Network. The three kinds of networks are nested as Figure 3.7 shows.

National Patient Referral Network

$\cup$

(Induced) State Patient Referral Network of State S

$\cup$

Intrastate State Patient Referral Network of State S

Figure 3.7: Three layers of referral networks.

For each state S. Each of these networks can be studied as simple undirected or directed networks, weighted or unweighted (wherein the weights are the number of referrals). These networks are also called *shared patient networks* [Mandl et al., 2014].

We introduce these and then describe various small scale or local network structures of interest whose prominence in the network can be tested against these models of macro-level structure. There are still a relatively small number of well-defined – or at least named – macro-level network structures. Three of interest for this project are the *random*, *small world* and *core-periphery* networks.

- Erdős-Renyi (ER) random network – is the traditional null model against which network structure is measured. The ER network on a fixed number n of nodes is constructed by independently joining any two vertices with an (undirected) edge with fixed probability p [Erdős and Renyi, 1959]. It is easy to see that the

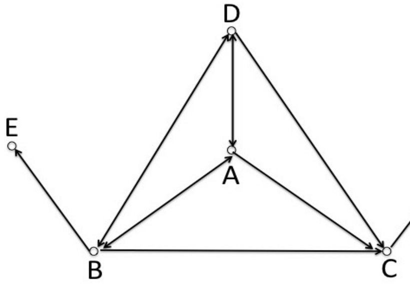


Figure 3.8: An illustrative directed network. The nodes A, B, C, D, E, and F represent different physicians. The arrow of an edge points from a referring physician to a referred physician (who accepts the patient referral).

expected degree for any vertex in such a network is $\mu = (n-1)p$, and that the degree distribution follows the binomial $B(n-1,p)$, which for large n is well approximated by a Poisson distribution with mean μ. Ascribing structure to a network derives from showing that in various important parameters it differs from the comparable ER network with probability $p = \mu/(n-1)$ where μ is the average degree of the actors in the network.

- Small world network – is defined as a network with greater than expected local connectivity and average path length smaller than expected in a comparable ER random network [Watts and Strogatz, 1998]. More rigorously, a network is a small world if it has a higher (local) *clustering coefficient* and much smaller characteristic path length than expected under the Erdós-Renyi random graph model. If the referral network is a "small world" one, it means that physicians collaborate closely on the treatment of patients.

- Core-Periphery structure – is a generative network model whose departure from the ER model is due to the network containing a "core" subset of interconnected nodes, which are also connected to a less interconnected subset of "peripheral" nodes [Yang and Leskovec, 2014]. For instance, in Figure 3.8, A, B, C, D are core nodes with connections to a collection of neighbors, while E and F are peripheral nodes.

A core-periphery structure might occur in healthcare if the practice of medicine is primarily driven by a subgroup of inter-connected physicians that impart tremendous influence. By comparison, it might be that some states have a more uniform network in which there exists no such subgroup. The "core-ness" of a node can be quantified via the assignation of a *Core-Periphery (CP)* score to each node [Rombach et al., 2014]. The range of the CP score is $[0, 1]$, with 1.0 indicating the node has the highest core quality. The extent to which a network has a generalized star structure can be captured by the Gini coefficient (cf. [Wikipedia,]) of the set of CP scores in the network. This is a standard measure of dispersion in a collection of numbers.

There are various structural metrics fundamental to describing any network (see [Newman, 2003, O'Malley, 2013] and the references therein) and so will be important for our analysis. The presence of a particular structural feature or phenomenon is ideally discovered by claiming that the observed network structure is highly unlikely to have arisen under a null model that exchanges randomness for the structural feature in question. In practice, investigators often claim that their network exhibits a certain trait by using the Erdós-Renyi (ER) network as a null model. Such a comparison risks confounding the feature in question with any other feature that is not controlled. The distributional comparisons are limited to single feature departures from the ER network. With this in mind, we describe various measures of small-scale network structure used herein and describe statistical tests of the extent of their prominence in the network beyond that expected by chance.

- Degree Statistics – in an undirected network, the *degree* of a node is the number of edges incident to the node, which is the same as the number of *neighbors* of the node, or in the referral networks, the number of distinct physicians that a given physician has referred to (shared patients with) and/or received referrals from.

In a directed network, there is an *indegree* and an *outdegree*. In Figure 3.8, for node A, indegree is 2 while outdegree is 3. The indegree of Node F is 1 and outdegree is 0. The *degree distribution* is the frequency distribution of the degrees (analogously for the in- or outdegree distribution).

Various families of degree distributions appear in the network literature. As mentioned, the undirected *Erdos-Renyi random network* produces an expected degree-distribution that is a binomial distribution with a probability parameter equal to the proportion of non-null ties. Asymptotically, as the number of referrals increases, the degree distribution will converge to Poisson. However, over the past decade or so, much attention has been paid to kinds of "heavy-tailed" distributions, especially those that follow a *power law*

$$y = Cx^{-\alpha}. \tag{3.3}$$

that are often found in data. Power laws can arise for a number of reasons (see [Mitzenmache, 2004, Newman, 2005]) and their discovery in data is but a starting point for a deeper investigation into an appropriate generative model. The measurement of a power law can be subtle. We use the estimation method in [Clauset et al., 2009] and perform calculations in *R*.

- Cluster coefficient – a *cluster coefficient* measures the extent to which nodes cluster together in a network. It is a measure taken on undirected networks of the frequency with which a "3-chain" – defined as a triple of connected nodes. The triple (A, B, C) in Figure 3.8 constructs a "triangle" when the graph is treated as undirected since any two of them are directly connected, but without an edge between A and F, the triple (A, C, F) is only a "connected" triple rather than a "triangle". *Global clustering* C_g measures the fraction of completed triangles over the entire network while *local clustering* C_l measures the average number of triads centered at a given node that are completed to triangles.

- Assortativity, Degree Distribution Correlation, Reciprocity – Various kinds of measures of connectivity can be supplemented by measures that get at *assortativity*, a general term for quantifying the degree to which "likes link to likes" (also called *homophily* in network science literature) where "like" can refer to any kind of metadata. An intrinsic kind of assortativity in any network is *degree assortativity*, often referred to as simply "assortativity". It measures the predilection of high degree nodes to attach to other high degree nodes and low degree to low degree. In directed networks there are thus four different kinds of degree assortativity: (in-, in-), (in-,out-), (out-,in-), and (out-,out-) depending on which kind of degree is taken into account. Let e_{AB} represent the weighted edge from node A to node B in Figure 3.8, A_{in} be the in-degree of node A and likewise define B_{in}. In this example, $A_{in} = 2$ and $B_{in} = 2$ and there are two possible indegree values of the two edge nodes. The (in-, in-)-assortativity can be described in terms of the Pearson correlation coefficient [1] between those two values for all edges. Since an edge from A to B does not necessarily mean there is another edge from B to A, $corr(A_{in}, B_{out})$ is not equal to $corr(A_{out}, B_{in})$. A large assortativity means physicians in the network tend to build connections to others who have similar degrees.

  Self-Degree Correlation measures the correlation of in- and outdegree on the node level [2]. For those nodes in Figure 3.8, the in-degree (e.g. A_{in} =2) might be in accordance with the out-degree (e.g A_{out}=3). While assortativity describes

[1] the covariance of the two variables divided by the product of their standard deviations

[2] measuring the relatedness between the number of referrals made with the number of referrals received

the relationship of two nodes on the same edge, self- (in- and out-) degree correlation is evaluated as the nodes' in- and outdegree.

Finally, *reciprocity* measures the pairwise relationship between two individual physicians. [3]

- Motifs – the physician-physician relationship is the core atomic structure of the referral network. Nevertheless, it makes sense – and is often useful – to attempt to identify other regularly repeating evolved substructures [O'Malley and Marsden, 2008]. Such subnetworks are called *motifs*. Two-node or dyadic motifs include null-dyads, directional dyads (e.g., Node B and E in Figure 3.8) and bidirectional or mutual dyads (e.g., Node A and D in Figure 3.8). A familiar example in an undirected network is the triangle representing the phenomenon that "a friend of your friend is your friend". In the case of directed referral networks, we are interested in exploring the landscape of small (three-node) motifs, or "triads". In a directed network there are 16 distinct kinds of triads (cf. Figure 3.12). Some researchers name them by the number of mutual, asymmetric and null dyads [Faust, 2010]. We describe the distribution of the 16 triads across the physician network and use factor analysis to group the triad types into categories that can be represented more parsimoniously in regression models.

3.2.2. Results: Network Statistics

Network Models. **Core-Periphery Structure.** We compute Core-Periphery scores and derive stats. Figure 3.9 gives an example of the CP (core-periphery) score distribution for the intra-state networks for states of DE, LA and CA in 2009. These states were picked because they have the minimum, median, and maximum of the

[3] the correlation of #referrals from A to B and B to A, where physicians A and B are connected with bidirectional edges in the referral network. It reflects the extent of quid pro quo in patient referrals between two physicians.

Gini coefficients for the CP scores in 2009. Recall that a large Gini coefficient of the CP scores implies the network has a strong Core-Periphery structure: there are a small number of nodes have a large CP score (close to 1.0) implying close proximity to the core, while the remaining nodes are in the periphery with a lower CP score.

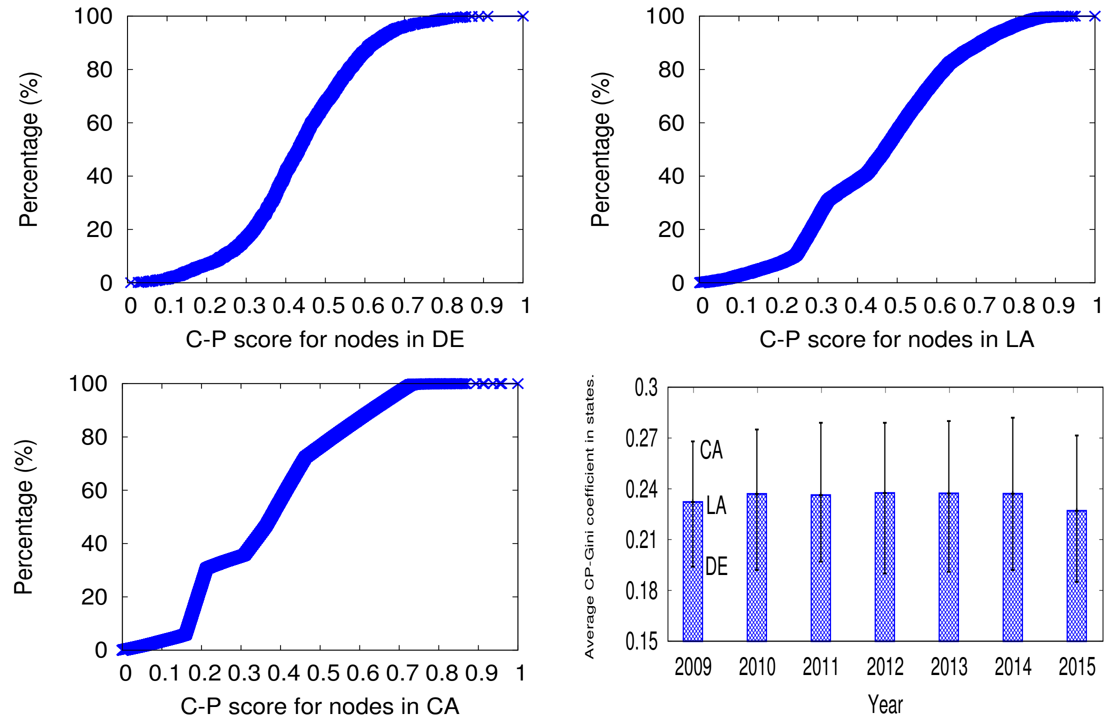


Figure 3.9: Counterclockwise from upper right: C-P score distribution of LA, DE, and CA (minimu, median, maximum), in 2009 and the distribution of Gini coefficients of C-P score among the 50 states over 2009-2015.

The uneven distribution of C-P scores suggests a strong Core-Periphery structure in these state networks. Strong Core-Periphery structure is a trait seen generally across all of the state-level networks.

Degree-, clustering-, and connectivity-related statistics. **Degree Distributions and Power Laws.** We computed the in- and outdegree distributions for both the national network and the fifty intrastate networks. The nearly zero p-value of the goodness of fit test against the null hypothesis rejects that the degree distribution is Poisson. Furthermore, the clear difference in terms of clustering coefficient in Table A.1 contributes to a rejection of the Erdós-Renyi random graph model for the data.

We next check for a power law. The intuition for considering a power law comes from a familiar generative model. For networks (cf., [Mitzenmache, 2004, Newman, 2005]): the so-called "rich get richer" process, this is wherein nodes acquire new connections at random but in proportion to their current number of connections. It is plausible that there are groups of physicians (e.g., certain types of specialists) that receive and possibly make many more referrals than others and furthermore that physicians accrue new ties in proportion to their existing number of ties. Reputation spread may also manifest as a power law. In contrast, if physicians with many referrals are less likely to accept new referrals (e.g., they stop taking new patients) and are content with their existing set of "partner physicians" for referrals, the degree distribution would be expected to be more uniform than depicted by a power law.

In a log-log plot, a power law will appear as a (roughly) straight line. The lefthand of Figure 3.10 shows the power law fitting figure for the 2015 Delaware Intrastate Referral Network. The straight line of the log of Delaware's (unweighted) degree distribution matches the form implied under a power law. The righthand side shows the distribution of the p-value statistic for testing the null hypothesis that the distribution in the network is a power law in the outdegree using the national 2012 data as an example.

The data in Table A.1 suggest that the outdegree distributions seem to have a stronger tendency toward power law than indegree. Herein we find the number of states with a p-value ≥ 0.05. Because a physician does not control who refers patients to them, the number of distinct physicians sending patients may exceed the proportional growth. This is supported by the observation that the indegree distribution has a greater spread than outdegree. Alternatively, the departure of the

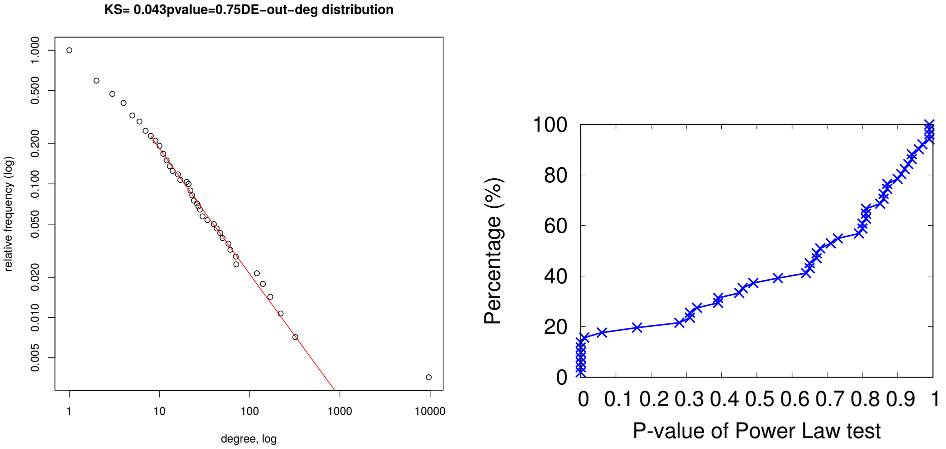


Figure 3.10: Log plot of the out-degree and frequency in DE, 2015. P-value distribution of out-degree Power Law test in 2012 for all states.

indegree distribution from a power law might be due to certain specialist physicians being absorbing nodes in the sense that they are the last step in the patient's care (e.g., a sub-specialist).

Assortativity. Table A.1 displays the average correlation coefficient between two degree values on edges of the 50 state induced referral networks. Given the directed nature of the networks, three kinds of degree assortativity can be measured. We find (in-,in-) and (out-,out-) degree correlations exhibit mildly **negative** assortativity, which means patient referral has a small tendency to occur between physicians who possess different levels of indegree or different levels of outdegree. The significance of the assortativity values against a null hypothesis of no assortativity is tested under an ER null network by using the fact that the asymptotic standard error of $0.5 \log((1 - r)/(1+r))$ is $SE = (n-3)^{-1/2} = 3.35 - 9.78 * 10^{-4}$, where r denotes the given Pearson correlation coefficient of the respective degree frequencies and n is the number of physicians in the network. Because the assortativity values are far from 0, it is clear that assortativity is significantly different from 0 in all cases.

Correlation of in-degree and out-degree. Table A.1 shows the measurement of correlations between indegree and outdegree on the same physician in several years. Since the correlation coefficients in all states are very close to 1.0, only average values are reported. The results imply that physicians who receive a lot of referrals also make a lot of referrals. The correlation may be inflated due to the fact that specialty is not controlled for and past research [Barnett et al., 2012a] has shown that degree varies substantially between specialties; if the correlation was measured within physician-type the correlation would likely be lower.

Reciprocity. If we consider the weight on edges in a directed network, Table A.1 shows the R-squared value and correlation coefficient of w_{ij} and w_{ji}. The bidirectional weights have strong correlations in different years. Reciprocity reflects the professional relationship between physicians. The observations support the idea that physicians refer patients back to the referring physician once the specialty appointment is complete or distinct patients see the physician dyad members in opposite orders. Either way, high reciprocity reflects stable collaboration.

Clustering coefficient. Figure 3.11 illustrates both global and local clustering coefficients of states in several years. The error bars show the range of the coefficient values.

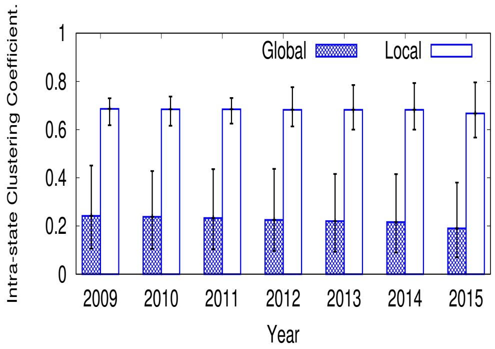


Figure 3.11: Clustering coefficients of state network in 2009-2015

Table A.1 shows the clustering coefficient in the whole national referral network. The local clustering coefficient is much larger than the global one, reflecting a positive correlation between geographic closeness and network flow. The expected local clustering coefficient in an Erdos-Renyi model [Erdós and Renyi, 1959] $p = \mu/(n-1)$ is much smaller than the measured results. Taken together with the above discussion, we conclude that the patient referral networks have small world character, so we know that those physicians in the network closely collaborate on treatment.

Motif analysis. Network "motifs" are commonly recurring small patterns of connectivity, often thought of as a network's "building blocks" [Milo et al., 2002]. Dyadic motifs are the simplest in structure having just two nodes and in a directed binary-valued network only a few possible states. If the motifs do not distinguish between the edge from physician A to B and that from B to A, there are only three dyadic patterns: no edge, one directional edge and bidirectional edges. While the no-edge case is dominant in terms of frequency, the fraction [4] of directional dyads and bidirectional dyads is around 24:76, implying a very high-level of reciprocity is present in the network. As a part of the exploration of patterns in patient referral networks we engaged in exploratory analysis to discover what kinds of triads in our directed networks are most prevalent. Figure 3.12 illustrates the 16 possible triads.

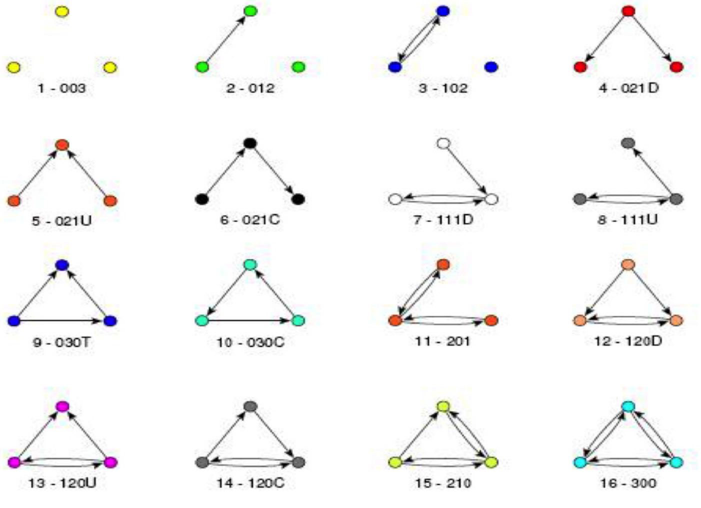


Figure 3.12: 16 kinds of triads.

Table 3.3 displays the Monte-Carlo estimated frequency of the various triad structures (i.e., randomly sample node 3-tuples and record the connectivity structure) over 2009-2015 in the national network (a Monte Carlo calculation of 10^8 random

[4] generated by Monte-Carlo sampling

draws was used because complete enumeration is infeasible). The completely disconnected triad (Triad 1 in Figure 3.12) is far and away the most prevalent and we do not record its number. The remaining 15 kinds of triads break up naturally in terms of order of magnitude of frequency into 7 groups: (1) Triads 2 and 3: Two physicians share patients in one or two directions; (2) Triad 11: a physician shares patients with two physicians mutually; (3) Triads 7 and 8: a physician shares patients with one physician mutually and with another physician in only one direction; (4) Triads 5,6,15, and 16: loose connections and close connections between three physicians; (5) Triads 12, 13, 14: a pair of mutually connected physicians with the third physician whose degree is two; (6) Triad 9: a triple that follows transitivity (if A refers a patient to B and B refers a patient to C the chance that A referred a patient to C is substantially greater than otherwise) and lastly (7) Triad 10 without transitivity.

Since the referral records do not contain patient ID, we cannot track the same patient and analyze the referral sequence. The rank order remains roughly the same over each year, suggesting that the structure of the network is stable in this regard. Triad 2 and Triad 3 are the two most popular triad patterns in the whole referral network, accounting for the majority of the triads in the state networks. These convey two of the most elementary care patterns. Under Triad 2, a patient encounters physician A followed by physician B and then is done. Under Triad 3 the patient emulates the care pattern of Triad 2 but then returns to see physician A again. The frequency distribution suggests that the network contains regions of high density, or even cliques, since some triads with more edges (T15 and T16), representing more complex care patterns within reciprocated referrals between 2 or 3 physicians, occur more frequently than triads with fewer edges (T9, T10, T12, T13, T14).

Table 3.3: Triad frequency for the U.S. national referral network.

ID	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2009	23433902	76245745	188	4096	5113	56061	28650	66	1	222166	171	127	157	1484	2073
2010	23747795	75929710	206	4426	5321	58748	28634	49	4	221342	176	117	141	1427	1904
2011	23865204	75802642	175	4999	5720	62887	29448	56	0	225125	166	124	144	1342	1968
2012	23892310	75764994	167	4989	5811	64475	30656	63	1	233164	163	99	127	1264	1717
2013	24202517	75439104	180	5648	6524	68943	31777	51	5	242030	155	114	125	1185	1642
2014	24405405	75233266	201	5803	6571	69167	32792	51	4	243538	174	104	109	1210	1605
2015	25421893	74265148	147	5160	6326	59787	31480	52	2	207622	140	90	86	948	1119

Table 3.4: The nearest states to centroids of K-means.

Year\#cluster	K=2	K=3	K=4	K=5
2009	ME MA	LA NC SD	NC OR SD TX	KY MA OR SD TX
2010	ME NC	LA PA SD	LA OR PA SD	LA ME OR PA SD
2011	ME MD	IL PA SD	LA NM PA SD	LA MT NM PA SD
2012	ME MA	ME PA TN	ME PA SD TN	LA ME MT PA TN
2013	ME MA	ME MI TN	MI MT OR TN	MI MT NE OR TN
2014	ME MA	ME MI MT	ME MI MT TN	ME MI MT NE TN

We may include the relative frequency measures for two of the three groups of triads allowing more flexibility in regards to including other predictors, interaction variables, and transformed predictors.

Diversity among states. From hereon, we discard the data in 2015 since the period of observation is not complete and many healthcare attributes are not available in 2015.

We apply the K-means clustering algorithm to the 50 feature vectors defined by the state-level network measures [An et al., 2018a], some of which are introduced in Section 3.2.1 as well. Figure 3.13 is a 2-d visualization, produced via multidimensional scaling (MDS). The red and blue coloring of the nodes represents the outcome of applying K-means with two clusters, for which the centroids are MA (red) and ME (blue), respectively. We find that the cluster represented by MA generally includes the states which have more physicians or larger population than those of the cluster represented by MA. Table 3.4 shows the centroid states of each cluster for $K = 2, 3, 4, 5$.

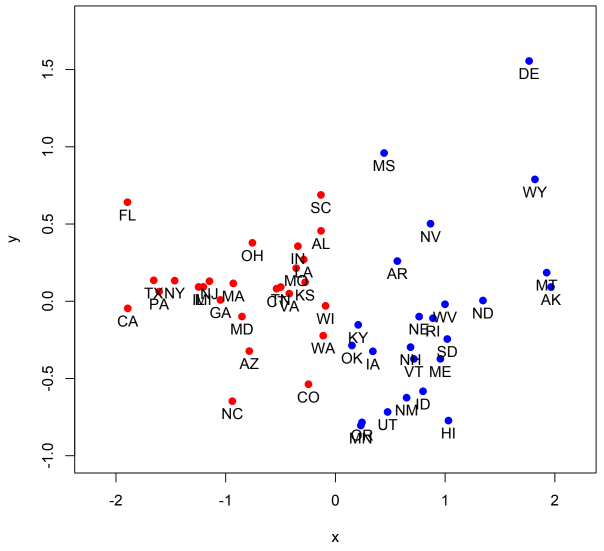


Figure 3.13: Multidimensional scaling (MDS) plot of 50 states based on feature vector in 2014. Two clusters are in red and blue.

3.2.3. Conclusion

In this analysis, we applied algorithms and methods from network science and statistics to explore many network features in the U.S. patient referral networks. Those network features describe both micro and macro patterns about patient referrals, such as power laws in some degree distributions, "small world" structure, Core-Periphery structure, motifs of triadic structures.

From 2009-2015 we found that the majority of network features are fairly stable. Our key results encompass both general or macro-level and micro-level network features. At the macro-level, the power law structure cannot be rejected in most cases, which suggests that these networks are "robust yet fragile" – i.e., robust to random failure, but susceptible to "targeted" attack (i.e., consciously specified removal). [Albert et al., 2000] The small-world property implies that physician networks are suitable for efficient information transfer and diffusion of innovations. [Watts, 1999] Analyses at both state and national network level tends to support the hypothesis of a "small world" and thus a fertile environment for diffusion (see also [Strogatz, 2001] and

[Kossinets et al., 2008] for other possible connections) and suggests a rich direction of future research. At a micro-level, the computation of actor specific network measures allows rankings of physicians to be constructed based on their importance in the referral network. Possible measures that can be used include degree, local clustering coefficient, CP score, and the number of external connections.

In general, the ultimate goal is to link the national physician network to individual patient data in order to perform patient-level analyses that account for patient demographics and clinical factors when assessing the association of the physician network and its salient subnetworks to important patient outcomes. For other contexts, the general network analysis is also applicable. We hope the derived network measures and structural patterns will boost the mining of more insights in a network.

The analysis of two cases: user behavior and network structure in this Chapter follows the routine at the beginning of the real practice of a data-driven project. We need to be familiar with the available dataset before finalizing a detailed requirement of a project. The following Chapter will present an example of feature engineering when our dataset is not diverse enough to support the project, as well as the numerical representation of information flow in a network.

Chapter 4

Feature Engineering and Entity Representation

In Chapter 4, we present two cases of feature engineering and entity representation: local search and the referral sequence. Local search relates to the case of walking and searching in a city and referral sequence considers the context of walking. They are both a kind of information walk in the sense introduced in Chapter 5. This is especially true for the feature engineering in Section 4.2, since it enables modeling in the pipeline of a data-driven project.

Local search helps users find certain types of business units (restaurants, gas stations, hospitals, etc.) in a given area. We are especially interested in the search for preferred business units near a user's current location. We call this the *local search problem*. Some merchants do not have much online content (e.g., customer reviews, business descriptions, opening hours, telephone numbers, etc.), this can pose a problem for traditional local search algorithms (e.g., vector space based approaches [Kalogeraki et al., 2002]). With this difficulty in mind, in Section 4.1 we present an approach to local search that incorporates geographic open data. Using the publicly available *Yelp* dataset [Yelp,] we are able to uncover patterns that link

Feature Engineering and Entity Representation

geographic features and user preferences. From this, we propose a model to infer user preferences that integrates geographic parameters. Through this model and its estimation of user preference, we develop a new framework for "local" (in the sense of geography) search that offsets a potential scarcity of features regarding the physical business units. Our initial analysis points to a meaningful integration of open geographic data in local search and points to several directions for further research.

Local search helps with a "physical walk" in a city, providing a way to predict/recommend to the walker a "next" place to go in a specific context. This is in some sense, an "information walk", wherein the user's previous and current locations can affect the prediction of a next position. Related - but different - is the analysis of and prediction for an information walk on an actual network, the aforementioned patient referral network. Therefore, extended from the network analysis in Chapter 3, we analyze the millions of referral sequences of patients' interactions with the healthcare system for each year in the 2006-2011 time period and relate them to cardiovascular treatment records. For a patient, a "referral sequence" records the chronological sequence of physicians encountered by a patient (subject to certain constraints on the times between encounters). It provides a basic unit of analysis in the broader *referral network*. We consider referral networks defined over a range of interactions as well as the characteristics of referral sequences, producing a characterization of the various networks as well as the physicians they comprise. The general method of entity representation in a network also works for other context beyond the referral sequence. The numerical representation of information flow will enable further predictive modeling.

Section 4.1

Local search

The work in this Section has already appeared in the refereed publication [An and Rockmore, 2016a].

Local search is responsive to the query for a certain type of "target" in the vicinity of a user's geographic location. In traditional web search, since some local business units do not contain significant text as a part of their online presence, information retrieval models based on word-document relationship may not work well. While current search engines generally are able to return satisfying results, they can be biased toward established businesses with a strong online presence. New business units that lack a significant online description/presence are still challenged by this "partial availability problem". Presuming that many new business units could also present excellent options for potential users, this is a problem/challenge/opportunity for enhancing user experience.

With this in mind, we examine how external freely available resources ("open data") can augment information to build an enhanced model for local search. For instance, the keywords in an advertisement of a local shop can be used as a proxy for the basic description of the shop. Basic geographic open data is also very useful. The locations of both user and business units enable the computation of the distance between them. With thoughtful design local search can be improved with geographic open data.

Current work tends to analyze a user's search log to improve local search. Teevan [Teevan et al., 2011] conducts a survey about mobile local search and describes the user's desired target in terms of distance and time, which suggests a rule of ranking in the local search problem. Lv [Lv et al., 2012] considers several user-related signals in ranking for mobile local search. Dragut [Dragut et al., 2014] merges similar search results in a local area with a consideration of user's ratings. Bernerich [Berberich et al., 2011] exploits direction requests, browsing logs and mobile search logs to refine search ranking. Meanwhile, Ahlers [Ahlers, 2013] introduces the "entity retrieval system" for Yellow Pages. Several papers note the geographic factors in search problems: Gan [Gan et al., 2008] investigates the properties of geo-queries and develops a new taxonomy for such queries. Lymberopoulos [Lymberopoulos et al., 2011] predicts click behaviors with high-level location features, such as states and zip codes.

The methods in the above papers have some limitations. Query log-based methods cannot perform well when a new user executes a query for a new local store. For the local search problem (in this project local search does not refer to the same-named optimization strategy in artificial intelligence), when the history records are not complete, improvements are derived from incorporating open data into the search model. Moreover, the user-oriented analysis should also take advantage of more detailed geographic and practical features beyond the simple distance. The geographic data that we request from open databases are details about local business units, such as the name of a store, the street address and the locations (accurate longitude and latitude). The more features we get, the more we may be able to improve local search. Other useful sources of information can also be included, including competitors, size of target stores and business categories, since these can affect a user's decision when choosing among several shops.

4.1.1. Geo Features vs. Preferences

Here we describe an open dataset, available on *Yelp* [Yelp,], and several geographic features that we wish to relate to user's choice. Our data analysis reveals several patterns linked to user preference.

***Yelp* dataset.** The *Yelp* dataset contains $1.6M$ reviews by $366,000$ users for $61,000$ business units. After applying a filter for the set of cities, which is that a city must have at least 10 business units of any kind listed, we are left with 96 cities in North America and Europe and a total of $60,503$ business units. For each business unit, the database provides the name, address, and its accurate location (latitude and longitude). Included are also reviews and linked ratings by customers. Though review content is also available, we do not dig into the natural language processing in this project (it presents a further consideration and potential opportunity). We can use the business unit location as the input for a secondary query to get more information from a Geocoder [Geopy-1.11.0,] database, then generate geographic features of business units, such as neighboring business units density and others in the following paragraph.

Features of interest. We explore the interactions of five features with the *Yelp* user ratings and the number of reviews (#reviews) per business unit. Several papers demonstrate the importance of incorporating a user's current location into mobile local search [Church and Smyth, 2008, Lv et al., 2012, Teevan et al., 2011]. While we account for that as well, we additionally consider the following information in our preference estimation model.

(1) **Significance of ratings and #reviews in an area**. For all business units in a city, we compute the average of all ratings and #reviews. The result will show whether local area matters in terms of user opinion. More statistical methods and criteria should be applied here in future research, such as weighted average and analysis of distribution about ratings and #reviews.

(2) **Average distance between a given business unit and the other business units in all types within the same city**. This is effectively a measure of business unit neighbor centrality. When a store is far from others, it looks like

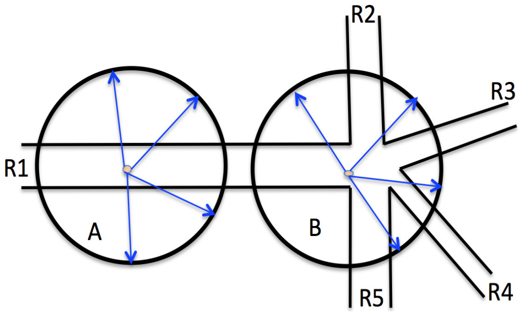


Figure 4.1: "Surrounding roads information". We look to incorporate the road address around certain center points of circles area in a road. We use the longitude and latitude of the center point of a circle to get the output of address names of points on the circumference. In Figure 4.1, given a radius of interest, we can determine circles of interest, A and B, with different centers along the road $R1$. We can compute the location of points on the circumference and query the corresponding address names from a geographic database. If we set the radius with different values, we can get the road names of more points near the central business unit.

an outlier away from the central business area of a city. More advanced metrics and methods in the detection of outliers can be applied.

(3) **Density of neighboring business units**. For a store, we count the number of neighbors of all types within a certain radius. We do not filter with the same type of business when counting the number of neighbors because different types of business might attract customers for each other. In general, high density may be linked to the existence of shopping centers or prosperous business areas.

(4) **Number of roads within a certain distance to a business unit**. This reflects the availability of local transportation. To find this, we query the addresses of several points near the business unit. The points are located on the circumference (without the limit of address query quota, we can set discrete values of the radius so as to get the address names of more points within a certain distance from the center point) of a circle at equal angle intervals, whose center is the business unit. Then we analyze the returned addresses to see the diversity (number of different roads by comparing road names) of roads nearby

the target. Considering the example situation in Figure 4.1, There we see stores A and B along the road $R1$. We draw two circles of a fixed radius around the two stores. The results of the query would include the full names of roads (e.g., $R2$ and $R3$) nearby.

(5) **Location of the business unit in a street or road.** "Location" is a categorical attribute like "middle" or "end". The attribute here is a relative concept. For the purposes of modeling we assume the business unit is located along a straight road, rather than some types of roads (e.g., highway and roundabout) without too many business units. Suppose a person is walking through blocks to find a store as the target of shopping. The in-street location might affect the possibility of seeing the store. We still query several points around the store, and count how many points on the circumference are on the same street with the central business unit. Consider again Figure 4.1, we introduce two circles. For the locations around a center which is in the middle of road $R1$, almost all points on the circumference are along the road $R1$, where the center of circle A is located in. As a comparison, the center of circle B is close to the right end of road $R1$, so the points on the circumference of circle B are located on different roads, thus fewer points are in the road $R1$. By this difference, we can judge the approximate location of a business unit on a road.

Patterns. The *Yelp* dataset offers the name, location, reviews and other information about a business unit. With the accurate location (longitude and latitude) of a point as the input, an open geographic database such as Geocoder [Geopy-1.11.0,] will return the full address of the point. Combined with the two data resources, we investigate the previous five features and produce the histograms in Figure 4.2.

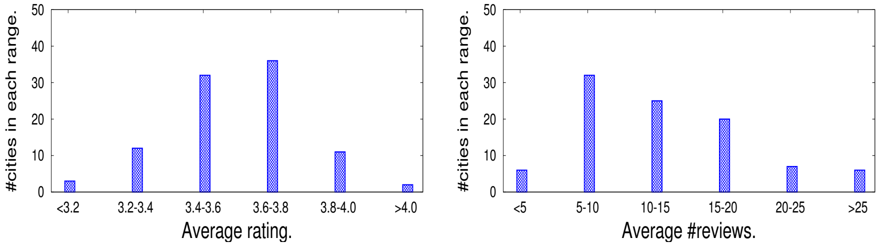


Figure 4.2: Histograms of rating data per city. Left: average rating. Right: average #reviews.

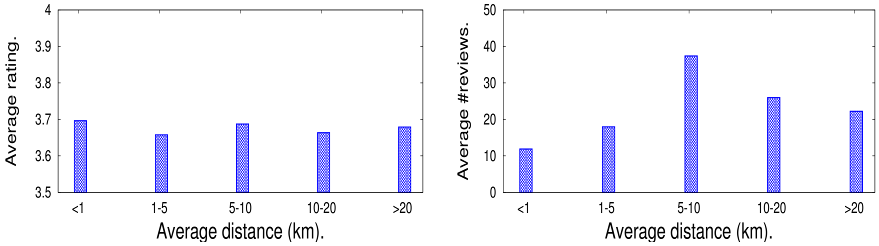


Figure 4.3: Effect of average distances. Left: average rating. Right: average #reviews.

On the left we see the histograms of the average ratings and on the right the average number of reviews for all business units, per city. The number of business units in a city varies from 11 to 13600. Though the majority of the average per city ratings are in range $[3.4, 3.8]$, the ratings have an obvious difference among cities, since the range of rating is an integer from 0 to 5 and users rarely give a rating lower than 3.2 (from the most left bin in the left histogram). We also find the uneven distribution of #reviews in the bottom histogram. Here we pick up the bin size using the knowledge of the mode and range. To sum up, Figure 4.2 tends to support the assumption that business unit location matters in terms of user's ratings and reviews on business units, so we should consider location in city scale (and perhaps with the other smaller scale geographic features) for user preference modeling.

Figure 4.3 shows the distributions of average rating and #reviews, when the average distances from one business unit to the others changes. Average distance is a form of geographic centrality. The average rating does not have a clear trend with the uniform distribution, but the uneven distribution of #reviews seems to match a

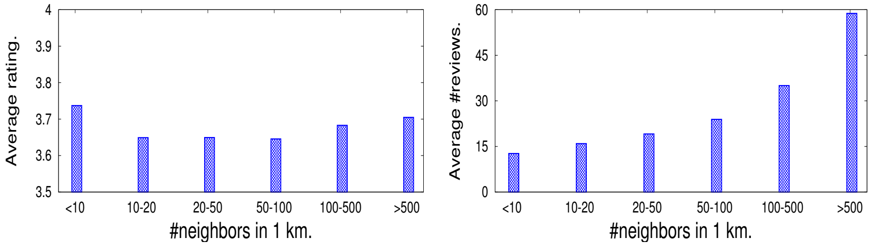


Figure 4.4: Effect of neighbors density. Left: Average rating. Right: average #reviews.

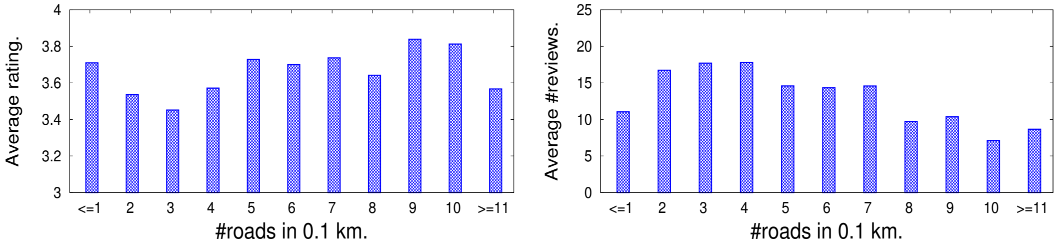


Figure 4.5: Effect of #roads nearby. Left: average rating. Right: average #reviews.

normal distribution or else. We find that if a business unit has an average distance of 5-10 km to others, it tends to receives the most reviews.

By Figure 4.4, we explore the relationship between #neighbors (number of neighbors) and reviews in terms of rating and #reviews. When a business unit has the least or the most neighboring business units within 1 km radius as the left subfigure shows, the ratings seem to be better than others. Though we are not sure why the low or high neighbor density might relate to a higher rating, the possible relationship suggests we consider the number of neighboring business units when predicting the user's reviews of a business unit. In addition, the bottom subfigure illustrates that more neighbors (high neighbor density) will bring more reviews. This corresponds with intuition: more people are attracted by more business units, and post more reviews there.

Figure 4.5 shows the relationship between #roads around a business unit and our statistics. We detect 12 points around a business unit with the same radius and equal interval (30 degree) between angles. The radius is 0.1 km and angles are (0, 30, 60

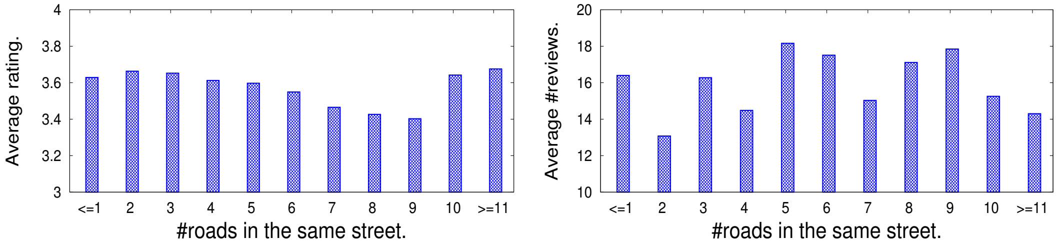


Figure 4.6: Effect of #roads in the same street. Left: average rating. Right: average #reviews.

... 330) degrees in anticlockwise direction starting from the x axis in a 2-D virtual plane. A comparison of the left half (0-5) and the right half (6-11) in the left subfigure finds a a general trend of more roads locate around a business unit correspond with higher average rating. In addition, when #roads becomes larger, #reviews decreases a little for some reason. One assumption is that a person might choose another way to go and miss some business units at road intersections, so a business unit with more surrounding roads might receive less reviews. To sum up, the #roads nearby seems to vary with the review statistics.

Figure 4.6 illustrates the relationship between location of a business unit in its street and our statistics. Limited by the query request quota of Geocoder, we randomly sample about three thousand business units from 27 cities and display the distributions in Figure 4.5 and Figure 4.6. Each business unit is treated as the center of a circle as Figure 4.1 shows, and we need the query of 12 points around the center to analyze the road information nearby, so it becomes a bottleneck given the limited times of address query. We find that business unit location in the middle or at the ends of a street corresponds weakly with a higher average rating. There is not much of a recognizable trend in #reviews.

The above figures suggest that geographic features do have some relationships with user ratings and #reviews. This in turn suggests that open geographic data can make an important contribution to local search.

4.1.2. User Preference Model

Let f_i denote the preference for business unit i and assume it has the form in Equation (4.1). It requires the known locations of the business unit and the user. It is a simple case since it does not include other neighboring business units as competitors. A larger value of f_i means the user is more likely to choose the business unit i in mind.

$$f_i = \frac{l^{\alpha_{l,i}} * t^{\alpha_{t,i}} * s^{\alpha_{s,i}} * g^{\alpha_{g,i}}}{c^{\beta_{c,i}}} \tag{4.1}$$

The values α_l, α_t, α_s, α_g and β_c are positive parameters reflecting user sensitivity, which is similar to weights in a linear function. Each item in the numerator should have its own exponent (α). The exponents denote the weights of several parameters. The multiplication form means that mismatch of a parameter may exclude the business from the short candidate list for a user. Users might be able to input their initial values, and the search algorithm can adapt the parameters with the response of query results. We also include several variables/functions in the numerator that might have a positive correlation with user's preference.

- The variable l captures the city's environmental bias factors. Different cities/towns have their own standard of rating and review style as Figure 4.2 and a previous work of click prediction in local search [Lymberopoulos et al., 2011] show.

- The variable t represents the text matching result. If the semantics of query inputs matches the type of business unit, t will be a larger value. For example, if a user would like to have a meal and input "Where to eat", then restaurants will have a higher value of all business units. Some open semantic data with latent vectors (such as Word2Vec[1]) might improve the matching performance.

[1] https://code.google.com/archive/p/word2vec/

- The variable s means the score of inner attributes for a business unit, including but not limited to the size of the store, the cleanliness, the opening hour, the quality of service. Other kinds of open data, such as customer's review and introduction on Yellow Pages, can also be added to determine the value of s.

- The variable g represents the score based on geographic factors. The density of neighbors, the feasibility of transportation (number of nearby roads) and the location of store in a road (middle or end) are possible attributes. The Figures 4.4 illustrate a possible relationship between geographic factors and user feedback, so we add this item into the model.

- The variable c represents the cost of traveling from the current location to the business unit, which has (generally) a negative correlation (when the cost for a business unit increases, the user will grade the business unit with a lower preference score) with user's preference so it is in the denominator. Limits on financial budget and time can affect user's choice. So it contains at least two parts, the time cost and money cost, depending on the way of traffic from user's current location to the target business unit. The user's current location is a key factor in the computation of c, since it determines the distance to the business unit. Several papers [Berberich et al., 2011, Lv et al., 2012, Teevan et al., 2011] point out the importance of distance in providing relevant recommendations.

Equation (4.1) is not the only possible form of a relationship between external factors and user preference. It might also take the form of a weighted sum, but this fractional form better reflects which factors have a positive or negative correlation with the preference. In addition, there are several functions behind t, s, g and c. Each function deals with factors of an aspect (e.g., geographic factors) and sets corresponding values to describe a user's preference.

In any real case, if the user has a general target (e.g., a shopping mall) rather than a clear query with a name (e.g., Walmart), the user might wander around a local business area. The neighbors of a store might compete with the store, or they might sell complementary goods. Equation (4.2) considers this case and encodes the effect of neighbors:

$$F_i = p * f_i + \sum_{k \in N_i} (1-p) * u(t_k, t_i) * \frac{f_k}{|N_k|} \tag{4.2}$$

- The variable F_i is total preference value with neighbor's contribution, which might work in the ranking part of a search engine.
- The variable f_i and f_k result from Equation (4.1). Store k is a neighbor of store i.
- N_i represents the set of store i's neighbors. $|N_k|$ means the size of the set. In traveling, a user might be attracted by other stores nearby, so the interest of a particular store can be affected either positively or negatively by a neighbor.
- The variable p is the probability of staying focused on the original target. It is a personal attribute about purchase behavior.
- The variable $u(t_k, t_i)$ describes the relationship between two business units. They may be cooperators or competitors.

Here the preference value F_i depends on two parts, the simple point-to-point interests and the neighbors' effects. To define the set of neighbors N_i, geographic open data must offer the locations of surrounding business units and the road information. To get $u(t_k, t_i)$, a comparison of keywords is necessary. For the personal parameters (α, β and p), the model should learn them using user's choices following query results. Over time, the search algorithm might provide customization according to

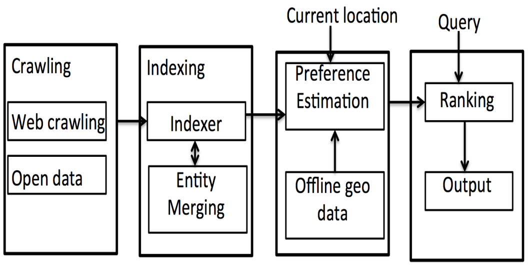


Figure 4.7: Revised structure of local search. It shows the data flow of the revised local search. After the crawling and indexing, one additional layer of preference estimation is added. It requires user's current location and query from open geographic data. The additional layer can give an estimation of user's preference. Finally, an input query will trigger the module of ranking combined with the estimation of preference.

user history. After the collection of query logs with user's location track, we can evaluate the model. The model puts more weights on features from open data, so even when the documents (set of words used in traditional information retrieval model) of business units are not complete, the revised model with geographic features might generate a list of preference values for a better ranking result.

4.1.3. Improved Local Search

Here we give a high level description of a user preference model incorporating geographic features. We describe the possible change in the structure of local search to comply with the model.

The use of open data. Though we mainly focus on the new incorporation of geographic open data, other types of open data can also contribute to a better (in terms of user experience and performance) ranking result in the structure. The classical structure of searching includes three sub-modules, which are *crawler, indexer,* and *query*. *Crawler* downloads webpages and *indexer* build indices for those words in webpages, then *query* responds to user input by returning the most highly related pages.

With this basic model in hand, here is a possible usage scenario of open data in searching. In the first step of crawling, the crawling from both online and offline open data (such as geographic databases, Yellow Pages brochure, social network reviews, etc.) should be performed. Since the local area often has a limited range of business unit candidates within a certain radius, it is possible to collect information from multiple aspects and resources, when the single resource cannot generate a large enough document set of business units. The second step is indexing. This necessitates execution of the challenging task of merging multiple descriptions of the same entity, acquired from diverse information resources. The third step is the modified design that incorporates local search. Since GPS on mobile devices enables a real-time location record, a user's current location can trigger the preference estimation model. The model will use the information of surrounding business units acquired from geographic open data. Semantic open data can also work in the matching of query and business category. The model will then produce a list of nearby business units with their preference values for a user. The last step is the response to a user's query. Traditional ranking results relate the semantic similarity between the input string and the candidate document with the index. Here we have another preference list based on the additional estimation model. A suitable mix of the two methods should improve the searching performance.

Advantages. The use of open data in preference estimation could solve the problem of insufficient web-available information about local business units. Besides, the added step of geographic analysis can also serve for a local recommendation system before the user's query. Meanwhile, the structure leaves room for incorporating other types of open data.

4.1.4. Conclusion

In this section, we analyze the patterns of relationships between geographic features derived from open geographic data and user preference, and describe a preference model that incorporates several detailed geographic features. We discuss the potential improvement derived from the structure of local search for better preference estimation. The initial analysis tends to support the idea that open data, and especially geographic open data, can be a powerful factor in estimating user preference, and local search incorporating a parser of geographic features might overcome a lack of descriptive words associated with business units. We used Yelp as a primary data source, but new geographical features will be beneficial to any location based services, such as shopping, traveling and entertainment.

Future possible directions of work include: (1) Collecting real query logs that track movement and evaluate the preference model and the revised local search. (2) Finding and determining more helpful geographic features. (3) Mining the patterns encoding the relationship geographic features and the preferences. At the same time, working on different scales of local data in terms of the size of a city and the radius of address query around a business unit.

Section 4.2 Referral Sequence

The work in Section 4.2 has already appeared in the refereed publication [An et al., 2018b]. It introduces a numerical feature representation of the referral sequence and suggests a schema for information flow in general. This section plays an important role in connecting the raw dataset and the following chapter of predictive modeling.

The language (and mathematics) of network science is well-adapted to the study of discretized and localized information and resource flow. In the particular case of healthcare records, a referral network generates various measures as a way of understanding patient care, healthcare resource allocation and treatment efficiency [An et al., 2018a]. A referral sequence for a given patient stores the date of the visit and interactions between a patient and each node on the sequence. Possibly because of specialty, different physicians might spend uneven amounts of time and effort (e.g., as measured by the relative value unit or "RVU")[2] during a typical encounter with a patient. We describe the referral sequence in terms of multiple features (e.g., the time between initial and final encounters or average RVU). Domains of investigation can range from the network of physicians in or attributed to a hospital, the Hospital Referral Region (HRR), or the entire United States referral network. A range of choices for edge weights can articulate different properties of these interactions. Given groups of referral network structural measures and referral sequence features, multilevel regression models and classification methods in machine learning have the potential to reveal relationships between the organization of patient flow in the healthcare system and the well-being of patients, and with this, insights into improving efficacy and resource allocation for our healthcare system.

Patient referral networks are a record of doctor interactions mediated by the sharing of a patient within a fixed timeframe. Through this interaction, information is shared. We are interested in understanding the process of this step by step sharing of information which we call an information walk. Classical models usually analyze the "explosive" spread of information on a social network (e.g., Twitter). This is a broadcast or epidemiological model wherein a given source node "infects" multiple targets.

Prior studies related to referral sequences have been limited in terms of the range of health records studied [Uddin et al., 2013, Uddin, 2016]. In this project, we

[2] RVU stands for "Relative Value Unit". This is a Medicare invention used in the calculation of reimbursements that encodes the "value" of a given procedure.

analyze a much larger dataset and also include new metrics related to the study of referral sequences and are able to compute detailed network measures in a much larger dataset (the TDI[3] dataset) of cardiovascular disease treatment, ranging from a local hospital or HRR to the current national referral network. Aggregating the data from thousands of local hospitals and hundreds of HRRs, we use statistical methods to validate the general patterns of referral sequences and referral networks.

We characterize the dynamics of changes of node position and type among all physicians on a referral sequence. In the case of cardiovascular treatment, we find evidence of key roles on a referral sequence, especially for the physicians with a specialty of cardiovascular and internal medicine. We also validate the prevalence of patterns of referrals indicating that physicians work with their professional acquaintances when choosing the target of a referral, i.e., regularly send patients to the physicians who have many common collaborators. As a secondary benefit, we then apply classification models to the cardiovascular referral network measures and referral sequence features to predict the teaching status of a hospital and a patient's treatment outcome (e.g., an indicator of death within 1 year after treatment). Our considerations of networks and referral sequences for cardiovascular treatment could clearly be adapted for other contexts. More specifically, given patient referral records tied to a different disease state, the metrics and methodologies we introduce here (e.g., the feature and pattern mining, model selection, analysis, etc.) could be directly adapted. In addition, our study has implications for research about a generalized notion of "referral sequence" in such contexts as information flow in online media or social networks.

Some specific contributions of our feature engineering and entity representation work include:

[3] The Dartmouth Institute for Health Policy and Clinical Practice

- Novel definition of the health records-based referral sequence as well as a novel definition of salient features for referral sequences generated from both network science and time series analysis.

- Quantification of a physician's position using centrality and other measures in the U.S. national cardiovascular referral network with the help of techniques specific to big data that are necessary for overcoming the infeasibility of using traditional algorithms for calculations at scale.

- Investigation of the patterns of millions of referral sequences in the referral network, which are validated by statistical tests.

4.2.1. Materials and Methodology

We used Medicare beneficiary claims data for all patients diagnosed with cardiovascular disease in the U.S. during 2006-2011 to build referral sequences and networks of the U.S. healthcare system. Here cardiovascular disease means that the patient suffers from arrhythmia, congestive heart failure, coronary-heart disease or peripheral vascular disease in the diagnostic codes of Medicare claims. This dataset is of interest for several reasons. It is on the one hand a kind of network "big data" (as we will see, giving rise to networks on hundreds of thousands nodes and millions of edges) in a research area (healthcare) where traditionally data analysis has not been accomplished at this scale. In particular, by focusing on the part of the national dataset related to disease diagnosis, we can begin to articulate and build out methodologies that relate to outcomes. Each such record contains the patient or "beneficiary" (Bene) identification (ID) number, physician National Provider Identification (NPI) number, visit date, RVU associated with the visit and other details. Since the NPI numbers for all physicians changed in 2007, some of the analysis we perform only obtains for the interval 2007-2011. Although claims data and other sources of patient-physician encounters have been previously used to form physician networks [An et al., 2018a, Landon et al., 2012, Mandl et al., 2014, Lomi et al., 2014, Shea et al., 1999], in this project we apply a more nuanced approach. The "referral sequence" is a maximal sequence of referrals [4].

4.2.2. Definition

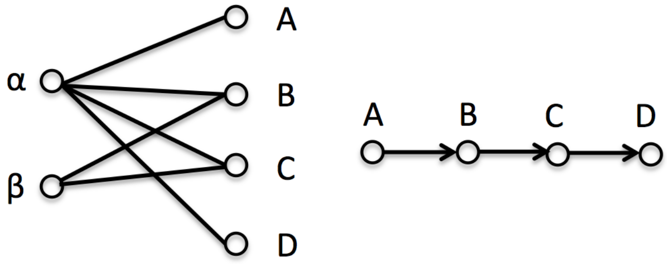


Figure 4.8: Bipartite graph between patients (α, β) and physicians (A, B, C, D). (L) An edge between a patient and a physician means the patient visits the physician. (R) A referral sequence of Patient α in chronological order.

In Figure 4.8, several edges connect two patients (α and β) to some physicians whom they have visited. Patient α visits four physicians (A, B, C, D). By sorting the four physicians according to the date of patient α's visit, we recover a sequence of four physicians reflecting the sequence of encounters.

4.2.3. Referral Network and Edge Weights

The *referral network* (over a given time period) is a directed network with node set given by the physicians present in the database over a fixed time period. If physician A refers at least one patient to physician B, this is represented by a directed edge from A to B. Given all referrals over a year, we are able to build the *U.S. national patient referral network of US physicians.* In this project, we mainly investigate micro-patterns of referral sequences for each patient in HRR/PHN

[4] the team of physicians involved in the treatment of a patient over the course of a given episode of illness

referral networks, while our prior work [An et al., 2018a] introduces macro-patterns derived from directed national, HRR, and state referral sub-networks. Herein, most of the network measures are also derived from directed referral networks, except a few measures from the corresponding undirected networks, such as diameter, clustering coefficient and giant component.

Edges can be weighted in a variety of ways. A simple unweighted edge (i.e., edge weight equal to 1) denotes simply a connection. More information is added if we use other natural metrics such as the number of referrals or the geometric mean of RVU. A novel metric that we define here is the "ranking based weight": Let the vector $r = (1, 2, \dots, n)$ denote the chronological "ranks"[5] of the encounters on a referral sequence consisting of n physicians. In this case for a given physician A, let n_A denote the number of encounters for physician A on the referral sequence, and let r_A be the sub-list of the ranks of the encounters with a physician in the referral sequence (so, if A was encountered on the first and last visits only, then $r_A = (1, n)$). In this way, n_A is the length of the r_A. The flow of patients from physician A to physician B is then given by

$$f_{AB} = \frac{\sum_{i<j} I(r_{Ai} < r_{Bj})}{n_A n_B} \tag{4.3}$$

and from B to A by

$$f_{BA} = \frac{\sum_{i<j} I(r_{Ai} > r_{Bj})}{n_A n_B} \quad . \tag{4.4}$$

To compute the ranking based weight of an edge, we compute a weighted sum of the patient ranking index flow in each referral sequence p containing both physician A and B. A referral sequence p might include multiple physicians, but the flow of patients in the referral sequence between physician A and B only relate to their sub-vectors r_A and r_B, without any impact from a third physician. The function of Equation 4.3

[5] The list of positions – denoting first, second,...,nth – in the sequence of n visits that make up the referral sequence.

increases in value at a rate proportional to a constant [Wikipedia, 2018] as n_A and n_B go to infinity, but we would like to account for the length of each referral sequence, so we add n_{Ap} and n_{Bp} and weigh the contribution from each referral sequence by their geometric mean in Equation 4.5.

$$w_{AB} = \sum_p (n_{Ap} n_{Bp})^{1/2} f_{ABp} \tag{4.5}$$

4.2.4. Referral Sequence Features

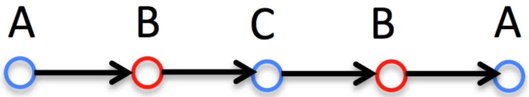


Figure 4.9: An example referral sequence with three physicians A, B, C. The patient visits them five times. Let's also assume that physician A and C are from the same HRR/hospital in blue, while physician B is from another HRR/hospital in red.

As discussed in Chapter 3 we introduce the use of various basic network measures for the study of patient referral networks and uncover macro-level network structures including general patterns of "power law" in degree distribution, "small-world" structure, core-periphery structure, and the existence of a "gravity law" in a state-level referral traffic map. In this project we focus on the referral sequence and to that end, introduce some metrics that get at the diversity of a referral sequence. Denote the number of visits on a referral sequence as N, the ith node on a referral sequence as P_i, the date of the encounter with the ith node as T_i, $1 \leqslant i \leqslant N$. With this notation we make the following definitions and illustrate them using the example in Figure 4.9 (note that in Figure 4.9, the nodes corresponding to the physicians are color-coded according to some affiliation datum – e.g., HRR or hospital):

- *Sequence length.* The total number of physicians on a referral sequence. A physician would be counted multiple times if the patient visits the physician repeatedly. It is 5 in Figure 4.9.

- *Recurrence.* A binary variable recording whether there exists i, j, with $1 \leqslant i<j \leqslant N$, and $P_i = P_j$. It is true (set to "1") in Figure 4.9 because of multiple occurrences of physicians A and B.

- *Time range.* $T_N - T_1$. It is the gap between the last visit and the first.

- *Average time gap between referrals on the referral sequence*: $\frac{T_N - T_1}{N-1}$.

- *Number of nodes before recurrence.* It refers to the first reappearance of a duplicate node. In our example, it is 3 since the first three nodes A, B, C are different from each other before the first duplicate node, B.

- *Physician distribution entropy.* This is the standard probabilistic definition of entropy $(-\sum_x p(x)\log_2(x))$ derived here from the physician occurrence probability over the sequence. In Figure 4.9, the frequencies of A, B, C are $2, 2, 1$ respectively. The physician distribution entropy of the related probability distribution $(0.4, 0.4, 0.2)$ is 1.522.

- *Hospital distribution entropy.* The entropy of the derived physicians' hospital distribution is another feature of diversity. Since we assume A and C are from the same hospital, the frequency distribution is $(3, 2)$ and the corresponding entropy is 0.971.

- HRR distribution entropy. The entropy of the physicians' HRR probability is another feature of diversity. It is the same value as PHN distribution entropy under the assumption that A and C are in the same HRR.

- *Main hospital.* It is a derived referral sequence feature of the hospital in which the most physicians on the referral sequence are working. It is the hospital with A and C in Figure 4.9.

- *Main or dominant HRR.* The HRR in which the most physicians are working. It is the HRR with A and C in Figure 4.9.

- *Number of pairs of nodes with reciprocal referrals on a referral sequence.* $\sum_{i,j} 1(1 \leqslant i < j \leqslant T-1, P_i = P_{j+1}, P_{i+1} = P_j)$. There are two pairs of nodes (A, B) and (B, C) which have such reciprocal relations.

4.2.5. Node Position Features

In a referral network, metrics related to node characteristics correspond to metrics of physician “importance”. Meaningful examples include local clustering coefficient, betweenness centrality, closeness centrality, eigenvector centrality, PageRank centrality [Page et al., 1999], core-periphery score [Rombach et al., 2014]. In addition, we adopt the notion of h-index to the patient referral network [Hirsch, 2005]. For a node in the national referral network, consider the array of indegrees for all nodes which refer patients to the node, then count the h-index of the indegree array, which means h referral source nodes have at least h indegree in the array.

Here are some of the features describing node position that are relevant to the context of referral sequences.

- *Number of sequences that contain the node.*

- *Number of sequences where the node is the initial visit.* In Figure 4.9, physician A is the first node.

- *Number of sequences where the node is the final visit.* In Figure 4.9, physician A is the end node.

- *Average index of the first-time occurrence in all sequences.* In Figure 4.9, the index of first-time occurrence for nodes A, B, C is $1, 2, 3$, respectively, so we can take the average over all referral sequences.

- *Number of sequences where the node occurs multiple times.* In Figure 4.9, nodes A and B occur twice.

- *Number of cross-HRR referrals proposed by the node.* In Figure 4.9, given the assumption that nodes A and C are from the same HRR, node A sends patients to node B in another HRR. Nodes B and C also form an edge that spans HRRs.

- *Number of cross-hospital referrals proposed by the node.* In Figure 4.9, given the assumption that nodes A and C are from the same PHN, node A sends patients to node B in another hospital. The same is true of nodes B and C.

4.2.6. Results

We process raw patient-physician encounter records, build referral paths/networks and derive the following patterns in Python, with the help of NetworkX [NetworkX-Developers, 2017]. We build the machine learning programs for treatment outcome prediction with scikit-learn [Scikit-community, 2017], and implement statistical tests and regression models in R.

Table 4.1 describes features of millions of referral sequences over 2006-2011. The average duration of each referral sequence is roughly 25 days (avg time range) and comprises about four nodes (avg length). About one-third of referral sequences have a node which the “defining patient” visits multiple times. The distribution of the referral sequences when weighted by hospital entropy is more diverse than when weighted by HRR entropy, which implies that a patient will more likely visit multiple hospitals in the same HRR than to have multiple visits in different regions (HRRs). Close to half of the pairs on a given referral sequence are reciprocating.

Table 4.1: Overall statistics of all referral paths in 2006-2011.

Year	2006	2007	2008	2009	2010	2011
#referral sequences	4.44M	4.45M	4.54M	4.59M	4.63M	4.66M
avg length	3.850	3.907	3.983	4.023	4.061	4.115
avg gap for a referral	8.509	8.506	8.369	8.352	8.230	8.060
avg time range	24.247	24.727	24.969	25.245	25.192	25.109
percent of sequences with recurrent nodes	33.418	32.879	32.836	32.784	32.573	32.301
avg #nodes before recurrence	4.087	4.130	4.179	4.196	4.223	4.271
avg physician entropy	1.400	1.410	1.423	1.427	1.436	1.448
avg hospital entropy	0.475	0.473	0.476	0.459	0.480	0.481
avg HRR entropy	0.107	0.109	0.108	0.105	0.112	0.116
avg bidirectional pairs in a sequence	0.450	0.455	0.465	0.474	0.476	0.479

In addition to the basic overall features for all referral sequences, we explore other patterns from other perspectives.

Index on Referral Sequence vs. Node Position in Network Corresponding "node position sequences" encode how a patient navigates along with physicians in terms of the physician position of importance in the referral network. Here we consider the node position sequence with respect to five node position measures in the national referral network: clustering coefficient, betweenness centrality, eigenvector centrality, PageRank centrality and h-index. Figure 4.10 shows an observed node position sequence represented by the local clustering coefficient of each node. After classical seasonal decomposition [Meyer, 2017] by moving averages on the sequence, the seasonal component tends to fluctuate, which suggests that physicians in the core and periphery parts appear alternately on the referral sequence.

Denote the N physicians on a referral sequence as $P = (P_1, P_2...P_N)$ and the node position value of P_i as C_i, so that the corresponding node position sequence can be denoted as $C = (C_1, C_2...C_N)$. Then the number of changes in trend $\sum_{i=2}^{N-1} 1((C_i - C_{i-1})(C_{i+1} - C_i) < 0)$ counts the change of sgn (positive, negative) of the difference

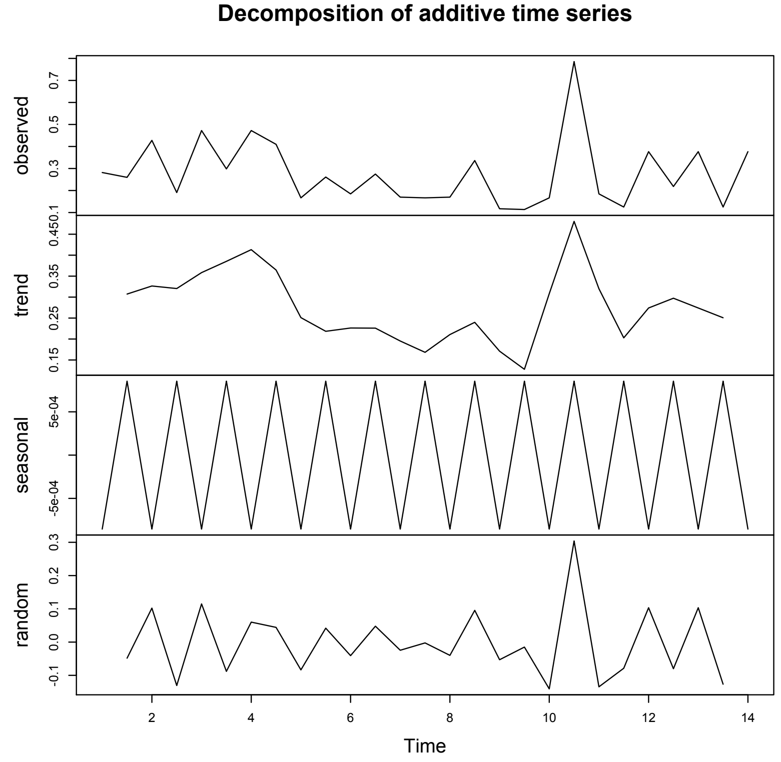


Figure 4.10: Observed local clustering coefficient of the nodes on a referral sequence, and the three components divided by time series decomposition. The seasonal component fluctuates along the time axis.

in the centrality of successive providers on referral sequence P_i, $2 \leq i \leq N-1$. The event $(C_i - C_{i-1})(C_{i+1} - C_i) < 0$ is defined as a change point. For each node in the middle of a referral sequence, if the neighboring nodes and itself satisfy the condition, it contributes one to the number of change points.

Table 4.2 shows the percentage of change points in terms of five kinds of node position measures in 2007-2011. In most cases, a patient will alternate visits between a physician with a larger centrality measure and one with smaller centrality measure. The pattern is stable in different years with all node centrality measures, which suggests that some core physicians in the national referral network help to link some physicians with fewer referrals for the patient's treatment.

Table 4.2: Percentage of change points in terms of increasing/decreasing trend in node position sequence of a referral sequence.

Year	2007	2008	2009	2010	2011
clustering coefficient	75.0	74.9	74.9	74.8	74.7
betweenness centrality	74.9	74.7	74.8	74.7	74.5
eigenvector centrality	74.3	74.2	74.2	74.1	74.0
PageRank centrality	74.8	74.6	74.7	74.6	74.5
h-index	70.7	70.6	70.8	70.8	70.8

Table 4.3: Comparison of average common connected nodes between neighbors on a referral sequence and the expectation in a random network with the same size.

Year	2006	2007	2008	2009	2010	2011
Random network	$3.60E-03$	$3.10E-03$	$3.00E-03$	$2.90E-03$	$2.80E-03$	$2.80E-03$
Referral network	25.13	24.64	24.95	24.97	24.95	24.96

The fluctuation suggests that on a referral sequence some physicians with relatively larger centrality measure might diagnose the disease and organize the referral sequence by referring the patient to nodes with lower centrality. This is the role that has been envisioned for primary care physicians in the health care system and prior network analyses [Barnett et al., 2012a] have found that the more prominent (i.e., central) primary care physicians are in an intra-hospital network, the less the average cost of care at that hospital.

Preference of collaboration Sometimes a physician might have multiple options in terms of the target of a referral, especially when the physician is located in the center of referral networks with a wide range of connections. We compute the average number of common connected nodes for neighboring nodes in a referral sequence P, given by $\frac{\sum_{i=1}^{N-1}|V(P_i)\cap V(P_{i+1})|}{N-1}$, where $V(P_i)$ is the set of neighboring nodes of node P_i in the national referral network.

Table 4.3 shows that on average the neighbors or direct collaborators on a referral sequence have 25 common collaborators in the national referral network, while the expected number in a random network is $p(AX, BX|AB) = (N-2)\frac{(M-1)(M-2)}{(C_2^N-1)(C_2^N-2)}$ (N is the number of nodes, M is the number of edges). Assume there is an edge between node A and B. Then the remaining $N-2$ nodes are candidates for common neighbors. With $M-1$ edges remaining in the whole network and C_2^N-1 remaining pairs of possible edges, the probability that A and a candidate neighbor X are connected is $\frac{M-1}{C_2^N-1}$, which is almost the same as the ensuring conditional probability that B and X are connected. The sum of probabilities over $N-2$ candidates leads to the resulting probability being multiplied by $N-2$ to yield the expected value for the network. The clear gap in Table 4.3 supports a hypothesis that physicians tend to work with an acquaintance or someone in the same community when a referral is required. Among the referral steps of all referral sequences in 2006-2011, only 33.2% are cross-PHN while 7.5% are cross-HRR referrals, which suggests that internal referral within the same hospital or HRR is the first choice. This suggests that actual geographic distance may be a factor for referral target selection. This would enable modeling of choice of referral targets as a ranking problem that would take into account geographic proximity (as well as possibly other factors).

4.2.7. Conclusion

We consider the new information sharing model of the information walk on a network and construct new features about a referral sequence in the referral network. Several exciting patterns show the power of proper feature engineering. For other contexts of information flow/walk, it is possible to define similar features for different tasks. For example, the sequence of webpages of user's browsing may generate specific features to improve a recommender system.

Chapter 5

Predictive Models about Information Flow

In Chapter 4 we did feature engineering for information flow in a network, now in Chapter 5 we take up three predictive tasks about information flow. The dynamic changes in a network make it a challenge to predict the future of ongoing information flow, and an accurate prediction of information flow would be valuable for the corresponding community of a social network.

First, we study the problem of walk-specific information spread in directed complex networks. An important and motivating example is the sequence of physicians visited by a given patient over a presumed course of treatment or health event. In this case, the patient (and her health record) is a source of "information" from one physician to the next. The records of transitions define the corresponding network, where the existence and times of visiting some nodes in history will influence the future possibility of visiting (transition) between a pair of nodes. Since we assume a context of information sharing and metadata in specific domains, we name these *information walks* and the problem in our research as *information walk prediction.* We build a Bayesian Personalized Ranking (BPR) model to predict the next node on a walk of a given network navigator using network science features. The problem is related to but different from the well-investigated link prediction problem [Martínez et al., 2017]. We present experiments on a dataset of several million nodes, showing that the application of network science measures in the BPR framework boosts hit-rate and mean percentile rank for the task of next-node prediction. The work in this Chapter has already appeared in the refereed publication [An et al., 2019, An et al., 2018b].

Second, We then move beyond the simple information walk to consider the derived network space of all information walks within a period, in which a node represents an information walk, and two information walks are connected if have nodes in common from the original (social) network. To evaluate the utility of such a network of information walks, we simulate outliers of information walks and distinguish them with the other normal information walks, using five distance metrics for the derived feature vectors between two information walks. The experimental results of such a proof-of-concept application show the utility of the derived information walk network for the outlier monitoring of information flow on an intelligent network.

Finally, based on the case of patient referral sequence, we apply machine learning methods to predict the outcome of the event (i.e., treatment) on the information flow.

The predictive models introduced in this Chapter are the core part of a machine learning project. Though they are built on the context of a kind of information flow in this thesis, in general they connect the previous steps of feature engineering and the following step of evaluation/improvement.

Section 5.1

Direction

With the knowledge of network science, it is natural to build a network for a group of people (or even any items which can help with information sharing) based on their pairwise interactions. The person-to-person communication in such a network turns into a *path* [Wikipedia, 2019], or more accurately a *walk* [Wikipedia, 2019], since it is possible (and in many contexts even likely) for the "walker" (e.g., news) to revisit some person (node).[1] Indeed, multiple "visits" can provide a kind of reinforcement of the information of interest that might be relevant to its learning or absorption. This node-by-node (e.g., person-after-person) information spread model – a "single-track" model – is a kind of epidemiological model but different from the classical diffusion/broadcasting models that are often used in the analysis of social media.

Single-track information spread is appropriate to our particular interest: the problem of *next visit prediction* of a walker in a network. Our original motivation arises from research on physician collaboration networks built by referrals [An et al., 2018b], where two nodes/physicians are directly connected with a weighted edge if they have been visited by the same patients within a given period. Patients "walk" this network in the course of a presumptive treatment event. A predictive application based on the features of such referral sequences may provide a better understanding of the process of collaboration among health professionals.

Furthermore, precise prediction of the next visited physician may help with the efficient allocation of medical resources for a patient's treatment. If we know a physician would probably have to treat many patients, we may prepare some

[1] One will recall that technically a "path" is a sequence of visits of connected nodes with no node visited more than once, while a "walk" only requires the sequence of visited nodes be connected.

assistance in advance. Other examples of single-track information walks in different contexts include a traveler visiting preferred places, consumers traversing stores in a shopping mall, or the work history of an employee. Indeed, a walker may be the first to ever traverse from one node to another – suggesting that these nodes did not connect each other in history records. Therefore, a more accurate framing is the problem of *visit prediction* for a walker in a *state space*. In the above instances the entire walk up to the last node may directly affect the selection of the next visited node, so that this problem is generally not a memoryless Markov chain.

Herein, exploiting both metrics proposed in our analysis ([An et al., 2018b] and [An et al., 2018a]) and classical network science measures, we propose a numerical score to model the preference/attraction between the last observed node on an information walk and any possible candidate node in the network. This score takes multiple feature vectors from the targeted information walk as well as several groups of involved nodes. Based on the preference score, we apply a general Bayesian Personalized Ranking (BPR) framework to represent the goal of next-node prediction in an objective function so that the problem could be solved by machine learning. Several network science measures (e.g., node centrality) in the national physician network facilitate the prediction for a pair of nodes, including those not directly linked in the past.

5.1.1. Proposed Models

We begin with a preference model for information walk prediction, then describe how to build a network of all information walks, and a proximity-based unsupervised framework for information walk outlier detection.

Given an observed information walk in a directed network, the first task is to predict the next visited node. To do so, we build a numerical preference/attraction score for the observed part of an information walk (including the last node visited and an overall feature comprising all past visited nodes) and any possible next-visited candidate. Therefore, when predicting which node would be more likely to be visited by a walker, we can compute and sort the preference/attraction scores over all candidate nodes. We then pick out a small number of nodes which have a comparatively large score. As a result, this prediction framework allows for the convenient detection of possible choices from the returned list (see Figure 4.8 for an illustration of the identification process). The definition of a preference score is a key component of the algorithm.

To formalize the problem, let P denote the set of all chronological node sequences (i.e., information walks). For an information walk $i \in P$, p_i represents the feature vector of the observed sequence of nodes at a time point T, c_i refers to the last node on information walk i before time point T, f_i is the first node on information walk i after T (i.e., the actual next visited node). Let J represent the set of possible candidates, which could cover a wide range of nodes, even the whole network except c_i, or just a subset of nodes in the network after filtering to speed up the computation if the network is large. $X(p_i, c_i, j)$ denote the preference/attraction score between the last observed node c_i, the overall walk feature p_i and a candidate $j \in J$ for the next node. We aim to derive an objective function and train the preference-related parameters to make $X(p_i, c_i, f_i) > X(p_i, c_i, j)$ for as many candidates $j \in J$ (and $j \neq c_i$) as possible. If so, it indicates that a model predicts the next node on an information walk (i.e., the future direction in a network space) more accurately.

Diverse groups of network science features, either exogenous (metadata) or endogenous (topological) to the observed walk, may boost the accuracy of information walk prediction. The features detailed in Chapters 3 and 4 (also see [An et al., 2018b] and [An et al., 2018a]) offer groups of such features useful for building our new preference score model. Table 5.3 shows a detailed list of features used here.

Table 5.1: Dimension of the model parameters/features in Equation 5.1.

Feature	Dimension	Note	Parameter	Dimension	Note
p	$M \times 1$	information walk	V	$M \times N$	walk-node interaction
β	$N \times 1$	last node	S	$M \times H$	walk-node interaction
f, γ	$H \times 1$	ground truth / candidate	U	$N \times H$	node interaction
d	$L \times 1$	profile similarity	w	$1 \times L$	profile weight

5.1.2. Preference/Attraction score

We define a preference score $X(p_i, c_i, j)$ in the BPR framework, called *BPR-IW* using the feature vector p_i of the information walk. The other factors in the preference/attraction score are the last/current node c_i of walk i and a node $j \in J$ as the candidate:

$$X(p_i, c_i, j) = p_i^T V \beta_{c_i} + p_i^T S \gamma_j + \beta_{c_i}^T U \gamma_j + wd(c_i, j) \tag{5.1}$$

where in Equation 5.1 the superscript T refers to the transpose operator for a matrix. p_i means the overall feature of the whole observed part of information walk i. β_{c_i}, γ_j represent the feature vector of the last node c_i and a candidate node j, respectively. $d(c_i, j)$ represents the distance between c_i and j in terms of their profile similarity based on the metadata. Three matrices V, S, U about the node-walk interactions will be trained as model parameters, which represent the feature interactions that exist in a theoretical Factorization Machine [Rendle, 2010] or Polynomial Regression [Theil, 1992] model. In addition, another parameter w (weights) adjusts the importance of node profile similarity, which corresponds to the last group of features in Table 5.3. To make the matrix operation in Equation (5.1) clear, Table 5.1 shows the dimension of several key parameters/vectors.

Equation (5.1) considers multiple factors when predicting the next visited node on an information walk. S, V represent the interaction between the initial part of the walk and the candidate/ground truth node, respectively, while U describes the extent of matching between the candidate and the last node on the walk which might influence the decision of the future direction. Network science provides the widely applicable features p, β, f, γ, since they can be computed from the topological structure of a network, regardless of the type of metadata in the network. As the profile distance d relies on the context (e.g., physician specialty), we distinguish it from the other features.

5.1.3. Learning BPR-IW model

Equation (5.1) defines a preference score $X(p_i, c_i, j)$ for sorting candidate nodes in J for an information walk i. When evaluating the ranking of candidate nodes for an information walk, it is convenient to get the scores for all candidates, and then pick the top-K candidates. In this way, the relative order of the score counts more than the actual values. The Bayesian Personalized Ranking (BPR) framework [Rendle et al., 2009] defines the objective function as finding the optimal fitting MAP estimator with the use of regularization to guide the choice of predictors. The crucial part of this Bayesian procedure is the evaluation of the posterior probability of the model parameters conditional on the network (i.e., the interactions among nodes stemming from patients' preferences about the next physician they visit). The procedure is presented mathematically in Equation (5.2):

$$\Theta = \underset{\Theta}{argmax} \sum_{i \in P_{train}} \sum_{j \in J \setminus \{c_i, f_i\}} \log \sigma(\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)) - \frac{\lambda_\Theta}{2} ||\Theta||^2 \tag{5.2}$$

where σ represents the sigmoid function $\sigma(x) = (1 + \exp(-x))^{-1}$. Using the sigmoid function, the gap between two preference scores for two candidate nodes is mapped into the interval $(0, 1)$ so that the loss function is defined even if the gap diverges to infinity when computing the optimal model parameters. The components of σ, $\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)$, describe the gap in the preference scores between the ground truth of the current walk, f_i, and another possible candidate, j. P_{train} refers to the training set information walks. In the objective function (5.2), Θ is a general set parameter to be learned in the training process, such as V, S, U, w introduced by Equation (5.1). We can use several random matrices/vectors drawn from a multivariate Gaussian distribution as initial values. The values of the model parameters will be optimized in the iterative training process. As the last item, λ_Θ regularizes the objective function to avoid overfitting.

According to the size of the dataset in Table 5.2, the number of pairs of information walks and candidate nodes $O(|P_{train}||J|)$ is huge (more than 1 billion). In this case, stochastic gradient descent (SGD) optimizes Equation (5.2) efficiently, which updates the set of parameters Θ based on the derived gradient in Equation (5.3). To update the parameters in each round of SGD with an information walk i and a candidate node j, the gradients of Equation (5.2) for a parameter $\theta \in \Theta$ are:

$$\begin{aligned}&\tfrac{\partial}{\partial\theta}(log\sigma(\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)) - \tfrac{\lambda_\theta}{2}||\theta||^2)\\&= (1 - \sigma(\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j))\tfrac{\partial}{\partial\theta}(\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)) - \lambda_\theta\theta\end{aligned} \tag{5.3}$$

The partial derivative of $\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)$ with respect to some parameter could be computed by Equation (5.1). Equation (5.4) gives the instances of S and U that are defined in Table 5.1. Note that due to an offset in the gap of two preference scores, it is not necessary to update V.

$$\begin{aligned}&\tfrac{\partial}{\partial S}(\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)) = p_i\gamma_{f_i}^T - p_i\gamma_j^T\\&\tfrac{\partial}{\partial U}(\hat{X}(p_i, c_i, f_i) - \hat{X}(p_i, c_i, j)) = \beta_{c_i}\gamma_{f_i}^T - \beta_{c_i}\gamma_j^T\end{aligned} \tag{5.4}$$

5.1.4. Evaluation

Dataset. The data for our analyses are the U.S. Medicare beneficiary insurance claims for a subgroup of patients over 2007–2011 explained in Chapter 4.

For the set of information walks P in a year, given an observation time point T we build the training set P_{train} to store the walks ending before T. The test set P_{test} includes the walks that are ongoing at time T. Figure 5.1 illustrates two examples. Since information walk A terminates before time point T, it is in the training set P_{train}. At the time point T, a node on walk B is still passing information to the next node, so walk B belongs to the test set P_{test}. In A and B, the observed red nodes contribute to the overall information walk feature p. For a walk in P_{train}, all nodes but the last one belong to the observed part, while the last node serves as the ground truth f. The candidate set J contains the ground truth f of all walks in P_{test}; thus it randomly samples a subset of nodes in the whole network.

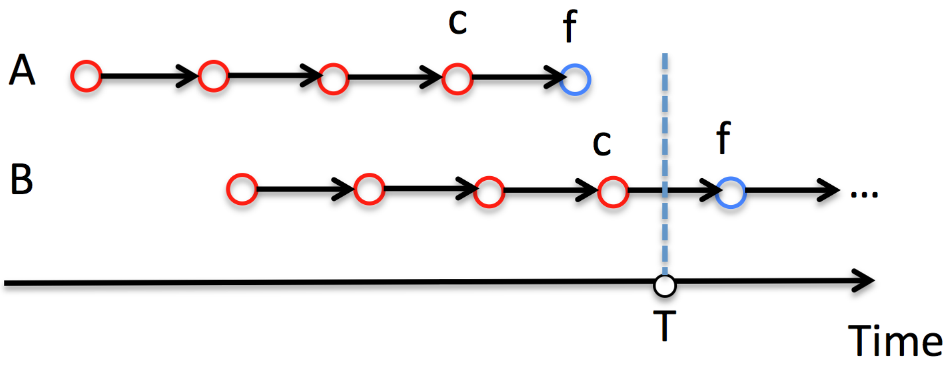


Figure 5.1: At a given time point T, two information walks (A and B) belong to the training and test set, respectively.

The U.S. physician collaboration network derived from the TDI dataset produced 4.66M information (referral) walks in 2011. The training and test set are defined as information walks with at least six visits. Table 5.2 presents the size of training and test sets at several time-points T, as wells as the candidate node set J. The size

Table 5.2: Size of training, test and candidate sets at different time points in 2011, which are derived from the TDI dataset.

Date of observation T	P_{train}	P_{test}	candidate nodes J
03/01	17.6K	18.7K	16.6K
05/01	51.8K	19.5K	17.3K
07/01	83.7K	16.6K	14.9K
09/01	113.1K	15.8K	14.3K
11/01	142.4K	16.8K	15.1K

Table 5.3: Features about information walks and related nodes including applicable network measures, new metrics defined by our past analysis [An et al., 2018b], and a few from the metadata of medical treatment records, such as Relative Value Units (RVU) of medical service. Our paper [An et al., 2018c] shows the full list of applied measures.

Group	Measures
information walk p	number of nodes on it, time range, pairs of mutually connected nodes, sum of RVU for all visiting, number of visited hospitals, average node PageRank values
next node/candidates (j and f)	clustering coefficient, PageRank, Hindex, number of initiated cross hospital referral region referrals
last node c	Beyond the features in the group of next node/candidate: time gap with last occurrence, RVU, a binary flag of multiple occurrences on the walk, a binary flag of working in the same hospital previous physician (node)
metadata for profile similarity $d(c, j)$	Indicators of the same specialty/residency hospital/hospital referral region, number of referrals in history.

of P_{train} increases from March to November, since it contains all information walks ending before T.

Table 5.3 groups by the measures of an information walk p, the feature vector γ of a candidate that of the ground truth f, the feature vector β of the last node c on an information walk. $d(c_i, j)$ refers to profile similarity between two physicians. Each group contains several representatives of the full list explained in our past works ([An et al., 2018b] and [An et al., 2018c]). We picked the above measures as they boosted predictive performance in other applications (e.g., the result of medical treatment along an information walk [An et al., 2018b]). To mitigate concerns about reverse-causality and to avoid the possible problem of predicting a variable with input features in the future, when we extract features of an information walk in some year (e.g., 2010), we use node centrality measures derived from the network in the previous year (e.g., 2009).

Baseline Methods. Our BPR-IW model is a general model for diverse contexts of information walks, not limited to the case of patient referrals. In addition to our proposed BPR-IW, the models/metrics below also generate a preference score X between a candidate node j and the last node c, so they could sort their available candidate nodes for a top-K subset as the prediction result.

Most popular (MP). $X(c, j) = e(c, j)$ It takes the edge weight in history between c and a directly connected neighbor. It refers to the number of referrals between two physicians. However, the range of candidates is limited.

The performance of traditional link prediction methods are used as benchmarks against which to compare the new methods. We also implement several representative methods, including **Common neighbors (CN)** [**Lorrain and White, 1971**], **Preferential attachment index (PA)** [**Liben-Nowell and Kleinberg, 2007**], **Adamic-Adar index** [**Adamic and Adar, 2003**] and **Jaccard index** [**Jaccard, 1901**]. Notably, these similarity metrics do not incorporate the other nodes on the observed part of an information walk, and are only applicable for the neighbors that interacted with node c before. However, our BPR-IW model extends the range of possibly predicted candidates, even without a direct edge or common connected nodes with the last node c.

Markov Chain (MC) [**Rendle et al., 2010**]. $X(c, j) = Prob(c.next = j | c, c.prev)$ The two-gram version incorporates the second-to-last node $c.prev$ so as to compute the frequency of state transition.

Long Short-Term Memory (LSTM). Given the corresponding node sequence of an information walk, we treat the features of all nodes (in Table 5.3) as the time series inputs into a LSTM model [Hochreiter and Schmidhuber, 1997]. We aim to explore whether the LSTM model could learn the hidden patterns based on the past node-to-node transitions to yield an output tensor that is very close to the ground truth f. However, the hit-rate of LSTM is lower than 0.01 under all parameter settings in our experiment. Another paper [Choi et al., 2016] reported a similar level of failure of LSTM when predicting the next medical visit.

Transition-based Factorization Machines (TFM) [Pasricha and McAuley, 2018]. The TFM model merges the current item, next item and user into a $1 \times n$ vector $\overrightarrow{y}$. It defines a preference score according to Equation (5.5), in which d^2 is the Euclidean distance function, $\overrightarrow{w}$ is a weight vector, $\overrightarrow{v}$ and $\overrightarrow{v}'$ represent latent embedding and translation vectors, respectively:

$$X(\overrightarrow{y}) = w_0 + \sum_{a=1}^{n} w_a y_a + \sum_{a=1}^{n} \sum_{b=a+1}^{n} d^2(\overrightarrow{v}_a + \overrightarrow{v}'_a, \overrightarrow{v}_b) y_a y_b. \tag{5.5}$$

The hit-rate (defined in Equation (5.6)) of TFM on the TDI referral data is less than 0.01 under all experimental settings, including an overall $\overrightarrow{y}$ with our proposed network measures and a comparatively plain $\overrightarrow{y}$ with three IDs only (walker, current and next node). The majority of the nodes in the network of physicians have a small node degree (< 4). Therefore, in such a cold-start environment TFM may not perform as well as that on a dense dataset [Pasricha and McAuley, 2018] consisting of frequent users and a part of nodes. Meanwhile, when most of the applied network measures are not categorical, TFM does not make full use of its advantage of dealing with the features in one-hot encoding. TFM enumerates all possible pairs of feature interactions, but some of them may not boost the prediction. As a highlight of TFM, it is better for the latent transition vector $\overrightarrow{v}'$ to depend on the past track (i.e., observed walk).

BPR-no-IW. $X(c, j) = wd(c, j)$. As a comparative method to BPR-IW, this model only takes the item of physician profile similarity in Equation (5.1) to show the power of the other network science measures about an information walk and the related nodes.

Results. For a pair consisting of walk i and its next node f_i as the ground truth, BPR-IW or any of the baseline models will return a sorted list of K candidate nodes R_i. Here we choose two evaluation metrics: hit-rate (HR) and mean percentile rank (MPR) defined by Equation (5.6). HR reflects the possibility of presenting the ground truth f to users in the returned list, while MPR corresponds to the expected efforts a user may take to find the ground truth.

$$\begin{aligned} HR &= \tfrac{1}{|P_{test}|} \textstyle\sum_{i \in P_{test}} 1(f_i \in R_i) \\ MPR &= \tfrac{1}{HR \times |P_{test}|} \textstyle\sum_{i \in P_{test}, f_i \in R_i} \tfrac{rank(f_i)}{K} \end{aligned} \tag{5.6}$$

A smaller MPR yet larger HR implies a more accurate predictive model, which indicates that users would see the ground truth on top of the user interface from sorting the returned candidates in decreasing order according to their preference scores. Since the hit-rate values of LSTM and TFM are less than 0.01, Figure 5.2 through Figure 5.7 only present the result of the other successful models. As for the parameters in training process, the λ_Θ in Equation (5.2) is 0.001 and the step size in the SGD updating process is 0.05.

Figure 5.2 and Figure 5.3 show the HR and MPR at several time points in 2011 for BPR-IW and other baselines, under the setting of $K = 20$ in the returned list. In terms of HR, BPR-IW beats the others and BPR-no-IW performs the second best.

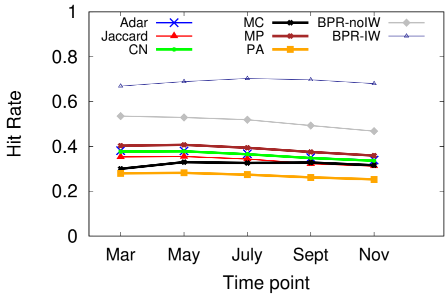


Figure 5.2: HR at several time points in 2011, when $K = 20$.

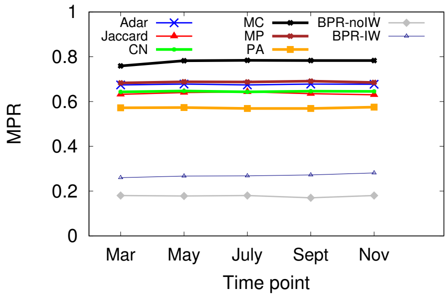


Figure 5.3: MPR at several time points in 2011, when $K = 20$.

The other baseline methods get close hit-rate values between 0.3 and 0.4. In addition, BPR-IW and BPR-noIW get the smallest MPR, which suggests the ground truth f would be located near the top of the returned list. For most of the models, the different observation time points do not result in obvious gaps in HR or MPR.

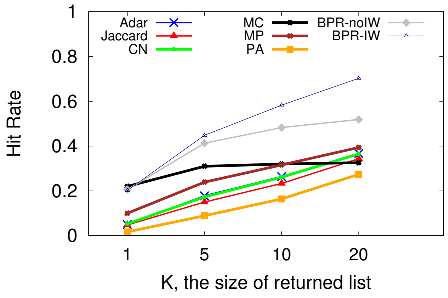


Figure 5.4: HR with different K on 07/01/2011.

Figure 5.4 and Figure 5.5 show the impact of K on HR and MPR on the same day of observation. Note that in Figure 5.5, MPR will be 1.0 for all models if the only returned candidate $(K = 1)$ hits the ground truth. When K increases from 1 to 20,

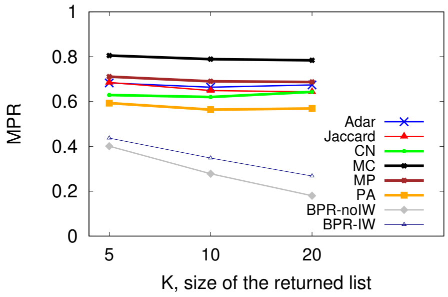


Figure 5.5: MPR with different K on 07/01/2011.

most of the models predict the next node better because the HR increases as well. For our proposed BPR-IW model, under the setting of $K = 20$, the HR is over 0.7 for the test set P_{test} with 10K+ information walks. For MPR, non-BPR models are almost stable when K increases, but BPR-IW and BPR-IW display a decreasing MPR from 0.4 to 0.2. As a result, it may be more desirable to choose a slightly larger K for BPR related models so that the walk prediction system could present more possible candidates to users, including the key node of ground truth f. We compare those models with different K values since it is relevant to user experience and needs to be accounted for in the design of a real application, like the number of pages returned on a webpage in response to a search query.

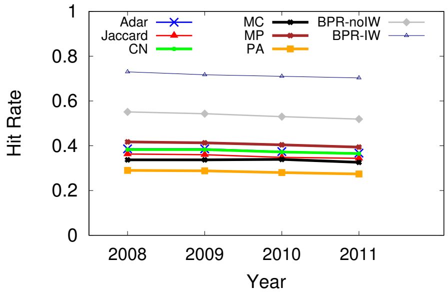


Figure 5.6: HR on 07/01 during 2008-2011, when $K = 20$.

Figure 5.6 and Figure 5.7 show the HR and MPR on 07/01 from 2008 to 2011, respectively. It seems that all models perform very stable on the same day in those

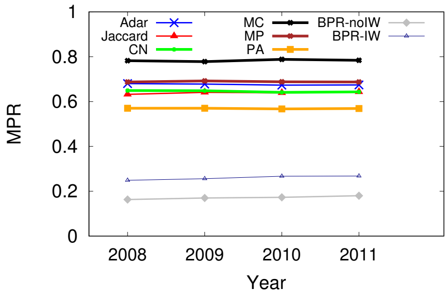


Figure 5.7: MPR on 07/01 during 2008-2011, when $K = 20$.

years, which tends to support that the network structure in the years of 2008-2011 may be steady as well.

Based on two basic features of an information walk, the length and time range, we implement min-max normalization and classify the test set into five groups based on the percentile. We compute the recall for the walks whose ground truth f is successfully predicted by the BPR-IW model. The stable performance in all five groups under varying size of R supports that BPR-IW does not adapt to one group (e.g., a longer information walk) much better than another, at least no obvious difference in terms of information walk length and time range.

Our initial experiments illustrate that features derived from network science and time series analysis for the nodes on an information walk greatly boost HR at the cost of only a slightly larger MPR. We believe it is more desirable and necessary to present the ground truth node to users than the comparative ranking within the list. Therefore, BPR-IW performs the best in our experimental settings. The classical link structure based metrics do not predict as well as BPR-IW, since they do not consider the feature p of the whole observed information walk. In addition, they are able to find candidates from the connected or other nearby nodes only, according to the network in history. The BPR framework does not predict the next node directly with a state transition probability. However, the output of relative ranking is enough for the users who do not want to determine the quantitative reasons behind the prediction. From the perspective of network research, we greatly recommend the application of network measures and the derived information walk features for further related projects. In addition, metadata also provides important features, since the data-specific features (e.g., physician profile similarity) appear presumably to help with successful prediction in the BPR-no-IW model.

5.1.5. Conclusion

We exploit the sequence of referrals in a physician collaboration network to solve the problem of next-node prediction on single-track information walks from a network science perspective, explore the network of multiple information walks, and implement a simulation test of information walks outlier detection to support the general idea of an information walk network.

We consider both newly derived information walk features and classical node centrality features to build a BPR-IW model of preference/attraction. The network-based measures yield a flexible BPR-IW model that identifies more possible candidate nodes than the traditional static link prediction method, because in BPR-IW it is not necessary for the last observed node to be directly connected with a candidate. BPR-IW works well on the TDI referral dataset according to a sensitivity analysis which tests both hit-rate and mean percentile ranking across multiple factors, such as the time point (within and cross-year) of observation and the number of nodes in the returned list. BPR-IW could be conveniently applied to other datasets, where network science measures will probably successfully model the structures and relationships among a set of items and nodes.

Since the BPR-IW model exploits general features derived from network analysis and time series analysis, it could adapt to different context of network (e.g., network of cities for traveling route and company network for career path). Based on the generalizability, domain experts may continue to add context-related features to boost the performance of BPR-IW prediction.

Section 5.2

Outlier

Section 5.2 moves beyond our published paper [An et al., 2018c] in the Complex Networks 2018 conference through its introduction and use of the relationships among multiple ongoing information walks. We also investigate the space of information walks with a network science model, in which each node represents an information walk and an edge connects two nodes of information walks if they share at least one common node (e.g., the same physician) in the originating network. We find several significant patterns in the new network of information walks and verify them via a statistical test.

A key contribution is our identification of criteria to label an information walk with different structural patterns in the network of information walks as an outlier. We use a simulation-based test of information walk outliers in the network of information walks in order to (1) demonstrate the efficacy of the model for the information walks network; (2) complement the proposed BPR-IW model of walk prediction since the users of an intelligent network platform may not have time to focus on every walk and check the prediction of its future direction while an overall outlier detection function can be used to filter some "abnormal" or "new" walks and remind users to check, so that users would be able to reduce the risk of loss made by abnormal information flow, or detect the benefits of novel walks in an early stage. In related work [Eswaran and Faloutsos, 2018, Ranshous et al., 2015, Savage et al., 2014, Takahashi et al., 2011], researchers have targeted different parts in a graph to build a specific outlier detection algorithm, including nodes, subgraphs, separate point-to-point edges (e.g., TCP-IP communication, connections between new accounts in social networks). Herein we are the first to implement outlier detection for a whole information walk, which differs from prior work due to the existence of the same single "walker" or information flow along the sequence of visited nodes.

We simulate the outlier information walks with random replacement of their nodes, explore the measures of an information walk in a network of information walks, and design five distance metrics (based on the walk features) within a general outlier detection framework to distinguish the simulated (outlier) information walks from those actually observed. Moreover, since an outlier information walk may be an abnormal or creative (e.g., new treatment procedure) case, the initial results suggest a way to contribute to a more intelligent network via outlier detection for ongoing information walks, which complements our proposed BPR walk prediction model.

5.2.1. Network of Information Walks

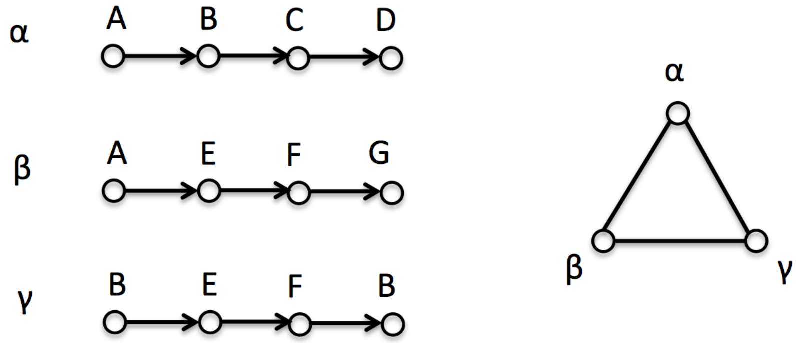


Figure 5.8: Three information walks with nodes from A, B, ..., G, and the corresponding network of information walks.

Here we define the network of information walks to model the space of all information walks, in which a node represents an information walk and two nodes are connected when they share at least one node in the originating network.

In the network of information walks, several edge weights distinguish the relationship between two connected nodes (i.e., information walks), such as the number of distinct common nodes, the Jaccard index [Jaccard, 1901] (size of intersection divided by size of union) of two sets of originating nodes on two information walks. Figure 5.8 shows an example of the network of information walks. Here α, β, γ are three information walks with several nodes $(A, B, \ldots, G)$. Since every pair of information walks share at least one node, the corresponding information walk network is an undirected 3-node clique. For the edge linking β and γ, the number of common nodes is 2 (nodes E and F), and the Jaccard index weight is $2/5 = 0.4$ because in total there are five kinds of nodes on them.

5.2.2. Outlier Detection for Information Walks

As a task for walk prediction, outlier detection identifies the information walks that deviate from the expected patterns of the observed track in an unsupervised set of information walks via the features derived from the network of information walks. The target of outlier detection is the entire information walk rather than a single node or edge on it. We hope that such a detector could provide an early stage “alert”, identifying “abnormal” or favorable novel information walks to improve the safety of subsequent carried information and the robustness of the network. For example, if we detect a referral sequence that does not follow the popular patterns in Section 4.2, it may suggest a delay in treatment or some improper referrals. To implement outlier detection, we need to define key features of the network of information walks and use these to design an algorithmic outlier detector.

An information walk grows node by node. Thus the evolution of an information walk could be presented by a series of cumulative feature vectors at each timestamp when the walker visits a new node. For example, the number of directly connected distinct nodes would increase since new nodes join the sequence. We present the general algorithmic outlier detection framework in Algorithm 3, which requires a distance metric function between any pair of information walks.

Algorithm 3 Proximity based outlier detection.

```
Input: A set of $n$ unsupervised information walks (IWs). A parameter $M$ to pick up the
  $M$th nearest neighbor for an outlier score.
Output: Pick up $K$ information walks (IWs) as the outliers.
  Compute the cumulative time series features $CF_i$ for each $IW_i \in$ {1,...,n}
  Outlierscore ← { }
  for $i \leftarrow 1$ to $n$ do
    Tmp-array ← [ ]
    for $j \leftarrow 1$ to $n$ do
      If $j\,!= i$ Tmp-array.append(Dist($CF_i$, $CF_j$))
    end for
    sort(Tmp-array)
    Outlierscore[Tmp-array[$M$]] ← $i$
  end for
  Tmp-array ← sort(Outlierscore.keys(), decreasing)
  Outlier-walks ← Outlierscore[Tmp-array[$1, \ldots K$]]
```

Algorithm 3 is an unsupervised proximity-based outlier detection framework. The key idea is to compute an "outlier-score" for each IW to pick the K information walks with the largest K scores. The data preparation step refers to Line 1 in Algorithm 3, where we compute the time series features for every information walk. In Lines 2-10, we compute the pairwise distance between two information walks with some metric (introduced later in this subsection) and treat the distance to the Mth nearest neighbor as the outlier score. Finally, in Lines 11-12, we sort the outlier score to get the Top-K candidates of outliers. With a time complexity of $O(n^2)$, Algorithm 3 more easily adapts to diverse kinds of proximity measures than statistical outlier detection methods that are reliant on assuming probability distributions of the residuals and models the degree to which IW is an outlier. A drawback is that the algorithm might be sensitive to the choice of M when defining the outlier score, making it necessary to tune the parameter M for each experiment.

Assume we extract P different measures of an ongoing information walk at T timeslots on the time axis. In total, the feature vector is then a $P \times T$ tensor. An equal-weighted distance function sums up the distance of each measure. Therefore, the *Dist* function in Line 6 could be transformed to a distance function between a pair of numerical arrays of each measure, but their lengths may be different due to the varying lengths of information walks. Denote the longer array as LA and the shorter one as SA and their lengths as l and s, respectively. Here we propose or apply five distance metrics to complete Algorithm 3.

- **Sliding substring matching (SSM)**. To match the shorter array SA, enumerate all s-length consecutive subarrays from LA and take the minimum Manhattan Distance between a subarray in LA and the SA.

- **Edit distance/Dynamic Time Wrapping (ED/DTW)**. Equation (5.7) describes the state transition equation for the dynamic programming model, in which $d(i, j)$ is the distance between the first i units in LA and the first j units in SA. The initial settings are $d(i, 0) = i \times \lambda$ for $i \in [1, l]$ $d(0, j) = j \times \lambda$ for $j \in [1, s]$. λ is the penalty factor to represent the cost of skipping a unit in an array. After the process of dynamic programming in Equation (5.7), the value of $d(l, s)$ is our desired distance.

$$d(i,j) = min \begin{cases} d(i-1, j-1) + abs(LA[i] - SA[j]), \\ d(i, j-1) + \lambda, \\ d(i-1, j) + \lambda \end{cases} \tag{5.7}$$

- **Interpolation**. Treats LA and SA as several discrete samples from a function of time in the interval [0, 1], in which the first unit in LA and SA is at zero while the last unit is at one. The rest of the non-extreme units are allocated with an equal interval. For example, if $LA = [0.1, 0, 2, 0.3, 0, 4, 0.5]$, the corresponding time intervals would be $(0, 0.1), (0.25, 0.2), (0.5, 0.3), (0.75, 0.4), (1.0, 0.5)$. To align SA and LA we take the simple linear interpolation for the corresponding points of LA to get new points that have the same time-index with SA. Finally, we compute the pairwise Manhattan Distance.

- **Longest common substring (LCS)**. The LCS method originally aims to find the longest subsequence common to two strings. In contrast to substrings, subsequences are not required to occupy consecutive positions within the original sequence. Two numerical units are treated as equal if their abstract distance is less than the threshold.

- **Sliding substring averaging (SSA)**. Starting from the first node in LA, set a sliding window of length of $l - s + 1$ and extract the average of those units in LA covered by the sliding window. The sliding window moves right one unit each iteration to generate s values from LA, so that it can compute the distance between the derived values and SA.

5.2.3. Patterns in Network of Information Walks

In this section we define some operations on walks that are slightly inspired by operations in algebraic topology.

We detect statistically significant (p-value less than 0.05) patterns between a pair or among several special information walks defined by some structural relationship. The following patterns are derived from the information walks network in the first quarter of 2011. They may suggest hidden patterns in the healthcare system for domain experts to explain and analyze the effects in further research.

- Citing the notion of path-homotopy from algebraic topology which explore structural similarity between curves, we focus on a pair of homotopic information walks as two information walks which share the common starting and ending nodes in the physician collaboration network. Because of the existence of two guaranteed common nodes, the homotopic information walks

Table 5.4: Comparison of three kinds of edge weights in the network of information walks, between the edges connecting homotopic (the same starting and ending nodes) walks and the others connecting two non-homotopic walks. The visiting records and RVU refers to the values of common physicians on the two walks.

	Jaccard index	number of visiting records	sum of RVU
homotopic pairs	0.552	24.48	45.00
non-homotopic pairs	0.234	10.28	19.65

are more closely connected in the network of information networks than a pair of non-homotopic walks. Table 5.4 shows the comparison, in which all the measures are found to be significantly different by a two-sample t-test.

- "Lifting" refers to a shortcut of a longer information walk. Assuming a longer information walk contains three consecutive nodes $A \rightarrow X \rightarrow B$, another shorter walk contains $A \rightarrow B$, and the rest of the nodes are the same, we treat the two walks as a pair of lifting walks. In the first quarter of 2011, there are 76K pairs of homotopic walks, and the shorter base walks have an average PageRank value of 1.07×10^{-5} while the longer extended walks have an average PageRank value of 1.20×10^{-5}. Meanwhile, when putting the middle node X between A and B in the originating physician collaboration network, we find a significant difference in the resulting PageRank centrality of the nodes. The order is X < A < B.

- Information walk composition exists among three groups of information walks. The first group ends with two nodes $A \rightarrow B$, the second one starts with two nodes $B \rightarrow C$, and the third contains the three nodes $A \rightarrow B \rightarrow C$ in the middle of the corresponding physician (node) sequence. Those three groups of information walks have significantly different PageRank values in the network of information walks, which are: the first group 1×10^{-5}, the second group 9.8×10^{-6}, the third 1.09×10^{-5}.

5.2.4. Simulation Test of Outlier Detection

Since the information walks in our physician collaboration network do not have a natural metric, we evaluate the framework of outlier detection and five distance metrics on a mixed set of the originating observed information walks and the simulated outliers. We exploit the training set at a time interval of observation defined by Figure 4.8 to get the neighbor (i.e., directly connected) list of every past node (physician). We then take all the information walks beginning within one month of the focal observation to sample from in order to form mixed set. Taking the observation date as 2010-03-01 as an example, from the IWs beginning in April 2010 we randomly pick up $2,500$ IWs as the normal cases and the other 2,500 IWs to generate outliers. To simulate an outlier, we keep the original starting and ending nodes of an IW but randomly replace all the middle nodes with others from the set of nodes located on a pool of IWs. The analysis period begins in the month following the observation period to provide the pool of IWs for node replacement. In this way, for a general test without a specific definition of an outlier information walk, the replacement operation at least alters the track of the whole information walk to some degree, but retains the basic source and target nodes.

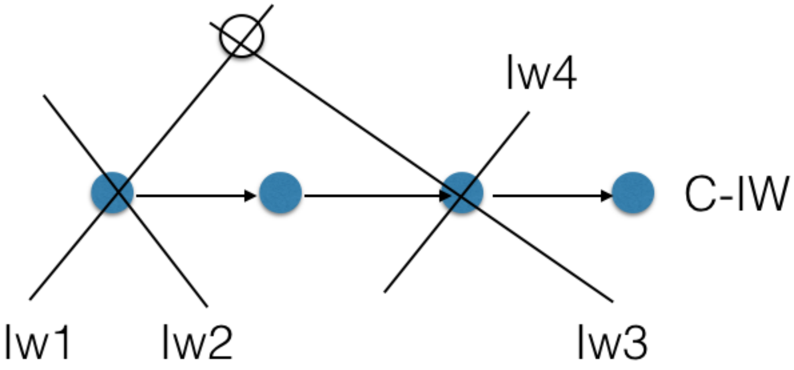


Figure 5.9: A current information walk (C-IW) consists of four colored nodes. Four different IWs share at least one node with C-IW. Besides, IW1 and IW3 have another common node.

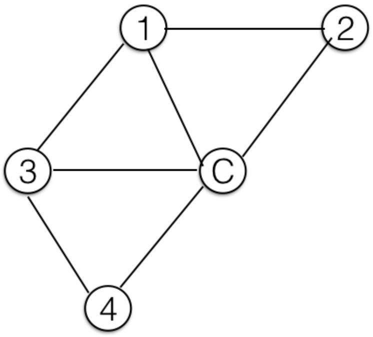


Figure 5.10: The corresponding network of information walks in Figure 5.9.

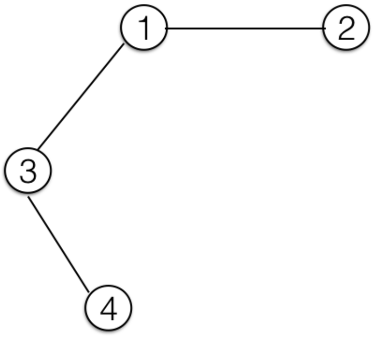


Figure 5.11: The remaining walk-subnetwork after dropping the current walk (C-IW) from the information walks network in Figure 5.10.

To apply the five distance metrics between a pair of information walks, we compute the following network science measures for an ongoing/current information walk at each step. They are either popular network measures or special measures to describe the relationship between the ongoing IW and its connected nodes in the network of IWs. Figure 5.9 gives an example of a current information walk (C-IW) with four connected IWs. Figure 5.10 and Figure 5.11 illustrate walk-subnetwork and remaining walk-subnetwork, respectively. The difference between these two local networks shows the alteration of the network itself if the IW is dropped. The comparison metrics are:

(1) Number of connected nodes in the network of information walk. Represent the set of nodes (walks) with the ongoing IW as the walk-subnetwork.

(2) Number of physicians which are the neighbor of at least one physician on the ongoing information walk.

(3) Number of physicians which are the neighbor of at least one physician on a walk in the walk-subnetwork.

(4) Average number of covered physicians: the value of measure (3) over that of measure (1).

(5) Average Jaccard index weight of those edges within the walk-subnetwork.

(6) Network strength of the walk-subnetwork, in terms of the weight of the number of common physicians.

(7) Variance of the edge weights in the walk-subnetwork centralization, using the number of common physicians as weights.

(8) Transitivity of the walk-subnetwork using the binary undirected edge.

(9) Survival rate of edges in the walk-subnetwork if the current IW (i.e., a node) is removed. Denote the left edges and their connected nodes as the remaining walk-subnetwork.

(10) Edge density in the remaining walk-subnetwork.

(11) Size of the largest connected component in the remaining walk-subnetwork.

The evaluation metric is hit-rate (precision), which means the percentage of outliers in the returned K candidates. Figure 5.12 shows the performance of five distance metrics under their optimal M about the choice of a similar neighbor for the outlier score. We tune the neighboring choice parameter M for each metric to maximize the hit-rate. Under different values of K, ED/DTW performs better than

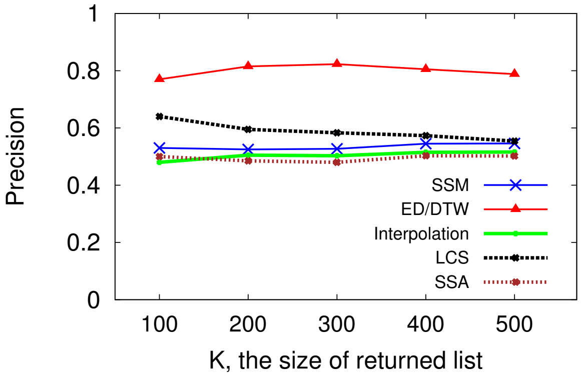


Figure 5.12: Precision of outlier detection under different Top-K returned walks setting.

others, and its optimal value is $M = 10$. The simulation test is a proof-of-concept of the application of features derived from the network of information walks, which suggests the possibility of the unsupervised proximity based information walk outlier detection. The distance metrics might work better on real outliers. Therefore, to be cautious, we should not judge the best metric based on the current simulation test. Furthermore, we also have multiple options for sampling normal cases and outliers, such as the Bootstrap and the Jacknife [Efron, 1992]. However, in the simulation test we set a balanced ratio between normal cases and the outliers. The selected feature set of 11 network measures may be expanded and optimized with feature engineering or statistical factor analysis in order to correctly detect an outlier in a new (unseen) dataset.

We try to define several operations and metrics in the space of IWs. The above outlier detection is just one application. Beyond that we can define the taxonomy of IW with a proper distance metric. Moreover, a taxonomy of IW networks would be available if we sample a few IWs from each IW network and compute their distance. Back to the definition of IW network, we have many other options, such as the common walker (e.g., patient), a larger threshold of the number of common visited nodes, or even a generative random model to produce different kinds of IW space.

Therefore, our work about IW outlier detection points out many future directions towards a theoretical framework of information walk.

5.2.5. Conclusion

The network of information walks has several significant patterns (e.g., high clustering coefficient) and provides several features for the simulation test of outlier detection, in which the Edit Distance/Dynamic Time Wrapping based metric performs the best over all metrics in a general proximity based unsupervised framework. Anticipated future work includes the prediction of real outliers defined by domain experts and the subsequent deployment of such an intelligent information walk prediction and detection system. The outlier detection framework is not limited by the context of referral network because of the general model of information walks network.

Section 5.3

Event Outcome

We guess the track of IW may affect the future status of the walker, such as the treatment outcome of a patient. Therefore, with a numerical representation of an IW, we are interested in the prediction of outcome for the event carried by the IW. The challenge main exists in the previous step of feature engineering since it is natural to apply some standard machine learning models here.

5.3.1. Dataset and Features

We next explore whether it is possible to predict the treatment outcome for a patient based on the measures and features of the physician referral network and the referral sequence. Here we take a dataset of Medicare patients diagnosed with Acute Myocardial Infarction (AMI) over 2006-2011, which by virtue of the serious nature of the medical event was always diagnosed in a hospital setting. Because AMI embodies a small subset of the total claims with cardiovascular disease diagnoses, these claims are a small subset of the claims used to construct the data set of referral sequences and the associated physician network. Therefore, there is no tautological dependency between the referral-sequence and network-based predictors based on the ensemble of cardiovascular care and the treatment outcomes of patients who experienced an AMI. The Medicare claims data record is analyzed for each patient to determine the treatments the patient received post-diagnosis and key follow-up medical events. The dataset has the following key attributes: Bene ID, admission date, death1yr (death or not within one year after index admitted date), PCI (indicator of Percutaneous Coronary Intervention within one year after index admitted date). By matching the AMI admission date with the date of visit to the first physician on a referral sequence for the same beneficiary, we get more than $100,000$ pairs of referral sequences and the corresponding AMI treatment and outcome variables.

The outcome death1yr and treatment PCI are both binary-valued random variables. We collect 69 kinds of features in Table A.2 from referral sequence and patient referral network analysis, which are in six groups: network measures of the dominant HRR on the referral sequence, referral sequence features (e.g., number of nodes, time range), average node positions on the referral sequence, average weights of edges in the national referral network covered by the referral sequence, features of the last physician on the referral sequence (e.g., PageRank value, #cross-PHN referral proposed by the physician), basic patient information (e.g., age).

5.3.2. Methods and Performance

In addition to the seven traditional classification models (LR, KNN, SVM, DT, RF, GBDT, AdaBoost [2]), we try to boost the performance of classification with the following methods.

- **Feature engineering.** Encoding categorical attributes, such as specialty of the key physician and the month of admission date. Features are extracted using both the exact matching referral sequence with the AMI record and the immediately preceding referral sequence within the 90 day period before the exact matching one, in order to capture the association between referral sequence features and subsequent treatment outcomes.

- **10-fold cross validations**. Accomplished by partitioning the original sample into a training set and a test set in rotation.

- **Undersampling.** Undersample some training cases to balance the ratio of positive/negative in training set.

- **Feature selection.** Apply Random Forest (RF) to sort features by their importance [Genuer et al., 2010], and pick up a subset of important features for classification models. Here the importance of a given feature is the increase in the mean error of a tree in the forest when the observed values of this feature are randomly permuted.

- **Voting for the final label.** Collect prediction result of each classification model and vote for the final prediction result of a test case.

[2] Logistic Regression; K-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, Gradient Boosting Decision Tree, Adaptive Boosting

- **Xgboost [Chen and Guestrin, 2016].** Upgrade the gradient boosting model from GBDT to Xgboost, which aims to strengthen regularization of trees and control overfitting.

GBDT has the highest F-score with its performance depicted in Table 5.5 for each year and outcome. Since we can tune parameters in a classification model to get a higher recall or precision, the F-score is more meaningful as an overall evaluation metric. The moderate F-score suggests that a lot of unmeasured variables contribute to treatment decisions and patients' survival. The lack of clinical detail and personal information such as heart rate and blood pressure weakens the power of machine learning models, but the referral sequence features and network measures support the above models to beat random prediction while the accuracy is almost as good as that of other diagnosis classification. A complex convolutional neural network (CNN) model [Fiterau et al., 2017] aims to predict osteoarthritis with much more (600+) directly related features (e.g., clinical measures, joint symptoms/function) and 7-day time series accelerometer sensory data, but the accuracy of baselines and the CNN ranges from 0.633 to 0.789. Table 5.6 shows the average F-score for death1yr and PCI classification on two separate groups divided by age. The power of referral sequence features differs, which means age is an important factor. As predictability does not necessarily imply causality, to attain rigorous causal inferences to the standard typical in medical research would require more study regarding potential confounding variables and possibly involve a randomized study. Moreover, if available, we should group by referral sequences based on clinical tests and demographics, because it will be clear to see the effects of referral sequences among a group of similar patients before treatment.

Table 5.7 shows the top 10 important features for two indicators in 2011, which are selected by the result of RF [Genuer et al., 2010]. For both death1yr and PCI,

Table 5.5: Classification results of GBDT for death1yr and PCI in 2007-2011.

PCI	2007	2008	2009	2010	2011	average F-score
Recall	0.703	0.700	0.702	0.695	0.694	
Precision	0.572	0.574	0.585	0.597	0.607	
F-score	0.631	0.630	0.638	0.642	0.647	0.638
death1yr						
Recall	0.702	0.698	0.710	0.704	0.682	
Precision	0.640	0.632	0.639	0.650	0.633	
F-score	0.669	0.663	0.672	0.675	0.657	0.667

Table 5.6: Average F-score in 2007-2011 of GBDT on groups divided by age.

	death1yr	PCI
Age<=75	0.592	0.695
Age>75	0.687	0.565

average time gap on the referral sequence is one of the most important features. We conjecture that the gap reflects whether the case is serious. In addition, total RVU of physicians on the referral sequence is predictive of death1yr (the patient outcome) and physician position (measured by PageRank) is predictive of PCI (the patient treatment received). The above significant features offer new directions for medical researchers to investigate with their domain knowledge.

GBDT's level of predictive accuracy was on average higher than LR for predicting PCI and higher than LR for predicting death within a year. However, the form of the model from LR is the most amenable to interpreting the model and determining which terms are the most predictive.

5.3.3. Conclusion

By linking AMI treatment and outcome variables to the corresponding referral sequences, we find several informative predictors with either larger feature importance or significant effects, such as the time gap between two visits on the referral sequence and the total RVU of all physicians' endeavors. The novelty of these referral sequence measures suggests that a deeper look into their significance is warranted. We have only just scratched the surface of the enormous potential for

Table 5.7: Top 10 important features for death1yr and PCI generated by Random Forest feature selection method [Genuer et al., 2010].

Rank	death1yr	PCI
1	total RVU of the referral sequence	average time gap on the referral sequence
2	total RVU of the previous referral sequence	indicator of patient's age in 66-70
3	average time gap on the referral sequence	average PageRank values of all physicians on the referral sequence
4	time range of the referral sequence	indicator of the key physician's specialty on the referral sequence as "interventional cardiology"
5	average index of the first-time occurrence on a referral sequence for the last physician	indicator of patient's age in 76+
6	local clustering coefficient of the last physician on the referral sequence	the number of referral sequences that include the last physician
7	times of being the end node on a referral sequence of the last physician on the referral sequence	indicator of the key physician's specialty on the referral sequence as "interventional cardiology"
8	times of being the first node on a referral sequence for the last physician	average #involved sequences among physicians on the referral sequence
9	indicator of patient's age in 76+	average times of being the first node on a referral sequence for all physicians on the referral sequence
10	average times of being the end node on a referral sequence for all physicians on the referral sequence	times of being the first node on a referral sequence for the last physician

using referral sequence features to improve predictions of treatment received and treatment outcomes. Understanding referral sequence patterns has the potential to ultimately help hospitals, physicians and patients towards the ultimate goal of building an optimal referral sequence for each patient with a better treatment outcome and providing the most effective allocation of resources in the network. By replacing the treatment outcome with other variables, it is possible to apply machine learning models on different context of information walk.

Section 5.4 Discussion

In this Chapter, we take the referral network as an example to build three predictive applications about an information walk: direction, outlier and event outcome. They are the central part in the pipeline of a data-driven project. They are general models without the limitation of the context of referral network. Moreover, this Chapter helps with the united organization of this thesis around the target of information walk.

Directions of further research include: an IW direction prediction model considering the effects of other IWs; an extension in the IW network space with the taxonomy of IW and IW network; a well-designed event outcome prediction model with better accuracy.

Chapter 6

Transparency with Network Visualization

In the previous chapters, all of our work relates to the question of prediction for a "walker" on a network. In some contexts the prediction is given directly to the walker as the walker walks. The user may or may not be happy with the prediction, but generally, the user/walker has no ability to modulate the recommendation algorithm. This can be frustrating and inefficient. In this Chapter we address the problem - in part - by considering the advantages of greater transparency in prediction.

We present a proof-of-concept of a visual navigation tool for a personalized "sandbox" of Wiki pages, as an example of transparent application over a network. The navigation tool considers multiple groups of algorithmic parameters and adapts to user activity via graphical user interfaces. The output is a 2D map of a subset of Wikipedia pages network which provides a different and broader visual representation – a map – in the neighborhood (according to some metric) of the pages around the page currently displayed in a browser. The representation schema includes the incorporation of a kind of transparency in the algorithmic parameters affecting the presentation of the landscape visualization, which in turn enables the delivery of a personalized canvas, designed by the user. A case study shows the combination of four different sourcing (i.e., identification and extraction of the neighboring pages) rules and three layouts over the same Wikipedia subnetwork. The basic schema is readily adapted to other search experiences and contexts. The framework of transparent visualization in this Chapter has already appeared in the refereed publication [An and Rockmore, 2019]. In this thesis, Chapter 6 represents the step of evaluation and improvement after we build a data-driven application. The transparent framework aims to provide better user experience, but we can also propose other tasks depending on the details of a project.

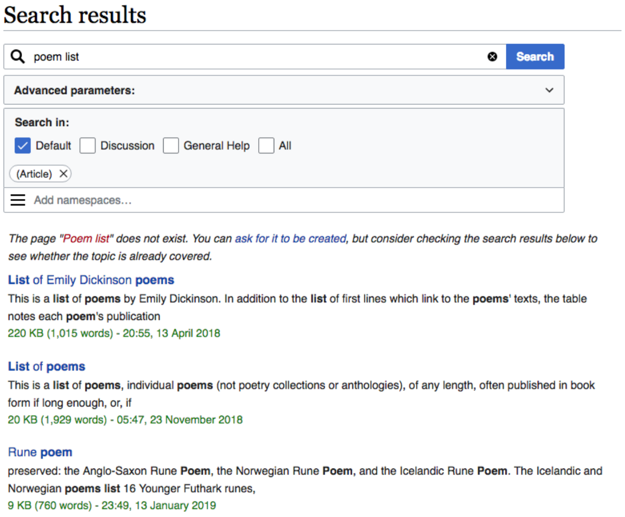
Section 6.1
Visualization of Wiki Pages

6.1.1. Introduction

Wikipedia is an important source of information [Thompson and Hanley, 2018]. For many people, going to Wikipedia is just the first step in an information search task. A standard search trajectory would then take place, realized as a sequence of clicks, effectively something of a constrained, yet still “random” walk from the Wikipedia page starting point, alike at least in spirit to the “random surfer” model that gave birth to PageRank [Brin and Page, 1998] (and Google) whose depth and penetration of the space of relevant webpage resources can and does depend on many contingent characteristics. Regardless of the starting point, in this click-by-click revealing of the relevant (one hopes!) knowledge, it may be easy to miss or get distracted away from the original motivation for inquiry. More broadly, in such a blinded navigation the user is unaware of the way in which the webpages she visits relate to one another. Inadvertently she may be stuck in cul-de-sac of narrowly defined information or strayed very far from her initial search goal. It is with this in mind, that we take on and suggest an alternate option, one that promotes a notion of visual search, that presents a map-like visual summary of a general candidate item (e.g., a Wiki page) neighborhood, thereby possibly promoting a broader field of vision in a search engine or recommender system and highlighting different criteria for navigation.

We propose a visual navigation tool for Wikipedia based network visualization, which allows users to select their preferred query target as the root page, and visualizes the local "sandbox" of related pages in the form of a 2D map on a "canvas" (viewing platform). Our motivation arises from the user experience of standard query on Wikipedia. Figure 6.1 shows a list of related pages when the query input is not exactly matched to a Wikipedia page. More often, Wikipedia will load a new page in the browser or redirect it to a similar one as Figure 6.2. We believe a visual navigation tool might be a useful broadening of our verified knowledge boundary during browsing better than such a list of results or unexpected redirection.

For example, a series of automatically updated maps of the surrounding pages in the network space could display a broader view of the information space that both illustrate the distance among those nodes (pages) while also providing some sense of context for the material on the page. A visual navigation tool based on Wiki page networks could also facilitate a user's understanding of the local network structure, and would bring more transparency to the query results. While the network structure articulates the link relationships between pages, the use of other kinds of metadata (from the user and other users as well from the webpages) raises the possibility of creating a non-link distance structure (metric) for the neighborhood, and with that, new possibilities for display and user interaction. User response to the 2D Wiki map might also offer interaction data for user-behavior oriented research projects. While the focus of this article is on the Wikipedia environment, the general framework of user-controlled network navigator is not limited to Wikipedia corpus. For example,



Search results

poem list

Advanced parameters:

Search in:

☑ Default □ Discussion □ General Help □ All

(Article) ✕

Add namespaces...

The page "Poem list" does not exist. You can ask for it to be created, but consider checking the search results below to see whether the topic is already covered.

List of Emily Dickinson **poems**

This is a **list** of **poems** by Emily Dickinson. In addition to the **list** of first lines which link to the **poems**' texts, the table notes each **poem**'s publication

220 KB (1,015 words) - 20:55, 13 April 2018

List of **poems**

This is a **list** of **poems**, individual **poems** (not poetry collections or anthologies), of any length, often published in book form if long enough, or, if

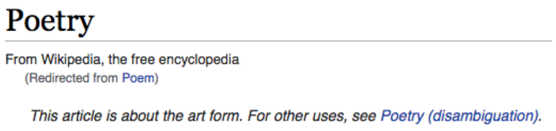
20 KB (1,929 words) - 05:47, 23 November 2018

Rune **poem**

preserved: the Anglo-Saxon Rune **Poem**, the Norwegian Rune **Poem**, and the Icelandic Rune **Poem**. The Icelandic and Norwegian **poems list** 16 Younger Futhark runes,

9 KB (760 words) - 23:49, 13 January 2019

Figure 6.1: Interfaces of query results on Wikipedia. Query result: a list of related Wikipedia pages.



Poetry

From Wikipedia, the free encyclopedia

(Redirected from Poem)

This article is about the art form. For other uses, see Poetry (disambiguation).

Figure 6.2: Interfaces of query results on Wikipedia. Redirection to a new page.

the dynamic graph visualization may also work as a recommender for online shopping or the World Wide Web as a whole.

We present some initial ideas around the design of a personalized visual navigation system on Wikipedia. Generally, the data flow starts from a seed Wikipedia page. A "sandbox" of related pages is defined by a distance threshold on the Wikipedia page network, in which a directed edge from page A to page B means page A cites B in the HTML source file. The navigator will interact with users to get the desired algorithmic parameters of the visualization to be personalized. Behind the user interface, diverse algorithms implement the tasks of node filtering, coordinate computation and edge selection due to a limited size of screen. Though researchers have intended to diversify the user experience of Wiki with visual effects [De Sabbata et al., 2015, Odor et al., 2018, Sáez and Hogan, 2018], our contributions include the possibility of real-time updated visual navigation that responds to users browsing in the open space of Wikipedia pages, and the fact that nodes on the screen are determined by the personalized algorithmic parameters directly set by users. As users are often only aware of the pages that comprise their browsing paths only relatively "blind" to any "surrounding" ones, we hope our design of such an immersive visual navigation would make for a more useful Wikipedia search experience, as well as for many other transparent recommender systems.

6.1.2. Data Pipeline

In this section, we present the data pipeline starting from a seed Wikipedia page to the visualization of related pages. The following steps combine user interface and algorithm-based computation for a personalized and transparent visualization. In this case, transparency means that users would know how the thing they are looking at is made, while customizability means that users have the power to directly change the input parameters in our proposed navigation system. We hope that with that transparency users will find it more useful and thus engage with it more, etc. in a productive feedback that will both enable deep exploration as well as free and broad exploration (i.e., "exploding" the filter bubble).

Wikipedia seed selection. The "seed page" represents the user-defined center or starting point for a neighborhood of Wikipedia pages of interest for a given topic.

Wikipedia page crawler. If a Wikipedia page cites another one in the main context, they are a pair of linked nodes in the network of Wikipedia pages. Among the billions of Wikipedia pages, in this preliminary proof-of-concept study we work with only a very small subset and limit the range of the crawler with some threshold on the distance between a candidate page to the seed.

Parse sandbox structure. Given a seed page, the downloaded subset of Wikipedia "nearby" pages is our "sandbox". Our processing and analysis do not edit their content. This step aims to build the network of the observed local

Wikipedia pages around the seed. There are various options for thresholding the neighborhood (e.g., all pages within some fixed linked distance of the seed). A Wikipedia (sub)-network not only contains the surrounding nodes, but also the edges among them. Here we define the weight of an edge between two nodes as the number Wikipedia pages that cite both of them. For a given node, this weight enables a sorting of its direct neighboring nodes (i.e., with a distance of one) in the network.

Set algorithmic parameters. Different from traditional fancy digital-art based user interface (UI) design, here we propose a framework for algorithmic visualization for a (sub-) network of Wikipedia pages. It contains three groups of parameters, set by users, to make the visualization more transparent:

(1) The rule of nodes sourcing and ranking. Here we apply four different methods:

   – Semantic content-based similarity

   – Graph structure

   – Collaborative filtering of users browsing

   – An overall *PageDist* (cf., [Leibon et al., 2018]) metric derived from link and content similarity

   The navigation tool could display a limited number of nodes within a canvas, compute the internode distance matrix and then use that for node placement/visualization. For example, to measure content similarity we compute the distance between two Wikipedia page titles according to the word vector representation GloVe [Jeffrey Pennington, 2014]. It is also possible to sort the neighboring nodes with some network science features (e.g., node degree, PageRank centrality). To simulate collaborative filtering, we assume that the frequency of concurrence on a third page is proportional to the probability of users preference for the two pages. The PageDist [Leibon et al., 2018] metric considers the commute distance [Yen et al., 2005] in a transition matrix which is derived from both in-out link structures and semantic similarity of texts. All the sourcing methods are independent with the link structure of the downloaded pages in the sandbox. In the previous step of crawling, we have applied a distance threshold. If the navigator serves the whole Wikipedia network without a radius in the crawling step, those sourcing methods could narrow down the range of candidates as well.

(2) Definition of "nearby" nodes on the canvas. According to the selected sourcing and ranking method for a small number of candidate nodes to display with the seed on a canvas, we could sort all surrounding nodes according to their feature similarity to the root page. Therefore, for any two nodes appearing on the canvas, their relative proximity to the center node would be in accordance with their rankings in the sorted list of their feature similarity to the root page. Another definition of proximity comes from the result of a user preference predictive model, where the neighboring nodes with a larger probability of preference will be closer to the center node. Those two settings may be in accordance with each other, but sometimes a user might explore some new and highly dissimilar pages rather than the most similar one. This might be especially true when looking for information about particularly divisive or "charged" subjects.

(3) Layout of nodes. We try to locate the current Wikipedia page at the center of a canvas, except in the case of using the 2D multidimensional scaling [Cox and Cox, 2000] (MDS). If the second setting (i.e., closeness to the center) is defined by the feature similarity from the sourcing rule, the surrounding nodes should follow the order of their distances to the root page in the feature vector space. We implement spiral and spectral layouts to adapt to a ranking of the selected

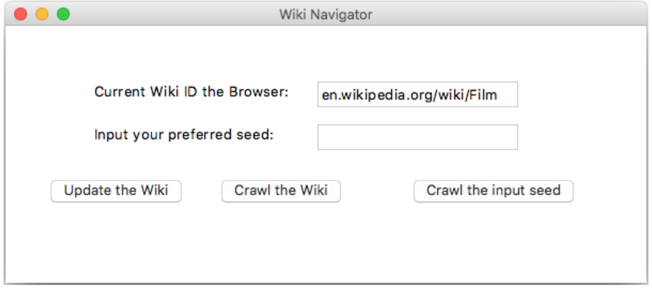
nodes. Both layouts point out the “close” neighbors and grant users the access to adjust parameters for their desired neighboring nodes. The assigned node coordinates in MDS match the idea of preserving between-node distances rather than the arbitrary design of spiral/spectral layouts.

Visualization. As a part of back-end algorithmic visualization, several factors might limit the actual effects, such as the size of the available screen (i.e., “canvas”) to present the Wikipedia network, the number of pixels in a fixed size canvas (i.e., resolution), and the suitable number of nodes/edges. In addition, the location of a node should follow the general direction defined by the layout. Therefore, except for the MDS option, we first compute coordinates in a polar system, then transform it to the 2D plane coordinates. This step needs the help of an external visualization package which places nodes on a 2D plane at the accurate coordinates, so that users could present a non-standard yet desired layout on a canvas. Though many algorithmic terms are introduced in this tool, non-expert users could compare the differences in visualization and – with a little experience and/or training – adjust parameters for their preferred result.

Update the Sandbox. A transparent navigation system could incorporate user activities, such as hyperlink clicks, revisiting a page or long-time browsing. Once monitoring the above activity, the system should return to the second step to crawl some new Wikipedia pages, and update the Wikipedia network with the following steps, such as a new seed page and new selected neighbors. In this way, the navigation tool could extend to an open Wikipedia space and gradually collect user preference records for other personalized services on Wikipedia.

6.1.3. System Implementation

In this section, we briefly introduce the implementation of a proof-of-concept visual navigation system for Wikipedia pages which follows the data pipeline. It is developed in Python to take advantage of multiple efficient existing programming packages. Selenium [Muthukadan, 2018] enables the detection of the current URL in a browser. Tkinter [Lundh, 2019] offers the UI modules (e.g., input frames and radio buttons) for Wikipedia seed confirmation in Figure 6.3 and algorithmic parameters settings in Figure 6.4. With the input of a seed Wikipedia page, Urllib [Python-Software-Foundation, 2019] downloads all the cited Wikipedia hyperlinks in the seed page with the help of a regression expression matching function. BeautifulSoup [Richardson, 2018] facilitates the analysis of hyperlinks in local HTML files so that we could build the network of Wikipedia pages in the "sandbox". NetworkX [NetworkX-Developers, 2017] could place a node at the given coordinates in a 2D plane, so the navigator displays the same layout as what users choose (see Figure 6.4). Here we show examples of the visualization.



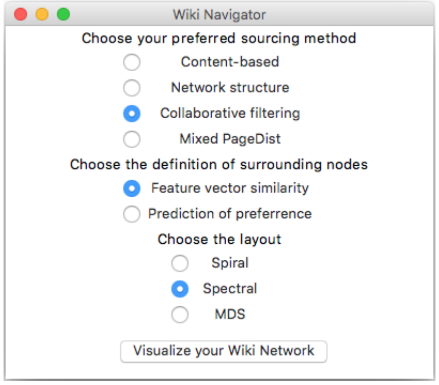
Wiki Navigator

Current Wiki ID the Browser: en.wikipedia.org/wiki/Film

Input your preferred seed:

Figure 6.3: The user interface of Wikipedia seed selection. Users could input a seed or select the current one in a browser.

The implementation we describe above thus assumes an offline deployment to determine a subset of edges and links on the screen. Extensions of this simple approach may include (1) a much wider range of online Wikipedia pages around the seed page (2) a combination of more advanced algorithmic settings without too much time cost.



Wiki Navigator

Choose your preferred sourcing method

○ Content-based
○ Network structure
● Collaborative filtering
○ Mixed PageDist

Choose the definition of surrounding nodes

● Feature vector similarity
○ Prediction of preferrence

Choose the layout

○ Spiral
● Spectral
○ MDS

Visualize your Wiki Network

Figure 6.4: The user interface of algorithmic parameter selection for network visualization.

Therefore, an upgraded version of Wikipedia navigation might be an online application deployed on a powerful server to execute the data pipeline fast. As a starting point, the proof-of-concept satisfies the proposed requirements, and several packages could be reused in the advanced version, too.

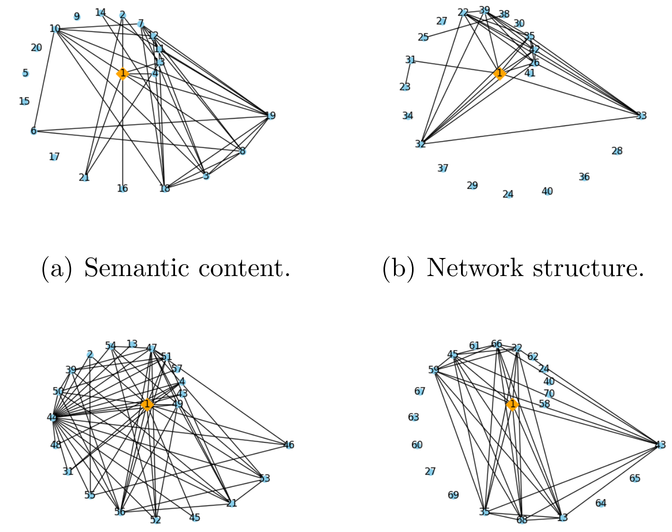
6.1.4. Case Study

To illustrate the diverse kinds of 2D maps for Wikipedia navigation, we take the Wikipedia page "Film" as the seed, and crawl all its direct neighbors at depth one, all of which are cited on the "Film" page. We set a threshold of 100 on the node degree (i.e., the number of links it has) to get a denser network with 8,083 directed edges and 151 nodes (pages). For a clear network visualization we only select top 20 neighboring nodes according to their feature similarity (depends on the choice of sourcing rule) to the seed node ("Film"), and display the top one-third of edges among those selected edges based on the edge weight defined by the times of concurrences on a third page.

Since we apply four kinds of sourcing and ranking methods introduced as the first group of algorithmic parameters, in total there are 80 nodes selected for all the maps, but some nodes might be selected by multiple sourcing rules. Table 6.1 shows a dictionary of them. Since an accurate user preference prediction requires real user

1	Film	36	Classical_Hollywood_cinema
2	Screenplay	37	Cult_of_personality
3	Documentary_film	38	Public_relations
4	Television	39	Principal_photography
5	Film_production	40	Color_motion_picture_film
6	Film_genre	41	Spectacle_(critical_theory)
7	Short_film	42	Script_breakdown
8	Art_film	43	Videography
9	Movie_studio	44	Main_Page
10	Independent_film	45	Film_industry
11	Sound_film	46	Cinematography
12	Silent_film	47	Special_effect
13	Soundtrack	48	Internet
14	Science_fiction_film	49	Visual_effects
15	Film_history	50	Post-production
16	Film_director	51	Storyboard
17	Film_editor	52	Film_score
18	Feature_film	53	Film_crew
19	Animation	54	Sound_effect
20	Film_release	55	Guerrilla_filmmaking
21	Film_editing	56	Filmmaking
22	Pitch_(filmmaking)	57	Streaming_media
23	Digital_object_identifier	58	American_Dream
24	Concentration_of_media_ownership	59	Film_treatment
25	News_broadcasting	60	Media_event
26	Shooting_schedule	61	Docufiction
27	Occupation_(protest)	62	Culture_industry
28	Cinema_of_the_United_States	63	Managing_the_news
29	Crowd_manipulation	64	Strike_action
30	Recuperation_(politics)	65	United_States
31	International_Standard_Book_Number	66	Daily_progress_report
32	Daily_production_report	67	Mainstream_media
33	Breaking_down_the_script	68	Screenwriting
34	Demonstration_(protest)	69	Political_satire
35	Roadshow_theatrical_release	70	Bollywood

Table 6.1: Dictionary of the nodes selected by four sourcing methods.



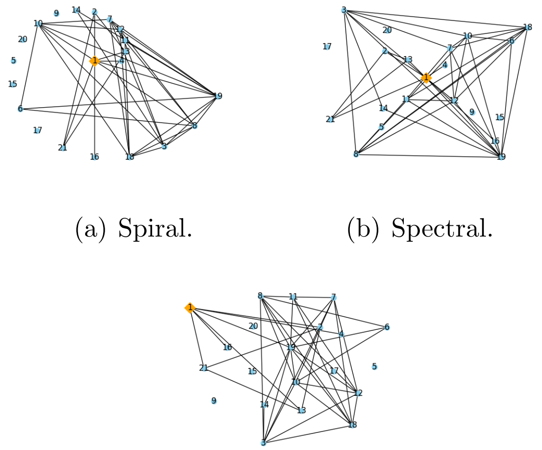
(a) Semantic content.

(b) Network structure.

(c) Collaborative filtering.

(d) A mixed PageDist [Leibon et al., 2018] metric.

Figure 6.5: 2D map visualization of different sourcing methods. The distance to Node 1 is derived from feature similarity. The common layout is spiral. An orange diamond represents the root page.



(a) Spiral.

(b) Spectral.

(c) MDS.

Figure 6.6: 2D map visualization under different layouts. The distance to Node 1 is derived from feature similarity. The common sourcing method is semantic content. An orange diamond represents the root page.

behavior data, we choose the second algorithmic parameter as “the distance based on feature vector similarity” instead of a user preference prediction.

As for layout options, we take the coordinates directly generated by MDS and compute spiral and spectral coordinates in a polar system, respectively. MDS exploits a pairwise distance matrix to present a sense of how near or far points are from each other in a low dimensional space (e.g., 2D plane) to users. The spiral and spectral layouts tend to prove that users may choose their personalized layouts beyond the traditional MDS visualization method, and the navigation system is flexible enough to support the function. In total, we exploit the navigation system to generate the enumerations of available sourcing-ranking methods and layouts, some of which are displayed in Figure 6.5 and Figure 6.6.

Figure 6.5 illustrates the spiral 2D maps of Wikipedia nodes according to four different sourcing methods. Their common algorithmic parameters suggest that the distance to the center node (“Film”) corresponds to the ranking of their feature similarity to that of the Node “Film”. For example, in Figure 6.5(a), the semantic content method treats Node 4 (“Television”) as the most similar neighbor to “Film”, and the second one is Node 13 (“Soundtrack”). The farthest neighbor is Node 19 “Animation”. In Figure 6.5(b), according to some network science feature (e.g., degree of a node within the sandbox), the most significant two nodes are “Spectacle_(critical_theory)” and “Shooting_schedule” (a daily plan of film production). For the collaborative filtering map in Figure 6.5(c), “Visual_effects” and “Videography” occupy the nearest two locations to the center. In the PageDist map (Figure 6.5(d)), “American_Dream” and “Bollywood” become the nearest neighbors. Users would recognize the obvious differences among the maps and choose their desired method for the following browsing.

With a limit of 20 or so nodes to a canvas in Figure 6.5, the four derived node sets have almost no intersection. That is, the different metrics produce very different neighborhoods in terms of their underlying node sets. If we use a larger bound of 50 nodes on a given canvas, the semantic-content set and network structure set have 10 nodes in common, the semantic-content set and collaborative filtering set share 16 nodes, while the intersection of collaborative filtering and PageDist contains 13 nodes. Going further, the first three sets (semantic, network and collaborative), have five nodes in common "Film_budgeting", "Cinematography", "Roadshow_theatrical_release", "Film_industry", "Principal_photography". The diverse navigation maps will have varying levels of utility to different user groups.

Figure 6.6 displays three layouts of the same subset of nodes according to the semantic content sourcing method, with the condition that the similar nodes of "Film" would be placed close to the center. For the MDS one (Figure 6.6(c)), the coordinates of all nodes are derived from a similarity matrix so that the node "Film" may not be at the center of the canvas. More importantly, MDS considers the mutual similarity between any pair of nodes on the canvas, while for the other layouts, the comparative distance is only meaningful between the root node "Film" and another node.

Since only the edges with a large enough weight could be added to the map, the dense edges suggest several local clusters, such as Nodes (4, 13, 11, 12, 7), or another group (18, 3, 8, 19) in Figure 6.6(a). Besides, the spiral layout clearly shows the similarity-based distance to the center node in an anti-clockwise order. For the second spectral layout in Figure 6.6(b), we allocate the nodes mainly in four directions (upper right, upper left, down right, down left). It might be more difficult to compare the distance to the center for two nodes (e.g., Nodes 9 and 14), but the spectral layout makes it possible to cluster the neighboring nodes into several groups and deploy each group along a "beam". In Figure 6.6(c), the MDS layout considers the distance matrix of all nodes in terms of the semantic vector of the corresponding Wikipedia page's title and computes their coordinates with a standard dimensionality reduction algorithm, so the root page "Film" is automatically placed in the upper left corner.

In this way, without the special color/shape, it might not attract users attention at the first glance. MDS is a popular standard visualization method, but when users choose the second algorithmic parameter about closeness in the navigator as "a probability from a predictive model", it is more difficult to define a complete distance matrix, especially between pairs of surrounding nodes.

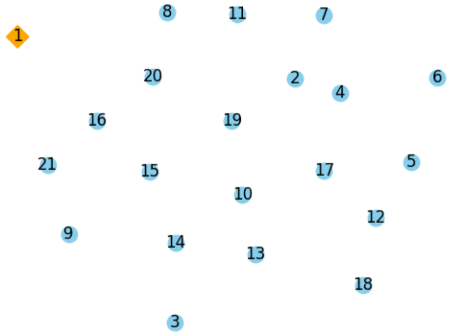


Figure 6.7: A non-edge version of Figure 6.6(c) with MDS.

Beyond the above algorithmic parameters and layout options, other visualization factors may be critical. Figure 6.7 displays the non-link version of MDS layout, in which the neighborhood is determined by a textual distance instead of link-based distance on the subnetwork. We would anticipate associating such a non-edge map with some kinds of "sliders" that would allow the picture to vary according to user feedback.

6.1.5. User Study

The user study contains two parts: iterative maps and personalized browsing. In the first stage, we will present the iterations of all possible maps over all algorithmic settings with a fixed root page ("Film"). After that, we grant users the access to tune

Table 6.2: Average NDCG and transparency score for each group of parameters, under the iteration mode with the fixed root page.

sourcing	NDCG	transparency
Content-based	0.584	3.752
Network structure	0.662	3.665
Collaborative filtering	0.63	3.633
Mixed PageDist	0.575	3.542

neighbor	NDCG	transparency
Feature vector similarity	0.5	3.534
Prediction of preference	0.726	3.761

layout	NDCG	transparency
Spiral	0.637	3.884
Spectral	0.677	3.795
MDS	0.524	3.268

the parameters for their preferred settings, and the root page will change along with their preference.

For each map, users will answer two questions:

- Choose at least three preferred nodes for browsing. Here we take Normalized Discounted Cumulative Gain (NDCG) [Valizadegan et al., 2009] to evaluate whether the ranking of surrounding nodes matches users selection.

- Evaluate the transparency of the visualization framework based on the current map. An integer from 1 to 5 refers to the degree users believe that the map shows more transparency than a plain list. The value of 1 means "strongly disagree", value of 2 means "disagree", value of 3 means "neutral", while value of 4 and 5 suggests positive feedback of "agree" and "strongly agree", respectively.

We invite 30+ Dartmouth students to take the user study[1]. Here are the initial results.

[1] They are randomly picked up in study rooms.

Table 6.3: Percentage among all choices made by users, average NDCG and transparency score for each group of parameters, under the personalized browsing mode with dynamic root page.

sourcing	Percentage	NDCG	transparency
Content-based	34.63	0.639	4.194
Network structure	29.61	0.708	4.038
Collaborative filtering	22.91	0.621	3.78
Mixed PageDist	12.85	0.642	3.957

neighbor	Percentage	NDCG	transparency
Feature vector similarity	40.22	0.519	3.93
Prediction of preference	59.78	0.748	4.084

layout	Percentage	NDCG	transparency
Spiral	34.08	0.718	4.066
Spectral	43.58	0.678	4.09
MDS	22.35	0.519	3.825

Table 6.2 shows difference in all three groups of parameters with a fixed root page. About the sourcing algorithm, network structure based method gets the highest NDCG, which suggests the best recommendation result, while content-based method brings the most transparency to users. In terms of the definition of neighboring nodes on the map, the setting of user preference predicted by our algorithm looks better in both NDCG and transparency. For the layout options, the spectral one is the best in terms of NDCG and the spiral one makes more users feel transparency.

Table 6.3 presents users preference when they are able to tune the parameters and feely browse among the sandbox of Wiki pages. About sourcing algorithm, content-base method looks the most popular one with the highest transparency score, while network structure method gets the highest NDCG. For the second group of parameters, prediction of preference beats feature similarity in all three measures. About layout, users prefer to view the map with spectral and feel the most transparency with it.

Table 6.4: Comparison between iteration mode and personalized mode.

	Iteration	Personalized
NDCG	0.613	0.656
transparency	3.648	4.022

Finally, Table 6.4 compares the two modes in user study, whose difference is whether users have the chance to tune parameters and view dynamic maps of nodes with various root pages. We find that personalized mode gets better recommendation results with more transparency.

6.1.6. Conclusion

We have presented a proof of concept for an open navigation tool of Wikipedia pages to broaden the understanding of the information context of Wikipedia pages to a user, along with a form of algorithmic transparency for the users to enable them to better understand why they get the current map of a vast Wikipedia network.

We find that the sourcing and ranking method can significantly affect the set of finally selected nodes on the canvas, and different layouts highlight (according to the different underlying metrics) different significant neighboring nodes in the corresponding local cluster on the map.

The user study validates our design and assumptions that users prefer to browse in the Wiki network with some parameters. With more behavioral data, we would like to apply the BPR-IW model to recommend new pages for users since the track of browsing is also an information walk. Conversely, the visualization framework may also explain referrals to patients if needed. The collected data could also contribute to other related research projects, such as a transparent online advertising/shopping platform. There is also the possibility of an upgraded version of navigation tool merged into a browser (e.g., a Chrome extension) or a back-end deployment on a web server to speed up the Wikipedia page visualization in the whole space of Wiki world, or even upgrade the current text-based browser to a visual-oriented one.

This Chapter works as an evaluation and improvement in a data pipeline, after the predictive modeling in a real data-driven project. Transparency is one of the key points for a better user experience. Our proposed framework of transparent interaction with users points out the next step of experiment and development in a real web/mobile application.

Chapter 7

Conclusion

Data-driven applications built with metadata of a network of users (or even a generalized network of individual items) become a hot topic in both academia and industry. This thesis considers various real problems and challenges in the data pipeline of such a project, which arise from data understanding, feature engineering, model building and user-experience oriented evaluation. The models and methods we propose in this thesis work as independent modules on different contexts, but they are connected by the topic of information walk, which is the general target to represent our specific research targets in this thesis. Here we summarize the original contributions of this thesis and list a few possible directions for further research.

- First, in terms of understanding the available dataset, we propose a generative hierarchical behavior model for phone usage, which targets every user in the corresponding social network.
- Second, in the step of feature engineering, we construct novel geographical features for the community of yelp users, and design a geographical module for local search and business recommendation.

- Third, we define a new type of information sharing in a network, the Information Walk (IW), as well as a high-level network of Information walks. Therefore, we are able to predict the future direction of every ongoing information walk, detect the outliers among all information walks, and predict the outcome of the event along with an information walk.

- Last but not the least, we propose a novel framework to improve the transparency of personalized recommendation, with customizability and feature space visualization during network navigation. The proof-of-concept on Wikipedia pages network gets positive feedback in our initial user study.

Our work described in the previous Chapters suggests the following natural directions for further research.

The generative hierarchical model could explain the user behaviors well, but the evaluation metrics talk about the sum of user behaviors in a period. In the future, a more impactful direction would be an accurate prediction of the time point when some activity happens.

We construct novel features from a geographical database, a kind of publicly available data. Once we have diverse kinds of external datasets, it would cost a lot of time on feature engineering. An autonomous framework of feature engineering on heterogeneous datasets would be appreciated.

In terms of information walk prediction, we assume that all the ongoing information walks are independent. However, in real cases, it is difficult to verify that. For example, if multiple applicants are interviewing with the same company, their next steps on career paths would be affected by others when the number of opening positions is limited. Therefore, an advanced model of information flow prediction should consider multiple information walks together.

About the framework of a transparent recommender, though we implement user study and find initial positive results, it would be desired to recruit more users and deploy the recommender system online, so that we can get a more convincing result.

Appendix A

Appendix

Table A.1: P-values for rejecting various Power Laws, assortativity, self-degree-correlation, reciprocity and clustering coefficient of the national patient referral network (or average among states) in 2009-2015.

Year	2009	2010	2011	2012	2013	2014	2015
in-degree p-value of national network Power Law	1.00	0.99	1.00	1.00	0.93	0.91	0.38
#states in-degree p-value>0.05	32	37	37	36	36	37	36
Average p-value of in-degree Power Law among states	0.4084	0.4084	0.4137	0.4245	0.4423	0.4388	0.4654
out-degree p-value of national network Power Law	1.00	1.00	0.99	0.97	0.00	0.97	0.74
#states out-degree p-value>0.05	39	43	42	37	37	38	40
Average p-value of out-degree Power Law among states	0.4545	0.5292	0.5913	0.5303	0.5190	0.4956	0.4484
Average (in, in) assortativity among states	-0.1084	-0.1083	-0.1101	-0.1126	-0.1132	-0.1137	-0.1217
Average (out, out) assortativity among states	-0.1104	-0.1108	-0.1125	-0.1150	-0.1157	-0.1161	-0.1245
Average (in, out) assortativity among states	0.0775	0.0752	0.0727	0.0692	0.0662	0.0633	0.0549
Average (out, in) assortativity among states	0.0800	0.0775	0.0750	0.0714	0.0684	0.0654	0.0569
State self in/out degree: average R-squared value	0.9717	0.9715	0.9712	0.9717	0.9710	0.9711	0.9692
State self in/out degree: average correlation coefficient	0.9858	0.9856	0.9855	0.9857	0.9853	0.9854	0.9845
State reciprocity: average R-squared value	0.9074	0.9094	0.9073	0.9053	0.9045	0.9015	0.8927
State reciprocity: average correlation coefficient	0.9524	0.9535	0.9524	0.9513	0.9509	0.9493	0.9445
global clustering coefficient of national network	0.0763	0.0740	0.0727	0.0682	0.0623	0.0609	0.0523
local clustering coefficient of national network	0.700	0.699	0.698	0.698	0.698	0.699	0.691
E(C) by Erdós-Renyi Model of national network	1.27e-4	1.23e-4	1.18e-4	1.13e-4	1.06e-4	1.02e-4	7.54e-5

Table A.2: Feature list of a referral sequence for treatment outcome classification.

Group of Features	Features and ID
Network measures in the dominant HRR	1:#nodes, 2:#edges, 3:indegree gini coefficient, 4:outdegree gini coefficient, 5:indegree power law test alpha, 6:outdegree power law test alpha, 7: diameter, 8:global clustering coefficient, 9:local clustering coefficient, 10: (in, in) assortativity, 11:self in/out degree coefficient, 12:referral reciprocity, 13:RVU reciprocity
Referral sequence	14:#nodes, 15:average time gap, 16: time range, 17:indicator of recurrence, 18: #nodes before recurrence, 19:physician distribution entropy, 20: PHN distribution entropy, 21:HRR distribution entropy, 22:average #common connected nodes between neighbors, 23:#pairs of nodes with reciprocal referrals, 37:#change points, 38:#previous referral sequence in the same year, 39:distance between the first visited hospital and the end one, 40:total RVU, 41:month of the first visit, 42:#visited teaching hospitals, 43:specialty of the key physician, 44:specialty of the last physician, 45:#visited PHN with negative (in-out) degree on PHN traffic map, 46:#visited PHN with positive (in-out) degree on PHN traffic map, 47:sum of (in-out) degree for all PHN on the referral sequence, 60:indicator of admitted by emergency department for the first node
Average node positions on the referral sequence	24:local clustering coefficient, 25:PageRank, 26:h-index, 27:#sequences which contains the node, 28:#sequences where the node is the starting one, 29:#sequences where the node is the end one, 30:index of the first-time occurrence, 31:#sequences where the node occurs multiple times, 32:#cross-HRR referrals proposed by the node, 33:#cross-PHN referrals proposed by the node
Average weights of edges covered by the sequence	34:#referrals, 35:RVU, 36:ranking based weight
Last physician on the referral sequence	48:RVU, 49:month of visit, 50:local clustering coefficient, 51:PageRank, 52:h-index, 53:#sequences which contains the node, 54:#sequences where the node is the starting one, 55:#sequences where the node is the end one, 56:average index of the first-time occurrence, 57:#sequences where the node occurs multiple times, 58:#cross-HRR referrals proposed by the node, 59:#cross-PHN referrals proposed by the node
Patient history information	61:age, 62:indicator of HIV, 63:indicator of asthmatic lung disease, 64:indicator of cancer, 65:indicator of dementia, 66:indicator of diabetes, 67:indicator of liver disease, 68:indicator of chronic non-asthmatic lung disease, 69:indicator of chronic renal disease

Bibliography

[Adamic and Adar, 2003] Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3):211–230.

[Ahlers, 2013] Ahlers, D. (2013). Business entity retrieval and data provision for yellow pages by local search. In *IRPS Workshop*.

[Albert et al., 2000] Albert, R., Jeong, H., and Barabasi, A. (2000). Error and attack tolerance of complex networks. *Nature*, 406:378–382.

[Amaral et al., 2000] Amaral, L. A. N., Scala, A., Barthelemy, M., and Stanley, H. E. (2000). Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152.

[An et al., 2018a] An, C., O'Malley, A. J., Rockmore, D. N., and Stock, C. D. (2018a). Analysis of the us patient referral network. *Statistics in medicine*, 37(5):847–866.

[An et al., 2018b] An, C., OMalley, A. J., and Rockmore, D. N. (2018b). Referral paths in the us physician network. *Applied Network Science*, 3(1):20.

[An et al., 2018c] An, C., OMalley, A. J., and Rockmore, D. N. (2018c). Walk prediction in directed networks. In *International Conference on Complex Networks and their Applications*, pages 15–27. Springer.

[An et al., 2019] An, C., OMalley, A. J., and Rockmore, D. N. (2019). Towards intelligent complex networks: the space and prediction of information walks. *Potentially accepted by the Journal of Applied Network Science.*

[An and Rockmore, 2016a] An, C. and Rockmore, D. (2016a). Improving local search with open geographic data. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 635–640. International World Wide Web Conferences Steering Committee.

[An and Rockmore, 2016b] An, C. and Rockmore, D. (2016b). Predicting phone usage behaviors with sensory data using a hierarchical generative model. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 75–87. Springer.

[An and Rockmore, 2019] An, C. and Rockmore, D. (2019). Open personalized navigation on the sandbox of wiki pages. In *the Proceedings of the Web Conference, to appear.*

[Balaraman et al., 2018] Balaraman, V., Razniewski, S., and Nutt, W. (2018). Recoin: Relative completeness in wikidata. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1787–1792. International World Wide Web Conferences Steering Committee.

[Bao et al., 2012] Bao, P., Hecht, B., Carton, S., Quaderi, M., Horn, M., and Gergle, D. (2012). Omnipedia: bridging the wikipedia language gap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1075–1084. ACM.

[Barabási et al., 2016] Barabási, A.-L. et al. (2016). *Network science.* Cambridge university press.

[Barnett et al., 2011] Barnett, M. L., Christakis, N., and Landon, B. (2011). Mapping physician networks and their association with health care delivery in us hospitals (md thesis). *Harvard Medical School*.

[Barnett et al., 2012a] Barnett, M. L., Christakis, N. A., OMalley, A. J., Onnela, J.-P., Keating, N. L., and Landon, B. E. (2012a). Physician patient-sharing networks and the cost and intensity of care in us hospitals. *Medical care*, 50(2):152.

[Barnett et al., 2012b] Barnett, M. L., Keating, N. L., Christakis, N. A., OMalley, A. J., and Landon, B. E. (2012b). Reasons for choice of referral physician among primary care and specialist physicians. *Journal of general internal medicine*, 27(5):506–512.

[Berberich et al., 2011] Berberich, K., König, A. C., Lymberopoulos, D., and Zhao, P. (2011). Improving local search ranking through external logs. In *Proc. of the SIGIR*, pages 785–794. ACM.

[Borgatti and Everett, 2000] Borgatti, S. P. and Everett, M. G. (2000). Models of core/periphery structures. *Social networks*, 21(4):375–395.

[Bourigault et al., 2014] Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014). Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 393–402. ACM.

[Bourigault et al., 2016] Bourigault, S., Lamprier, S., and Gallinari, P. (2016). Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 573–582. ACM.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In Crouch, M. and Lindsey, T., editors, *Computer Networks And ISDN Systems*, pages 107–117. Elsevier.

[Buffa and Gandon, 2006] Buffa, M. and Gandon, F. (2006). Sweetwiki: semantic web enabled technologies in wiki. In *Proceedings of the 2006 international symposium on Wikis*, pages 69–78. ACM.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

[Choi et al., 2016] Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016). Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM.

[Church and Smyth, 2008] Church, K. and Smyth, B. (2008). Who, what, where & when: a new approach to mobile search. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 309–312. ACM.

[Clauset et al., 2009] Clauset, A., Shalizi, C., and ME, N. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

[CMS,] CMS. Physician shared patient datasets. `https://questions.cms.gov/faq.php?faqId=7977`. Accessed Sept 1, 2016.

[Cosley et al., 2007] Cosley, D., Frankowski, D., Terveen, L., and Riedl, J. (2007). Suggestbot: using intelligent task routing to help people find work in wikipedia. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 32–41. ACM.

[Cox and Cox, 2000] Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.

[De Sabbata et al., 2015] De Sabbata, S., Çöltekin, A., Eccles, K., Hale, S., and Straumann, R. (2015). Collaborative visualizations for wikipedia critique and activism. In *Ninth International AAAI Conference on Web and Social Media*.

[Dragut et al., 2014] Dragut, E. C., Dasgupta, B., Beirne, B. P., Neyestani, A., Atassi, B., Yu, C., and Meng, W. (2014). Merging query results from local search engines for georeferenced objects. *ACM Transactions on the Web (TWEB)*, 8(4):20.

[Efron, 1992] Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in Statistics*, pages 569–593. Springer, Germany.

[Erdós and Renyi, 1959] Erdós, P. and Renyi, A. (1959). On random graphs, i. *Publ. Math*, 6:290–297.

[Eswaran and Faloutsos, 2018] Eswaran, D. and Faloutsos, C. (2018). Sedanspot: Detecting anomalies in edge streams. In *2018 IEEE ICDM*, pages 953–958. IEEE.

[Farrahi and Gatica-Perez, 2010] Farrahi, K. and Gatica-Perez, D. (2010). Probabilistic mining of socio-geographic routines from mobile phone data. *IEEE Journal of Selected Topics in Signal Processing*, 4(4):746–755.

[Faust, 2010] Faust, K. (2010). A puzzle concerning triads in social networks: Graph constraints and the triad census. *Social Networks*, 32(3):221–233.

[Fiterau et al., 2017] Fiterau, M., Bhooshan, S., Fries, J., Bournhonesque, C., Hicks, J., Halilaj, E., Ré, C., and Delp, S. (2017). Shortfuse: Biomedical time series representations in the presence of structured information. *ArXiv preprint ArXiv:1705.04790*.

[Flöck et al., 2015] Flöck, F., Laniado, D., Stadthaus, F., and Acosta, M. (2015). Towards better visual tools for exploring wikipedia article development–the use case of gamergate controversy. In *Ninth International AAAI Conference on Web and Social Media*, pages p48–55.

[Gan et al., 2008] Gan, Q., Attenberg, J., Markowetz, A., and Suel, T. (2008). Analysis of geographic queries in a search engine log. In *LocWeb Workshop*, pages 49–56. ACM.

[Genuer et al., 2010] Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236.

[Geopy-1.11.0,] Geopy-1.11.0. Python geocoding toolbox. `https://pypi.python.org/pypi/geopy/`. Accessed Dec.15, 2015.

[Ghoshal et al., 2009] Ghoshal, G., Zlatić, V., Caldarelli, G., and Newman, M. (2009). Random hypergraphs and their applications. *Physical Review E*, 79(6):066118.

[Gomez-Rodriguez et al., 2011] Gomez-Rodriguez, M., Balduzzi, D., and Schölkopf, B. (2011). Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 561–568. Omnipress.

[Gundala and Spezzano, 2018] Gundala, L. A. and Spezzano, F. (2018). Readers demanded hyperlink prediction in wikipedia. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1805–1807. International World Wide Web Conferences Steering Committee.

[Harder et al., 2017] Harder, R. H., Velasco, A., Evans, M., An, C., and Rockmore, D. (2017). Wikipedia verification check: A chrome browser extension. In *Proceedings* *of the 26th International Conference on World Wide Web Companion*, pages 1619–1625. International World Wide Web Conferences Steering Committee.

[He et al., 2017] He, R., Kang, W.-C., and McAuley, J. (2017). Translation-based recommendation. In *Proceedings of the 11th RecSys Conference*, pages 161–169. ACM.

[He et al., 2012] He, X., Song, G., Chen, W., and Jiang, Q. (2012). Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 463–474. SIAM.

[Hirsch, 2005] Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *PNAS*, 102(46):16569.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Jaccard, 1901] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579.

[James et al., 2018] James, C., Pappalardo, L., Sirbu, A., and Simini, F. (2018). Prediction of next career moves from scientific profiles. *arXiv preprint arXiv:1802.04830*.

[Jeffrey Pennington, 2014] Jeffrey Pennington, Richard Socher, C. D. M. (2014). Glove: Global vectors for word representation.

[Kalogeraki et al., 2002] Kalogeraki, V., Gunopulos, D., and Zeinalipour-Yazti, D. (2002). A local search mechanism for peer-to-peer networks. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 300–307. ACM.

[Kashima and Abe, 2006] Kashima, H. and Abe, N. (2006). A parameterized probabilistic model of network evolution for supervised link prediction. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 340–349. IEEE.

[Kempe et al., 2003] Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM.

[Kimura and Saito, 2006] Kimura, M. and Saito, K. (2006). Tractable models for information diffusion in social networks. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 259–271. Springer.

[Kivelä et al., 2014] Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *Journal of complex networks*, 2(3):203–271.

[Kossinets et al., 2008] Kossinets, G., Kleinberg, J., and Watts, D. (2008). The structure of information pathways in a social communication network. 14th ACM SIGKDD conference. pp. 435-443.

[Lamprecht et al., 2015] Lamprecht, D., Helic, D., and Strohmaier, M. (2015). Quo vadis? on the effects of wikipedias policies on navigation. *Links*, 80:100.

[Lamprecht et al., 2017] Lamprecht, D., Lerman, K., Helic, D., and Strohmaier, M. (2017). How the structure of wikipedia articles influences user navigation. *New Review of Hypermedia and Multimedia*, 23(1):29–50.

[Landon et al., 2012] Landon, B., Keating, N., Barnett, M., Onnela, J., Paul, S., O'Malley, A., Keegan, T., and Christakis, N. (2012). Variation in patient-sharing networks of physicians across the united states. *JAMA*, 308(3):265–273.

[Lee et al., 2011] Lee, B. Y., McGlone, S. M., Song, Y., Avery, T. R., Eubank, S., Chang, C.-C., Bailey, R. R., Wagener, D. K., Burke, D. S., Platt, R., et al. (2011). Social network analysis of patient sharing among hospitals in orange county, california. *American journal of public health*, 101(4):707–713.

[Leibon et al., 2018] Leibon, G., Livermore, M., Harder, R., Riddell, A., and Rockmore, D. (2018). Bending the law: geometric tools for quantifying influence in the multinetwork of legal opinions. *Artificial Intelligence and Law*, 26(2):145–167.

[Leibon and Rockmore, 2013] Leibon, G. and Rockmore, D. N. (2013). Orienteering in knowledge spaces: The hyperbolic geometry of Wikipedia mathematics. *PloS One*, 8(7):e67508.

[Li et al., 2017] Li, L., Jing, H., Tong, H., Yang, J., He, Q., and Chen, B.-C. (2017). Nemo: Next career move prediction with contextual embedding. In *Proceedings of the 26th International Conference on World Wide Web*, pages 505–513.

[Liao et al., 2013] Liao, Z.-X., Pan, Y.-C., Peng, W.-C., and Lei, P.-R. (2013). On mining mobile apps usage behavior for predicting apps usage in smartphones. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 609–618. ACM.

[Liben-Nowell and Kleinberg, 2007] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

[Lomi et al., 2014] Lomi, A., Mascia, D., Vu, D. Q., Pallotti, F., Conaldi, G., and Iwashyna, T. J. (2014). Quality of care and interhospital collaboration: A study of patient transfers in Italy. *Medical Care*, 52(5):407.

[Lorrain and White, 1971] Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1):49–80.

[Lundh, 2019] Lundh, F. (Accessed: Jan, 2019). Tkinter, python interface to tcl/tk. `https://docs.python.org/2/library/tkinter.html`.

[Lv et al., 2012] Lv, Y., Lymberopoulos, D., and Wu, Q. (2012). An exploration of ranking heuristics in mobile local search. In *Proc. of the SIGIR*, pages 295–304. ACM.

[Lymberopoulos et al., 2011] Lymberopoulos, D., Zhao, P., Konig, C., Berberich, K., and Liu, J. (2011). Location-aware click prediction in mobile local search. In *Proc. of the CIKM*, pages 413–422. ACM.

[Mandl et al., 2014] Mandl, K. D., Olson, K. L., Mines, D., Liu, C., and Tian, F. (2014). Provider collaboration: cohesion, constellations, and shared patients. *Journal of general internal medicine*, 29(11):1499–1505.

[Martínez et al., 2017] Martínez, V., Berzal, F., and Cubero, J.-C. (2017). A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69.

[Meyer, 2017] Meyer, D. (Accessed: Nov. 2017). Classical seasonal decomposition by moving averages. `http://stat.ethz.ch/R-manual/R-devel/library/stats/html/decompose.html`.

[Milo et al., 2002] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827.

[Mitzenmache, 2004] Mitzenmache, M. (2004). A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251.

[Muthukadan, 2018] Muthukadan, B. (2018). Selenium with python.

[Myers and Leskovec, 2010] Myers, S. and Leskovec, J. (2010). On the convexity of latent social network inference. In *Advances in Neural Information Processing Systems*, pages 1741–1749.

[Nakagawa and Shaw, 2004] Nakagawa, Y. and Shaw, R. (2004). Social capital: A missing link to disaster recovery. *International Journal of Mass Emergencies and Disasters*, 22(1):5–34.

[NetworkX-Developers, 2017] NetworkX-Developers (Accessed: Nov. 2017). Networkx. `http://networkx.github.io`.

[Newman, 2003] Newman, M. (2003). The structure and function of complex networks. *SIAM Review*, 45:167–256.

[Newman, 2005] Newman, M. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351.

[Odor et al., 2018] Odor, G., Bugliarello, E., and West, R. (2018). Poster: How did wikipedia become navigable? In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee.

[O'Malley, 2013] O'Malley, A. (2013). The analysis of social network data: An exciting frontier for statisticians. *Statistics in Medicine*, 32(4):539–555.

[O'Malley and Marsden, 2008] O'Malley, A. and Marsden, P. (2008). The analysis of social networks. *Health Services & Outcomes Research Methodology*, 8:222–269.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[Pasricha and McAuley, 2018] Pasricha, R. and McAuley, J. (2018). Translation-based factorization machines for sequential recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 63–71. ACM.

[Python-Software-Foundation, 2019] Python-Software-Foundation (Accessed: Jan, 2019). Urllib, open arbitrary resources by url. `https://docs.python.org/2/library/urllib.html`.

[Raj et al., 2012] Raj, A., Kuceyeski, A., and Weiner, M. (2012). A network diffusion model of disease progression in dementia. *Neuron*, 73(6):1204–1215.

[Ranshous et al., 2015] Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., and Samatova, N. F. (2015). Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223–247.

[Reinhold, 2006] Reinhold, S. (2006). Wikitrails: Augmenting wiki structure for collaborative, interdisciplinary learning. In *Proceedings of the 2006 international symposium on Wikis*, pages 47–58. ACM.

[Rendle, 2010] Rendle, S. (2010). Factorization machines. In *2010 IEEE ICDM*, pages 995–1000. IEEE.

[Rendle et al., 2009] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of The Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press.

[Rendle et al., 2010] Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th World Wide Web Conference*, pages 811–820. ACM.

[Richardson, 2018] Richardson, L. (2018). Beautiful soup documentation. `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`.

[Rombach et al., 2014] Rombach, M., Porter, M., Fowler, J., and Mucha, P. (2014). Core-periphery structure in networks. *SIAM Journal on Applied Mathematics*, 74(1):167–190.

[Sáez and Hogan, 2018] Sáez, T. and Hogan, A. (2018). Automatically generating wikipedia info-boxes from wikidata. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1823–1830. International World Wide Web Conferences Steering Committee.

[Saito et al., 2008] Saito, K., Nakano, R., and Kimura, M. (2008). Prediction of information diffusion probabilities for independent cascade model. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 67–75. Springer.

[Savage et al., 2014] Savage, D., Zhang, X., Yu, X., Chou, P., and Wang, Q. (2014). Anomaly detection in online social networks. *Social Networks*, 39:62–70.

[Schaffert, 2006] Schaffert, S. (2006). Ikewiki: A semantic wiki for collaborative knowledge management. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on*, pages 388–396. IEEE.

[Scikit-community, 2017] Scikit-community (Accessed: Nov. 2017). Scikit-learn: Machine learning in python. `http://scikit-learn.org/stable/`.

[Sen et al., 2017] Sen, S., Swoap, A. B., Li, Q., Boatman, B., Dippenaar, I., Gold, R., Ngo, M., Pujol, S., Jackson, B., and Hecht, B. (2017). Cartograph: Unlocking thematic cartography through semantic enhancement. In *22nd International Conference on Intelligent User Interfaces, IUI 2017*, pages 179–190. Association for Computing Machinery.

[Serrat, 2017] Serrat, O. (2017). Social network analysis. In *Knowledge solutions*, pages 39–43. Springer.

[Shea et al., 1999] Shea, D., Stuart, B., Vasey, J., and Nag, S. (1999). Medicare physician referral patterns. *Health Services Research*, 34(1 Pt 2):331.

[Shin et al., 2012] Shin, C., Hong, J.-H., and Dey, A. K. (2012). Understanding and prediction of mobile application usage for smart phones. In *Ubicomp*, pages 173–182. ACM.

[Strogatz, 2001] Strogatz, S. (2001). Exploring complex networks. *Nature*, 410:268–276.

[Takahashi et al., 2011] Takahashi, T., Tomioka, R., and Yamanishi, K. (2011). Discovering emerging topics in social streams via link anomaly detection. In *2011 IEEE ICDM*, pages 1230–1235. IEEE.

[Teevan et al., 2011] Teevan, J., Karlson, A., Amini, S., Brush, A., and Krumm, J. (2011). Understanding the importance of location, time, and people in mobile local search behavior. In *Proc. of the MobileHCI*, pages 77–80. ACM.

[Theil, 1992] Theil, H. (1992). A rank-invariant method of linear and polynomial regression analysis. In *Henri Theils Contributions to Economics and Econometrics*, pages 345–381. Springer, Germany.

[Thompson and Hanley, 2018] Thompson, N. and Hanley, D. (2018). Science is shaped by wikipedia: Evidence from a randomized control trial.

[Uddin, 2016] Uddin, S. (2016). Exploring the impact of different multi-level measures of physician communities in patient-centric care networks on healthcare outcomes: A multi-level regression approach. *Scientific Reports*, 6:20222.

[Uddin et al., 2013] Uddin, S., Hamra, J., and Hossain, L. (2013). Mapping and modeling of physician collaboration network. *Statistics in medicine*, 32(20):3539–3551.

[Valizadegan et al., 2009] Valizadegan, H., Jin, R., Zhang, R., and Mao, J. (2009). Learning to rank by optimizing ndcg measure. In *Advances in neural information processing systems*, pages 1883–1891.

[Wagner et al., 2014a] Wagner, D. T., Rice, A., and Beresford, A. R. (2014a). Device analyzer: Large-scale mobile data collection. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):53–56.

[Wagner et al., 2014b] Wagner, D. T., Rice, A., and Beresford, A. R. (2014b). Device analyzer: Understanding smartphone usage. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 195–208. Springer.

[Watts, 1999] Watts, D. (1999). Network dynamics and the small world phenomenon. *American Journal of Sociology*, 105(2):493–527.

[Watts and Strogatz, 1998] Watts, D. and Strogatz, S. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442.

[Wikipedia,] Wikipedia. Gini coefficient. Technical report. `https://en.wikipedia.org/wiki/Gini_coefficient`. Accessed Oct 1, 2016.

[Wikipedia, 2016] Wikipedia (2016). Research:wikipedia navigation vectors. https://meta.wikimedia.org/wiki/Research:Wikipedia_Navigation_Vectors.

[Wikipedia, 2019] Wikipedia (2019). Path (graph theory). Accessed: Jan. 2019.

[Wikipedia, 2018] Wikipedia (Accessed: May. 2018). Big o notation - wikipedia. https://en.wikipedia.org/wiki/Big_O_notation.

[Xu et al., 2013] Xu, Y. et al. (2013). Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns. In *ISWC*, pages 69–76. ACM.

[Yang and Counts, 2010] Yang, J. and Counts, S. (2010). Predicting the speed, scale, and range of information diffusion in twitter. *Icwsm*, 10(2010):355–358.

[Yang and Leskovec, 2014] Yang, J. and Leskovec, J. (2014). Overlapping communities explain core-periphery organization of networks. *Proceedings of the IEEE*, 102(12):1892–1902.

[Yelp,] Yelp. Yelp data challenge dataset. https://www.yelp.com/dataset_challenge/dataset/. Accessed Dec.2, 2015.

[Yen et al., 2005] Yen, L., Vanvyve, D., Wouters, F., Fouss, F., Verleysen, M., and Saerens, M. (2005). clustering using a random walk based distance measure. In *ESANN*, pages 317–324.

[Yu et al., 2007] Yu, K., Chu, W., Yu, S., Tresp, V., and Xu, Z. (2007). Stochastic relational models for discriminative link prediction. In *Advances in neural information processing systems*, pages 1553–1560.

[Zlatić et al., 2009] Zlatić, V., Ghoshal, G., and Caldarelli, G. (2009). Hypergraph topological quantities for tagged social networks. *Physical Review E*, 80(3):036118.