To appear in Social Interaction in Virtual Worlds, eds. Lakkaraju, Sukthankar, and Wigand, Cambridge University Press, 2017

1

Using Massively Multiplayer Online Game Data to Analyze the Dynamics of Social Interactions

Alireza Hajibagheri, Gita Sukthankar[a]
Kiran Lakkaraju [b]
Hamidreza Alvari [c]
Rolf T. Wigand, Nitin Agarwal [d]

Abstract

Human societies are inherently complex and highly dynamic, resulting in rapidly changing social networks, containing multiple types of dyadic interactions. Analyzing these time-varying multiplex networks with approaches developed for static, single layer networks often produces poor results. To address this issue, our approach is to explicitly learn the dynamics of these complex networks. Our research focuses on two problems: 1) analyzing the effects of aggression and cooperation on the network structure and 2) modeling link formation patterns across network layers.

To study these problems, we created a rich dataset extracted from observing social interactions in the massively multiplayer online game Travian. Most online social media platforms are optimized to support a limited range of social interactions, primarily focusing on communication and information sharing. In contrast, relations in massively-multiplayer online games (MMOGs) are often formed during the course of gameplay and evolve as the game progresses. To analyze the players' behavior, we constructed multiplex networks with link types for attack, communication, and trading. In this chapter, we illustrate the versatility of the Travian dataset with case studies of how to analyze different aspects of social network dynamics.

Keywords: Multiplex Networks; Link Prediction; Community Detection; Massively Multiplayer Online Games (MMOGs)

[a] Affiliated with Department of Computer Science, University of Central Florida, Orlando, FL
{alireza,gitars}@eecs.ucf.edu
[b] Affiliated with Sandia National Labs, Albuquerque, NM
klakkara@sandia.gov
[c] Affiliated with Arizona State University, Tempe, AZ
halvari@asu.edu
[d] Affiliated with University of Arkansas, Little Rock, AR
{rtwigand,nxagarwal}@ualr.edu

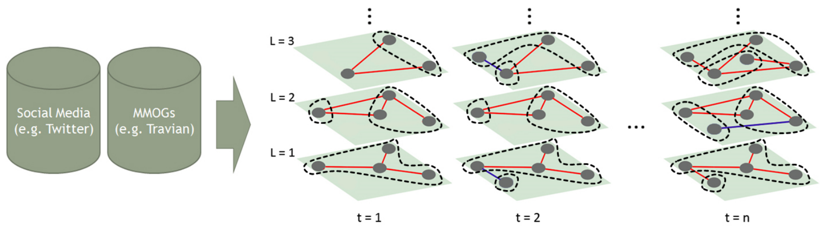


Figure 1.1 A dynamic multiplex network with changing community structure. Each person is associated with a node in a network. The network structure is *dynamic* and changes over time as new users join and social connections are formed. Adding *multiple layers* to the network allows a variety of interactions to be represented within the same network. Users self-organize into *communities* based on shared interests that also change over time.

1.1 Introduction

The natural flux of people's changing social ties and interests generates a dynamic social network. This network can be observed by capturing daily or weekly snapshots of user activities in massively multiplayer online games (MMOGs), allowing these environments to serve as "laboratories" for studying large-scale human behaviors. It is informative to visualize this data as a set of graphs for each time period, where vertices correspond to users and edges represent interactions. Multiple types of dyadic associations can be represented by encoding the data as a multiplex network, where the links at each layer represent a different type of interaction between the same set of nodes. Often these network layers coevolve, due to interdependencies between the social processes represented by different layers. The main goal of our research is to study large-scale human behaviors in coevolving multiplex networks (Figure 1.1). Due to the lack of good standardized datasets, there has been relatively little research on dynamic multiplex networks, as compared to static single layer ones. Our research has been conducted using a dynamic multiplex network dataset collected from the Travian massively multiplayer online game.

Massively multiplayer online games (MMOGs) are highly graphical 2- or 3-D videogames played online, allowing individuals to interact not only with the gaming software (the designed environment of the game and the computer-controlled characters within it) but with other players as well. These virtual environments are persistent social and material worlds where players are usually free to do as they desire. They are notorious for their peculiar combination of designed "escapist fantasy" yet emergent "social realism" (Kolbert, 2001): in a setting of wizards and elves, princes and knights, people save for homes, create basket indices of the trading market, build relationships of status and solidarity, and worry about crime. Given their increasing domination of the entertainment industry, widespread and growing popularity with people of all age groups, ethnicities, and economic classes, and purported addictive quality for those who plug in, MMOGs are quickly becoming a key form of entertainment and a major mechanism of socialization for young and old alike; they are ripe for cultural/cognitive analysis of the social and material practices attending them.

Most online social media platforms are optimized to support a limited range of social interactions, primarily focusing on communication and information sharing. In contrast, relations in massively-multiplayer online games are often formed during the course of gameplay and evolve as the game progresses. Even though these relationships are conducted in a virtual world, they are cognitively comparable to real-world friendships or co-worker relationships (Yee, 2006). The amount and richness of social intercourse makes it possible to observe a broader gamut of human experiences within MMOGs such as World of Warcraft (Thurau and Bauckhage, 2010), Sony EverQuest II (Roy et al., 2013; Keegan et al., 2010), and Travian (Korsgaard et al., 2010; Wigand et al., 2012) than can be done with other data sources. They have been particularly valuable for studying groups, teams, and organizations, since banding together yields economic and combat advantages in most games. Geographically-separated players must work together to achieve shared goals using a similar combination of email, chat, and videoconferencing as remote employees, hence game guilds can be viewed as analogous to virtual workplace organizations (Korsgaard et al., 2010; Wigand, 2017). By observing the changing social interactions within the Travian MMOG, we have been able to model the evolution of the social network and its constituent communities while comparing the model predictions against ground truth information collected from player logs. Moreover it is possible to study the concurrence of different types of social interactions within multiplex networks. This chapter presents an overview of our research on link prediction and community detection within the Travian MMOG.

1.2 Travian MMOG

Travian is a popular browser-based real-time strategy game with more than 5 million players. Games can be played in over 40 different languages on more than 300 game servers worldwide. Playing with up to 20,000 users on one server with scarce resources, actors soon find themselves in a social dilemma (Dawes, 1980), which is typical for organizations, project teams and economies where parties need to both coordinate and compete with one another. Participants start the game as chieftains of their own villages and can choose to be a member of one of three tribes (Gaul, Roman, or Teuton). Each of these three tribes has its own advantages and disadvantages. For instance, Teutons produce the cheapest military units and are the best raiders, whereas Gauls are the best at living in peace and have fast units and merchants. Players seek to improve their production capacity and construct military units in order to expand their territory through a combination of colonization and conquest. Each game cycle lasts a fixed period (a few months) during which time the players vie to create the first civilization to complete construction on one of the Wonders of the World. In the race to dominate, actors form alliances of up to 60 members under a leader or a leadership team. Alliances are equipped with a shared forum, a chat room and an in-game messaging system. Similar to the real world, teamwork and negotiation skills play a crucial role in game success.

Conflicts in Travian can be divided into two categories: attacks and raids. The goal of an attack is to destroy its target, whereas raids are meant to gather bounty and are much less vicious. The armies will do battle until at least one side is reduced in strength by 50%, and therefore the loss on both sides is usually smaller. A trade is an exchange of different resources (gold, wood, clay, wheat) necessary to upgrade a village's buildings. In Travian, villages may trade their resources with other villages if both villages have a marketplace. Travian has an in-game messaging system (IGM) for player communication. IGMs can also include broadcast messages, i.e. messages sent to all players by the game moderators. Note these messages were not included in our experiments as their volume could introduce bias in the results. For our analysis, we used data collected from a server in Germany specifically designed for research purposes. The data set contains a variety of tables including logs and reports from different actions of users. To study the dynamics of social processes within the game, we structured the multiplex network into raid, trade, and communication layers.

1.3 Network Analysis

Network structure analysis has become an increasingly important aspect of understanding user behavior on social media platforms. This methodology places relations and links among entities, or people, at the center of investigation. In the last decade, much research has been performed on characterizing the dynamics of complex systems and extracting non-trivial properties using massive network data from social, biological, and technological sources. Example applications include: predicting future links among the actors of a network (Liben-Nowell and Kleinberg, 2007; Bringmann et al., 2010), detecting and studying the structure of communities (Alvari et al., 2016) and mining common user behavior patterns (Benevenuto et al., 2009; Cook et al., 2010).

Many studies on online social networks, the WWW, and biological networks have focused on the macroscopic properties of static networks (Faloutsos et al., 1999; Albert and Barabási, 2002; Broder et al., 2000; Strogatz, 2001). However, social networks are not static. They are dynamic structures that evolve over time either by the addition of new vertices or by new connecting edges. Thus, modeling network dynamics is important and the focus of a number of research efforts (Backstrom et al., 2006; Barabási and Albert, 1999; Leskovec et al., 2008). Also in the real world, networks are often multiplex, containing multiple types of relationships; in some cases, aggregating different interaction types loses information about the structure and function of the original system (Kurant and Thiran, 2006; Buldyrev et al., 2010). For instance, the same set of individuals in a social system can be connected through friendship, collaboration, communication and co-location relationships; in massively multiplayer online games, players have a variety of interactions such as trading, messaging and attacking. In these systems, each type of relationship may have a different semantic meaning, relevance, importance, and cost, so that treating all the links as being equivalent discards key information. Multiplex networks serve as a better description of these systems; each node appears in a set of different layers, and each layer describes all the edges of a given type.

Recently, a considerable amount of effort has been devoted to the characterization and modeling of multiplex networks, with the aim of creating a consistent mathematical framework to study, understand and reproduce the structure of these systems. For instance, it is feasible to model multiplex networks using a statistical mechanics approach (Bianconi, 2013). Another alternative is to simply extend classical network metrics to handle multiple layers (Sole-Ribalta et al., 2013; De Domenico et al., 2013a) and to model the growth of systems of this kind (Nicosia et al., 2013). An active research area is characterizing the dynamics and the emergent properties of multilayer systems, especially with respect to contagious properties (Saumell-Mendiola et al., 2012), information propagation (Buono et al., 2014; Min and Goh, 2013), cooperation (Gómez-Gardenes et al., 2012), diffusion processes (Gomez et al., 2013) and random walks (De Domenico et al., 2013b). The subsequent sections of the chapter chronicle a set of studies that we completed using multiplex networks extracted from the Travian MMOG, and the algorithms that we developed for modeling the evolution of communities and the dynamics of link formation.

1.4 Analyzing the Effects of Aggression on Network Structure

MMOGs have been a fertile testing ground for many types of human studies, enabling scientists to overcome key difficulties in studying social dynamics by providing an experimental platform for collecting high resolution data over longer time period (Thurau and Bauckhage, 2010; Korsgaard et al., 2010; Wigand et al., 2012; Roy et al., 2013). One research question of interest is how conflict shapes the underlying social network; in MMOGs, conflict and cooperation are inextricably linked since many attacks are launched by coalitions of players to gain resources, control territory, or subjugate enemies. It is easier to study aggression in virtual worlds since it is both more common and simpler to quantify.

In real-life there are myriad potential motivations for choosing to fight. Humphreys and Weinstein (2008) categorized key determinants of participation in conflicts as being long-term grievances (i.e. economic or political disenfranchisement), selective incentives (money or safety), and community cohesion. Community cohesion predicts that a person is more likely to join the conflict if they are members of a tightly-knit community and their friends have already joined. This factor is the most relevant to fighting within MMOGs. Not only are there conflicts between guilds and alliances, but pick-up groups may spontaneously form to tackle larger challenges such as boss fights (Bennerstedt et al., 2012).

In Travian, attacking (raiding) is one of the easiest pathways for gaining the necessary resources for growing one's civilization, and players need to rush to grow their civilizations within a short period of time.

Table 1.1 *Travian attack, message, and trade network statistics.*

Parameter/Network	**Attack**	**Message**	**Trade**
# of Vertices	4418	3092	2649
Frequency	633105	451669	271039
Diameter	17	9	10
Avg. Path Length	5.312	3.471	2.849
Avg. Degree	7.998	14.591	32.828
Avg. Clustering Coefficient	0.065	0.319	0.154

Here we study 1) how the structure of the attack layer differs from the communication and trade layers and 2) how communication, trade, and geographic connections affect the likelihood of two players engaging in hostilities. For this study, we used data from one Travian game cycle played on a high speed server in an expedited game (a period of 144 days). Our analysis was conducted on a 30 day period in the middle of the game cycle. This period has fewer transient bursts of activity and a more stable network than the early period (which has many less committed players who drop out) and the late period where the focus is on the Wonder of the World construction.

Table 1.1 shows the statistics for attack, trade, and message networks during the time period selected for this analysis. In these networks, each node represents an individual player, and directed edges represent attacks, trades, and messages between players. The attack graph in Travian has a higher diameter, lower average degree, and lower clustering coefficient than either the message or trade graphs.

The degree distributions of attack, messages, and trades conform to a power law distribution (Figure 1.2). Clauset et al. (2009) proposed a robust estimating technique to estimate the parameters of a power law; to verify the distributions, we used this method which employs a maximum likelihood estimator. This model calculates the goodness-of-fit between the data and the power law. If the resulting value is greater than 0.1 the power law is a plausible hypothesis for the data, otherwise it is rejected.

Assortativity is a preference for a network's nodes to attach to others that are similar in some way. Though the specific measure of similarity may vary, network theorists often examine assortativity in terms of a node's degree. Correlations between nodes of similar degree are found in the mixing patterns of many observable networks. For instance, in social networks, highly connected nodes tend to be connected with other high degree nodes. This tendency is referred to as assortative mixing, or

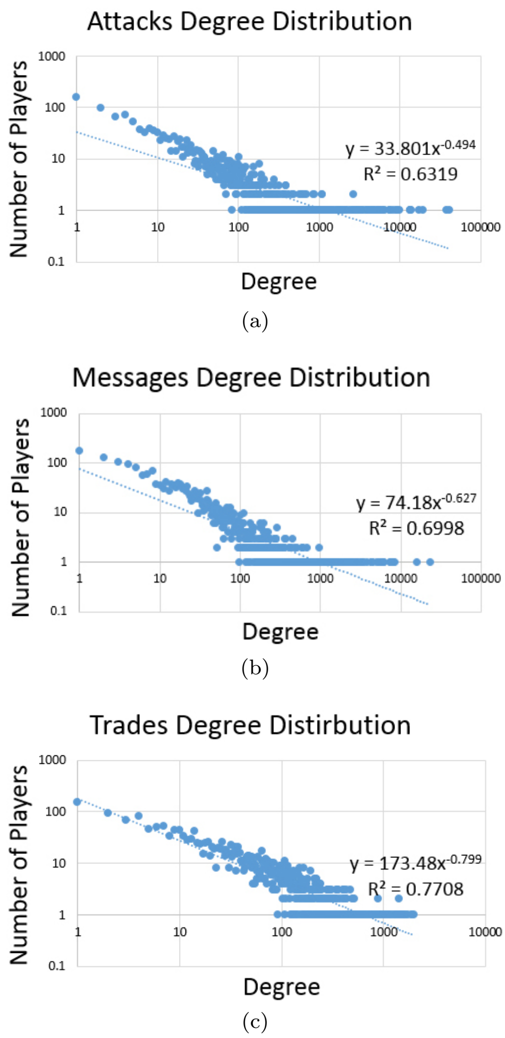
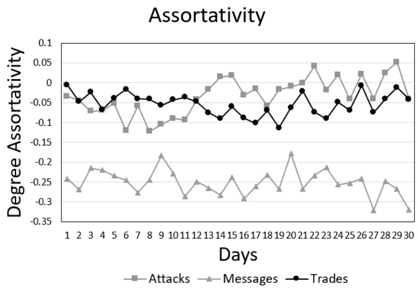


Figure 1.2 Degree distribution (log-log scale) for (a) attack, (b) message, and (c) trade networks

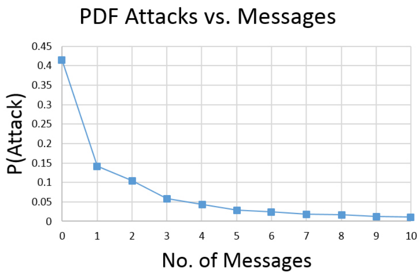
assortativity. On the other hand, technological and biological networks typically show disassortative mixing, or dissortativity, as high degree nodes tend to attach to low degree nodes (Newman, 2002).

For Travian, as shown in Figure 1.3, while the message network displays disassortative mixing, attack and trade networks tend to show a non-assortative mixing. This suggests that players who send more mes-



(a)

Figure 1.3 Travian node degree assortativity

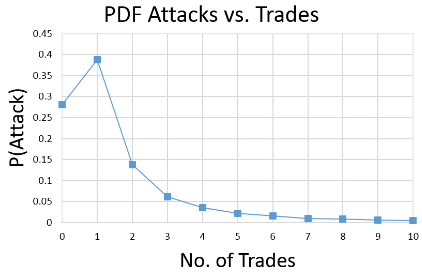


(a)

Figure 1.4 Probability of attacks occurring between a pair of users vs. the number of messages they have exchanged (P(Attack and Message=x))

sages are in contact with others who rarely send messages; communication in Travian often flows from alliance leaders outward to the other alliance members, reflective of a spoke-hub communication structure. In contrast, the degree of the members appears to be an unimportant consideration in dictating connectivity in the attack and trade networks. Non-assortative networks may arise either because the networks possess a balanced number of assortative and disassortative links or because a greater number of links in one direction is counterbalanced by a greater weight in the other (Piraveenan et al., 2012).

Attacks in Travian are generally inversely proportional to other types of activity. In Travian, in 41% of cases, players do not attack other players with whom they have been in contact at least once (Figure 1.4). A large number of players do not attack players with whom they have traded resources. As shown in Figure 1.5, 28% of the attacks in Travian



(a)

Figure 1.5 Probability of attacks occurring between a pair of users vs. the number of trades they have made (P(Attack and Message=x))

occurred between two players without any trade history. Trading with other players indicates that they have desirable resources, making them worth attacking, and after only one trade, the players are unlikely to have established the sense of trust that may deter an attack. We believe that in some cases players who have never traded together or exchanged messages are geographically separated; hence they are less likely to attack each other because they are unaware of each other's existence. To test this hypothesis, we analyzed the probability of attack based on the distance between player territories in Travian (Figure 1.6). To estimate distance, we calculated the territory centroids by averaging the latitudes and longitudes of the villages. Then, standard Euclidean distance was used to measure the distance between each pair of players in the attack network. Our analysis shows that attacks between immediate neighbors are frequent. Attacks with close (but not immediate) neighbors are common, followed by a decay in attack activity with distance. Attacks are generally rare between alliance and guild members, indicating a strong level of trust in those relationships. In Travian, 4% of the attack edges are between two players within the same alliance.

Similar to real-life, social structures play a significant role in the likelihood of inter-player conflict. In summary, our analysis reveals the following:

1 The attack network has a higher diameter, lower average degree, and lower clustering coefficient than either the message or the trade networks.
2 All networks have similar power law degree distributions, but different degree assortativity. The Travian attack network shows non-assortative mixing.

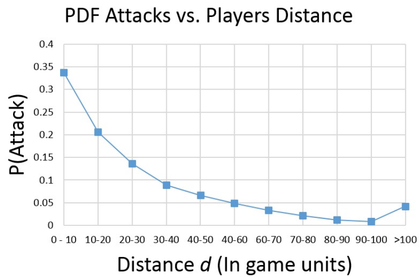


Figure 1.6 Probability of attacks based on players' distance from each other

3 The general trend is that attacks are inversely proportional to message frequency, trade frequency, and distance, with some specific exceptions. Players rarely attack fellow alliance or guild members.

1.5 Modeling the Evolution of Alliance Structures

In addition to facilitating our understanding of aggression, Travian is also an interesting testbed for studying cooperation, since forming a strong alliance is an important stepping stone towards achieving the final objective of creating the Wonder of the World. Here we analyze how alliances change and evolve during the course of the Travian game cycle. Although lacking in formal alliances, most real-world social networks are inherently dynamic and are composed of communities that are constantly changing in membership. As a result, recent years have witnessed increased attention toward the challenging problem of detecting evolving communities. As the network changes, user communities evolve and can grow, shrink, or disappear. Intuitively we expect more edges inside the community compared to its outside, i.e. intra-connections tend to be more common than inter-connections. Community detection can help us understand the hidden social structure of the user populations, but the dynamic aspect of networks can pose problems for standard algorithms.

Formally, given snapshots $\mathbf{T} = \{T^t \mid \forall t, t = 1, ..., M\}$ of a dynamic network and their corresponding underlying graphs $\mathbf{G}^t = (V^t, E^t)$, with $n^t = |V^t|$ vertices and $m^t = |E^t|$ edges, where t=1,...,M, we aim to detect community structure $\mathbf{C} = \{C^t \mid \forall t, t = 1, ..., M\}$ of the network. The process of community detection is treated as an iterative game performed in a dynamic multi-agent environment in which each node of

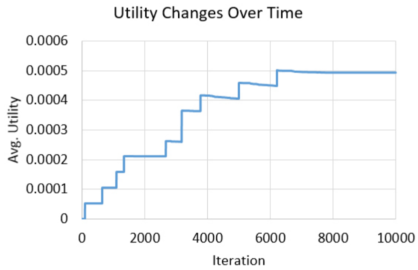


Figure 1.7 Change in average utility summed over all nodes vs. iteration for the Travian-Trades dataset (one snapshot with 964 nodes). The algorithm converges after 6680 iterations which requires 2.8 seconds to complete.

the underlying graph is a selfish agent who decides to maximize its total utility u_i. For every snapshot of the network, a set of agents, one representing each node in the graph, is created to play the community formation game. The community structure is initialized either with a set of singleton communities or with communities passed from previous snapshots. During game play, an agent is randomly selected (without replacement) from the pool; it selects an action (join, leave, switch, or no op) by calculating the action that yields the highest utility. After the agent plays, the community structure is updated. The game is played until the number of agents changing communities between permutations falls below the threshold, or the maximum iteration is reached. Figure 1.7 shows an example of the convergence in utility vs. iteration. The algorithm maintains a candidate set of multiple community assignments per agent until the last iteration and then selects the assignment with the highest utility function as the final disjoint partition. Our method, D-GT (**D**ynamic **G**ame **T**heoretic community detection, originally introduced in (Alvari et al., 2011)), outperforms several other state of the art methods for detecting changing alliances within the Travian game. We also created a version of the algorithm, **D-GTG** (D-GT with passing Ground Truth) to handle cases where the alliance structure is partially known. For instance, MMOG guilds and alliances often have a leadership council that is openly publicized or easily inferred based on the content of chat messages. D-GTG leverages this information by using a select seed group of ground truth communities with predefined size to initialize the algorithm.

Table 1.2 *Dataset Summary*

Data	**Messages**	**Trades**
Min # of nodes	1,373	964
Max # of nodes	2,100	1,336
Min # of edges	8,511	8,080
Max # of edges	19,242	10,221
# of snapshots	30	30

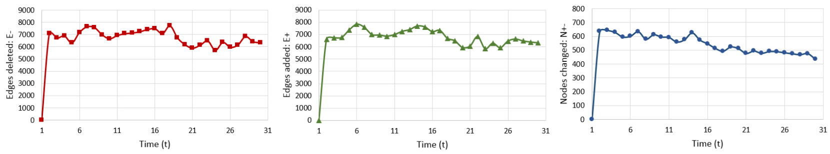


Figure 1.8 The structural changes in the Travian Trades dataset over 30 snapshots.

1.5.1 Evaluation

For this evaluation, we used two layers of Travian multiplex network: Messages and Trades. Travian has an in-game messaging system (IGM) for player communication which was used to create our Messages network. Each player can submit a request to trade a specific resource. If another player finds this request interesting, he/she can accept it and the trade will occur; this data was used to build the Trade network. About 70% of messages are exchanged between users in the same alliance (community) making it more predictive of community structure than the Trades network since only 30% of edges in this network represent trades occurred between players within the same alliance. The structural changes in both Travian datasets are shown in Figures 1.8 and 1.9; statistics are provided in Table 1.2. Modeling the evolution of alliances is harder during periods of significant structural change, when large numbers of edges are being added and deleted.

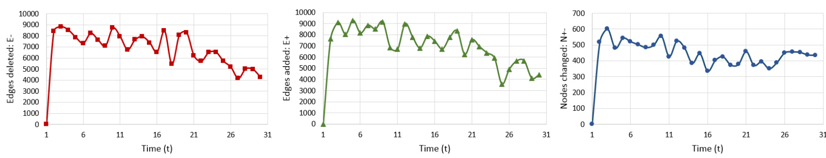


Figure 1.9 The structural changes in the Travian Messages dataset over 30 snapshots.

We compare D-GT with the following community detection baselines:

- **LabelRankT (Xie et al., 2013).** LabelRankT functions according to the generalized LabelRank, in which each node requires only local information during label propagation processing. Several parameters must be set before running the algorithm on the data; we used the best performing values reported in the original paper.
- **iLCD (Cazabet et al., 2010).** iLCD is another well known community detection approach for dynamic social networks which works by first adding edges and then merging the similar ones. It takes the dynamics of the network into account.
- **OSLOM (Lancichinetti et al., 2011).** The Order Statistics Local Optimization Method (OSLOM) is a versatile community detection algorithm that can handle most types of graph properties including edge directions and weights, overlapping communities, hierarchies and community dynamics. It is based on the local optimization of a fitness function expressing the statistical significance of clusters with respect to random fluctuations.
- **InfoMap (Rosvall and Bergstrom, 2008).** InfoMap is a static community detection method that calculates the probability flow of random walks and decomposes the network into modules by compressing a description of the flows. Since this is a static algorithm, we run it separately on each snapshot.
- **Louvain (Blondel et al., 2008).** The Louvain method is a static community detection approach designed to optimize modularity using heuristics. Small communities are found by optimizing modularity locally for all nodes. Then each community is grouped into a single node, and the first step is repeated. We run this algorithm separately on every network snapshot.

Algorithms were evaluated together on a system with 12G of RAM and an Intel CPU 2.53 GHz, and all reported results were averaged over ten repetitions. The best way to measure the performance of a community detection algorithm is to determine how similar the partition delivered by the algorithm is to the desired partition, assuming ground truth information about the community membership exists. Out of several existing measures (Fortunato, 2010), we selected the standard version of normalized mutual information (NMI) (Danon et al., 2005), which is computed as follows:

$$\mathbf{I}_{norm}(\mathbf{X},\mathbf{Y}) = \frac{2I(X,Y)}{H(X)+H(Y)}, \tag{1.1}$$

where $I(X,Y)$ is mutual information between two random variables X and Y (i.e. two community partitions) (MacKay, 2003):

$$\mathbf{I}(\mathbf{X},\mathbf{Y}) = \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)P(y)}, \tag{1.2}$$

Here $P(x)$ indicates the probability that $X = x$ and joint probability $P(x,y)$ equals to $P(X = x, Y = y)$. $H(X)$ and $H(Y)$ are the entropies of X and Y, respectively. NMI lies in the range [0,1], equaling one when two partitions X and Y are exactly identical and zero when they are totally independent.

Figure 1.10 shows the average performance of D-GT vs. OSLOM, LabelRankT, iLCD, InfoMap and Louvain. Unlike many social media datasets, the Travian dataset contains ground truth alliance membership information that can be used to calculate the NMI. D-GT outperforms all other methods ($p < 0.01$) on this metric. Figure 1.11 shows D-GT's performance at predicting the number of alliances, as measured by summed absolute difference between predicted and actual community numbers (lower is better). Note that it is possible to do acceptably well on the NMI metric while still incorrectly estimating the actual number of communities in the dataset. OSLOM also scores well on both metrics (NMI and number of communities). Additionally, it is useful to examine how the number of predicted communities varies between consecutive snapshots. In most cases, the number of communities should remain relatively stable, since the structure of real-world communities rarely changes completely in short period of time. This is definitely true in Travian, where the number of alliances changes relatively slowly. Figure 1.12 shows the number of predicted communities vs. time on the Travian (Trades) dataset; all of the methods make more consistent predictions over time than LabelRankT.

In some scenarios, it is plausible that the community membership of a small number of agents is known in advance, and the community detection procedure should leverage this information. To handle this problem, we developed a variant (D-GTG: D-GT with passing Ground

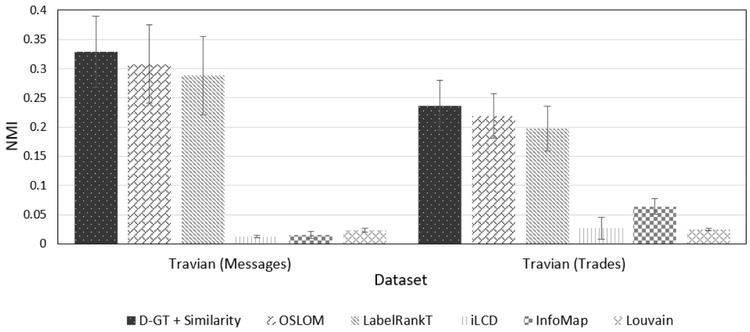


Figure 1.10 Normalized mutual information (NMI) evaluation metric on the two Travian datasets with ground truth community membership information; results are averaged over all snapshots. D-GT, LabelRankT, OSLOM, iLCD, InfoMap and Louvain.

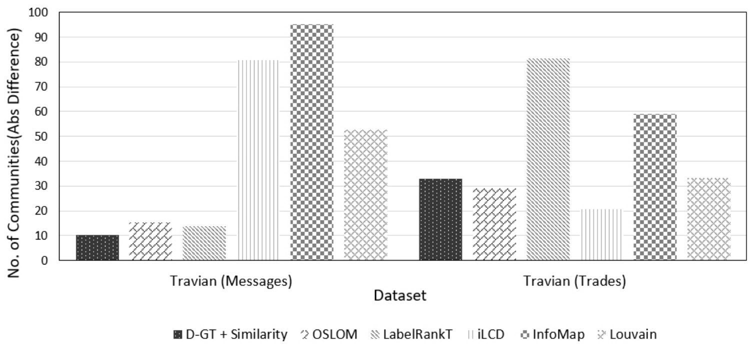


Figure 1.11 Absolute difference between the predicted number of communities and the actual number for the two Travian datasets. D-GT and OSLOM achieve the best performance overall at correctly predicting the number of alliances.

Truth). Fig. 1.13 shows the performance improvements from increasing the size of the seed groups from 0–20% of the total number of agents for the Travian (Messages) dataset, and Fig. 1.14 shows the performance increase for Travian (Trades). Note that extracting community membership information from the network structure of Travian (Trades) is a difficult problem because only 30% of the edges in Travian (Trades) occur between players within the same alliance (community). Also the dataset has a high number of isolated nodes; about 50% of the nodes do not belong to any alliance.

In summary, our results demonstrate that D-GT can accurately track

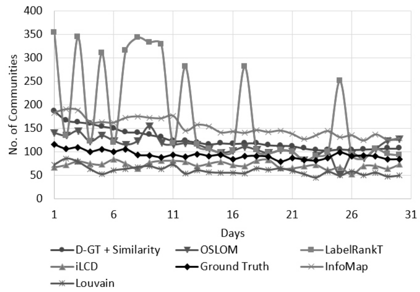


Figure 1.12 Number of predicted communities vs. time for the Travian (Trades) dataset. LabelRankT's predicted number of communities varies drastically between time steps, whereas all other algorithms make more consistent predictions.

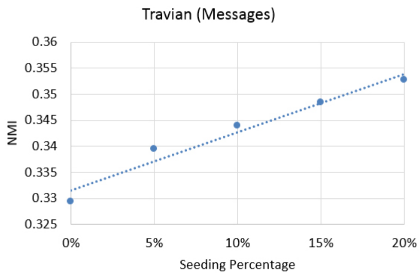


Figure 1.13 D-GTG NMI vs. seed group size on Travian (Messages)

the evolution of alliance structure. For this task, it outperforms other dynamic community detection methods including LabelRankT, iLCD, and OSLOM. In cases where the community membership of a small number

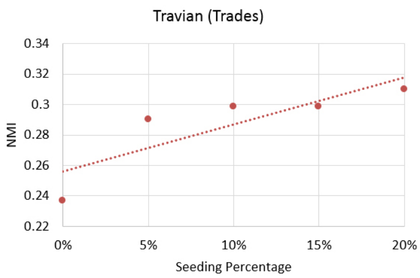


Figure 1.14 D-GTG NMI vs. seed group size on Travian (Trades)

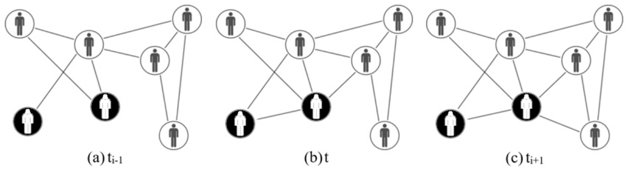


Figure 1.15 Evolution of a network over time. Black nodes have higher *rates* of link formation. This behavior can only be captured by taking temporal information into account; RPM identifies these nodes through the use of time series.

of players (e.g. the guild leadership) is known D-GTG can leverage this information to improve the NMI score.

1.6 Rate Prediction Model for Link Formation

Many social networks are constantly in flux, with new edges and vertices being added or deleted daily. Fully modeling the dynamics that drive the evolution of a social network is a complex problem, due to the large number of individual and dyadic factors associated with link formation. Here we focus on predicting one crucial variable–the *rate* of network change. Not only do different networks change at different rates, but individuals within a network can have disparate tempos of social interaction. This section describes how modeling this aspect of network dynamics can ameliorate performance on link prediction tasks. We introduce a new supervised link prediction framework, RPM (Rate Prediction Model). In addition to network similarity measures, RPM uses the predicted rate of link modifications, modeled using time series data.

1.6.1 Problem Formulation

The problem of link prediction in dynamic networks is defined as: Let graph G be the social network of interest denoted as (V, E), where V is the set of nodes and $E \in V \times V$ is the set of (directed or undirected) interactions. Let G_t be the subgraph of G containing the nodes and edges recorded at time t. In turn, let G_{t+1} be the subgraph of G observed at time $t + 1$. Using network structure up to time t, our goal is then to predict future structure of the network at time $t + 1$.

1.6.2 Background

Link prediction approaches commonly rely on measuring topological similarity between unconnected nodes (Al Hasan and Zaki, 2011; Getoor and Diehl, 2005; Wang et al., 2007). It is a task well suited for supervised binary classification since it is easy to create a labeled dataset of node pairs; however, the datasets tend to be extremely unbalanced with a preponderance of negative examples where links were not formed. Topological metrics are used to score node pairs at time t in order to predict whether a link will occur at a later time $t'(t' > t)$. However, even though these metrics are good indicators of future network connections, they are less accurate at predicting *when* the changes will occur (the exact value of t'). To overcome this limitation, we explicitly learn link formation rates for all nodes in the network; first, a time series is constructed for each node pair from historic data and then a forecasting model is applied to predict future values. The output of the forecasting model is used to augment topological similarity metrics within a supervised link prediction framework. Prior work has demonstrated the general utility of modeling time for link prediction (e.g., (Huang and Lin, 2009; Berlingerio et al., 2009; Potgieter et al., 2009)); our results show that our specific method of rate modeling outperforms the use of other types of time series. Networks formed from different types of social processes (e.g., trades vs. communication) may vary in their dynamics, but our experiments show that RPM outperforms other standard approaches on multiple types of datasets.

1.6.3 Time Series

To construct the time series, the network G observed at time t must be split into several time-sliced snapshots, that is, states of the network at different times in the past. Afterwards, a window of prediction is defined, representing how further in the future we want to make the prediction. Then, consecutive snapshots are grouped in small sets called frames. Frames contain as many snapshots as the length of the window of prediction. These frames compose what is called Framed Time-Sliced Network Structure (S) (Soares and Prudêncio, 2012). Let G_t be the graph representation of a network at time t. Let $[G_1, G_2, ..., G_T]$ be the frame formed by the union of the graphs from time 1 to T. Let n be the number of periods (frames) in the series. And let w be the window of prediction. Formally, S can be defined as:

$$S = \{[G_1, ..., G_w], [G_{w+1}, ..., G_{2w}], ...[G_{(n-1)w+1}, ..., G_{nw}]\}$$

For instance, suppose that we observed a network from day 1 to day 9, and our aim is to predict links that will appear at day 10. In this example, the forecast horizon (window of prediction) is one day. Our aim here is to model how the networks evolve every day in order to predict what will happen in the forecast horizon. Figure 1.15 shows an example of the evolution of network over time.

1.6.4 Network Similarity Metrics

Here, we use a standard set of topological metrics to assign scores to potential links:

1 Common Neighbors (CN) (Newman, 2001) is defined as the number of nodes with direct relationships with both members of the node pair:

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{1.3}$$

where $\Gamma(x)$ is the set of neighbors of node x.

2 Preferential Attachment (PA) (Barabási *et al.*, 2009; Liben-Nowell and Kleinberg, 2003) assumes that the probability that a new link is created is proportional to the node degree $|\Gamma(y)|$. Hence, nodes that currently have a high number of relationships tend to create more links in the future:

$$PA(x, y) = |\Gamma(x)| \times |\Gamma(y)| \tag{1.4}$$

3 Jaccard's Coefficient (JC) (Tan et al., 2005) assumes higher values for pairs of nodes that share a higher proportion of common neighbors relative to total number of neighbors they have:

$$JC(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{1.5}$$

4 Adamic-Adar (AA) (Adamic and Adar, 2003), similar to JC, assigns a higher importance to the common neighbors that have fewer total neighbors. Hence, it measures exclusivity between a common neighbor and the evaluated pair of nodes:

$$AA(x, y) = \sum_{z \in |\Gamma(x) \cap \Gamma(y)|} \frac{1}{log(|\Gamma(z)|)} \tag{1.6}$$

These metrics serve as 1) unsupervised baseline methods for evaluating the performance of RPM and 2) are also included as features used by the supervised classifiers.

1.6.5 Method

RPM treats the link prediction problem as a supervised classification task, where each data point corresponds to a pair of vertices in the social network graph. This is a typical binary classification task that could be addressed with a variety of classifiers; we use the Spark support vector machine (SVM) implementation. All experiments were conducted using the default parameters of the Spark MLlib package: the SVM is defined with a polynomial kernel and a cost parameter of 1. Algorithms were implemented in Python and executed on a machine with Intel(R) Core i7 CPU and 24GB of RAM.

In order to produce a labeled dataset for supervised learning, we require timestamps for each node and edge to track the evolution of the social network over time. We then consider the state of the network for two different time periods t and t' (with $t < t'$). The network information from time t is used to predict new links which will be formed at time t'. One of the most important challenges with the supervised link prediction approach is handling extreme class skewness. The number of possible links is quadratic in the number of vertices in a social network, however the number of actual edges is only a tiny fraction of this number, resulting in large class skewness.

The most commonly used technique for coping with this problem is to balance the training dataset by using a small subset of the negative examples. Rather than sampling the network, we both train and test with the original data distribution and reweight the misclassification penalties. Let $G(V, A)$ be the social network of interest. Let G_t be the subgraph of G containing the nodes and edges recorded at time t. In turn, let $G_{t'}$ be the subgraph of G observed at time t'. In order to generate training examples, we considered all pairs of nodes in G_t. Even though this training paradigm is more computationally demanding it avoids the concern that the choice of sampling strategy is distorting the classifier performance (Lichtenwalter et al., 2010).

Selecting the best feature set is often the most critical part of any machine learning implementation. In this dissertation, we supplement the standard set of features extracted from the graph topology (described in the previous section), with features predicted by a set of time series.

Let $F_t(t = 1, ..., T)$ be a time series with T observations with A_t defined as the observation at time t and F_{t+1} the time series forecast at time $t + 1$. First, we analyze the performance of the following time series forecasting models for generating features:

1 **Simple Mean**: The simple mean is the average of all available data:

$$F_{t+1} = \frac{A_t + A_{t-1} + ... + A_{t-T}}{T}$$

2 **Moving Average**: This method makes a prediction by taking the mean of the n most recent observed values. The moving average forecast at time t can be defined as:

$$F_{t+1} = \frac{A_t + A_{t-1} + ... + A_{t-n}}{n}$$

3 **Weighted Moving Average**: This method is similar to moving average but allows one period to be emphasized over others. The sum of weights must add to 100% or 1.00:

$$F_{t+1} = \sum C_t A_t$$

4 **Exponential Smoothing**: This model is one of the most frequently used time series methods because of its ease of use and minimal data requirements. It only needs three pieces of data to start: last period's forecast (F_t), last period's actual value (A_t) and a value of smoothing coefficient,α, between 0 and 1.0. If no last period forecast is available, we can simply average the last few periods:

$$F_{t+1} = \alpha A_t + (1 - \alpha) F_t$$

We identify which time series prediction model produces the best rate estimate, according to the AUROC performance of its RPM variant. Parameters of weighted moving average and exponential smoothing were tuned to maximize performance on the training dataset. Figure 1.16 shows that the best performing model was Weighted Moving Average with $n = 3$ and parameters C_1, C_2 and C_3 set to 0.2,0.3, and 0.5 respectively.

1.6.6 Results

Table 1.3 gives the network statistics for each of the datasets used in the evaluation. Our evaluation measures receiver operating characteristic (ROC) curves for the different approaches. These curves show achievable true positive rates (TP) with respect to all false positive rates (FP)

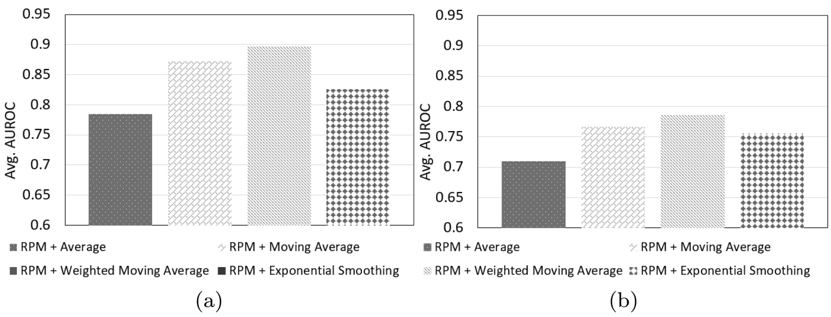


Figure 1.16 Performance of RPM using different forecasting models on (a) Travian Messages and (b) Travian Trades. Weighted Moving Average is the best performer and is used in RPM.

by varying the decision threshold on probability estimations or scores. For all of our experiments, we report area under the ROC curve (AUROC), the scalar measure of the performance over all thresholds. Since link prediction is highly imbalanced, straightforward accuracy measures are well known to be misleading; for example, in a sparse network, the trivial classifier that labels all samples as missing links can have a 99.99% accuracy.

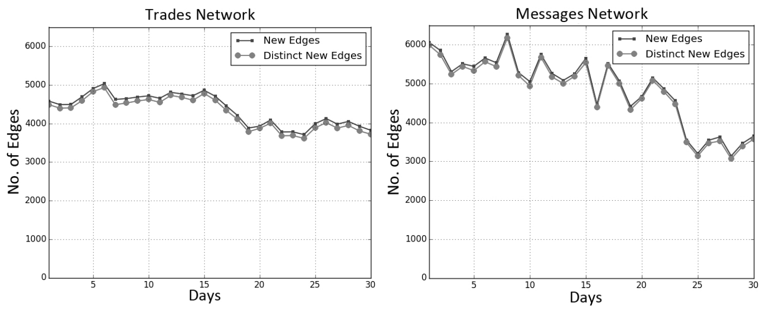


Figure 1.17 Dynamics of the Travian network (trades: left and messages: right). The line with square markers shows the new edges added, and the line with circle markers shows edges that did not exist in the previous snapshot.

In all experiments, the algorithms were evaluated with stratified 10-fold cross-validation. For more reliable results, the cross-validation procedure was executed 10 times for each algorithm and dataset. We benchmark our algorithm against **Supervised-MA** (Soares and Prudêncio,

Table 1.3 *Dataset Summary*

Data	**Travian (Messages)**	**Travian (Trades)**
No. of nodes	2,809	2,466
Link (Class 1)	44,956	87,418
No Link (Class 0)	7,845,525	5,993,738
No. of snapshots	30	30

Table 1.4 *AUROC Performance*

Algorithms / Networks	Travian(Messages)	Travian(Trades)
RPM	**0.8970**	**0.7859**
Supervised-MA	0.8002	0.6143
Supervised	0.7568	0.7603
Common Neighbors	0.4968	0.5002
Jaccard Coefficient	0.6482	0.4703
Preferential Attachment	0.5896	0.5441
Adamic/Adar	0.5233	0.4962

2012). Supervised-MA is a state of the art link prediction method that is similar to our method, in that it is supervised and uses moving average time series forecasting. In contrast to RPM, Supervised-MA creates time series for the unsupervised metrics rather than the link formation rate itself. **Supervised** is a baseline supervised classifier that uses the same unsupervised metrics as features without the time series prediction model. As a point of reference, we also show the unsupervised performance of the individual topological metrics: 1) **Common Neighbors**, 2) **Preferential Attachment**, 3) **Jaccard Coefficient**, and 4) **Adamic-Adar**. Table 1.4 presents results for all methods on Travian (communication and trade layers). Results for our proposed method are shown using bold numbers in the table; in all cases, RPM outperforms the other approaches. Two-tailed, paired t-tests across multiple network snapshots reveal that the RPM is significantly better ($p < 0.01$) on all four datasets when compared to Supervised-MA.

We discover that explicitly including the rate feature (estimated by a time series) is decisively better than the usage of time series to forecast topological metrics. The rate forecast is useful for predicting the source node of future links, hence RPM can focus its search on a smaller set of node pairs. We believe a combination of topological metrics is useful for predicting the destination node, but that relying exclusively on the topological metrics, or their forecasts, is less discriminative.

The performance of RPM relies on three innovations: 1) explicit modeling of link formation rates at a node level, 2) the usage of multiple time series to leverage information from earlier snapshots, 3) training and testing with the full data distribution courtesy of the Spark fast cluster computing system. Rate is an important concept in many generative network models, but its usage has been largely ignored within discriminative classification frameworks. For instance, the stochastic actor-oriented model of network dynamics contains a network rate component that is governed by both the time period and the actors (Snijders et al., 2010). RPM does not attempt to create a general model of how the rate is affected by the properties of the actor (node), but instead predicts the link formation rate of each node with a time series. By accurately identifying the most active individuals in the social network, RPM achieves statistically significant improvements over related link prediction methods. Our experiments were performed on networks created by a variety of social processes, including communication and trading; they show that the rate of link generation varies with the type of network.

Link Prediction in Coevolving Multiplex Networks

As social media platforms offer customers more interaction options, such as *friending*, *following*, and *recommending*, analyzing the rich tapestry of interdependent user interactions becomes increasingly complicated. Although standard social network analysis techniques (Scott, 2012) offer useful insights about these communities, there is relatively little theory from the social sciences on how to integrate information from multiple types of online interactions. Rather than organizing this data into social networks separately chronicling the history of different forms of user interaction, dynamic multiplex networks (Kivela et al., 2014) offer a richer formalism for modeling the social fabric of online societies. This section introduces a comprehensive framework, MLP (Multiplex Link Prediction), in which link existence likelihoods for the target layer are learned from the other network layers. These likelihoods are used to reweight the output of a single layer link prediction method that uses rank aggregation to combine a set of topological metrics.

A multiplex network is a multilayer network that shares the same set of vertices across all layers. This network can be modeled as a graph $G =< V, E >$ where V is the set of vertices and E is the set of edges present in the graph. The dynamic graph $G = \{G_0, G_1, ..., G_t\}$ represents the state of the network at different times. The network is then defined as: $G_t =< V, E_t^1, ..., E_t^M >$ with $E_t^\alpha \subseteq V \times V$, $\forall \alpha \in \{1, ..., M\}$, where each set E_t^α corresponds to the edge set of a distinct layer at time t. Thus a dynamic multiplex network is well suited for representing diverse user activities over a period of time. Here, we address the problem of predicting future user interactions from the history of past connections. Assuming the data is represented as a graph, our goal is then to predict the structure of graph G_t with α as the target layer, using information from previous snapshots as well as other layers of the network.

MLP is a hybrid architecture that utilizes multiple components to address different aspects of the link prediction task. We seek to extract information from all layers of the network for the purpose of link prediction within a specific layer known as the target layer. To do so, we create a weighted version of the original target layer where interactions and connections that exist in other layers receive higher weights. After reweighting the layer, we employ the collection of node similarity metrics on the weighted network. To express the temporal dynamics of the network, we use a decay model on the time series of similarity metrics to predict future values. Finally, the Borda rank aggregation method is employed to combine the ranked lists of node pairs into a single list that predicts links for the next snapshot of the target network layer. Each component of the model is explained in more detail in the following sections.

1.6.7 Multiplex Likelihood Assignment and Edge Weighting

This component leverages information about cross-layer link co-occurrences. During the coevolution process, links may be engendered due to activity in other network layers. Some layers may evolve largely independently of the rest of the network, whereas links in other layers may be highly predictive of links in the target layer. In our proposed method, a weight is assigned to each layer based on its influence on the target layer. Weights are calculated using a likelihood function:

$$w_i = Likelihood(\text{Link in } L^{Target} | \text{Link in } L^i) \tag{1.7}$$

where L^i and w_i represent the ith layer and the weight calculated for it respectively. L^{Target} indicates the target layer for which we want to predict future links. The *Likelihood* function computes the similarity between the target layer and the ith layer; to do this, we use the current ratio of overlapping edges. Next, we calculate weights for every node pair by checking the link correspondence between two layers using the likelihood of a link being present in the target layer given the existence of the link in the other layer at any other previous snapshot. This orders other layers in terms of their relative importance for a specific target layer. The process assigns higher weights to node pairs which occur in more than one layer (multiplex edges). The rate of link formation is incorporated into the model as the first term of the edge weight. Algorithm 1 shows the process of assigning likelihoods to layers and reweighting the adjacency matrix.

Algorithm 1 Likelihood Assignment and Edge Weighting

Input: Edge sets $(E^1, ..., E^M)$ for M layers where E^α is the edge set of target layer
Output: E^α_w weighted adjacency matrix for layer α (target layer)
//Calculate weights for the layers
for $i \in \{1, 2, ..., M\} - \{\alpha\}$ **do**
    $w_i = \textit{Likelihood}(\text{Link in } L^\alpha | \text{Link in } L^i)$
end for
//Weighting target layer
for edge $e \in E^\alpha$ **do**
    $w_e = \textit{rate} + \sum_{i=1 \& i \neq \alpha}^{M} w_i \times \textit{linkExist}(e)$
end for

The term *rate* is defined as the average value of the source node's out-degree over previous timesteps. Function *linkExist* is used to obtain information about a link's existence in other layers during previous snapshots. It checks each layer for the presence of an edge and returns 1 if an edge is present in that layer.

1.6.8 Node Similarity Metrics

This section provides a brief description of the topological and path-based metrics for encoding node similarity that are used within our MLP framework to create ranked score lists for each node pair. These techniques are often used in isolation as unsupervised methods for link prediction. Note that $\Gamma(x)$ stands for the set of neighbors of vertex x while $w(x, y)$ represents the weight assigned to the interaction between node x and y. We use the same metrics used by RPM: 1) common neighbors (CN), 2) Jaccard's coefficient (JC), 3) preferential attachment

(PA), and 4) Adamic-Adar coefficient (AA). Additionally, MLP uses the following metrics:

- **Resource Allocation (RA)**
  RA was first proposed in Zhou et al. (2009) and is based on physical processes of resource allocation:

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(y,z)}{\sum_{c \in \Gamma(z)} w(z,c)} \tag{1.8}$$

- **Page Rank (PR)**
  The PageRank algorithm (Brin and Page, 2012) measures the significance of a node based on the significance of its neighbors. We use the weighted PageRank algorithm proposed inDing (2011):

$$PR_w(x) = \alpha \sum_{k \in \Gamma(x)} \frac{PR_w(x)}{L(k)} + (1-\alpha) \frac{w(x)}{\sum_{y=1}^{N} w(y)} \tag{1.9}$$

  where $L(x)$ is the sum of outgoing link weights from node x, and $\sum_{y=1}^{N} w(y)$ is the total weight across the whole network.
- **Inverse Path Distance (IPD)**
  The Path Distance measure for unweighted networks simply counts the number of nodes along the shortest path between x and y in the graph. Note that $PD(x,y) = 1$ if two nodes x and y share at least one common neighbor. In this article, the Inverse Path Distance is used to measure the proximity between two nodes, where:

$$IPD(x,y) = \frac{1}{PD(x,y)} \tag{1.10}$$

  IPD is based on the intuition that nearby nodes are likely to be connected. In a weighted network, IPD is defined by the inverse of the shortest weighted distance between two nodes.
- **Product of Clustering Coefficient (PCF)**
  The clustering coefficient of a vertex v is defined as:

$$PCF(v) = \frac{3 \times \#\text{ of triangles adjacent to v}}{\#\text{ of possible triples adjacent to v}} \tag{1.11}$$

  To compute a score for link prediction between the vertex x and y, one can multiply the clustering coefficient score of x and y.

1.6.9 Temporal Link Structure

Given the network history for T time periods, we need to capture the temporal dependencies of the coevolution process. To do so, our framework uses a weighted exponentially decaying model (Acar et al., 2009). Let $\{Sim_t(i,j), t = t_0+1, ..., t_0+T\}$ be a time series of similarity score matrices generated by a node similarity metric on a sliding window of T successive temporal slices. An aggregated weighted similarity matrix is constructed as follows:

$$Sim_{(t_0+1)\sim(t_0+T)}(i,j) = \sum_{t=t_0+1}^{t_0+T} \theta^{t_0+T-t} Sim_t(i,j) \tag{1.12}$$

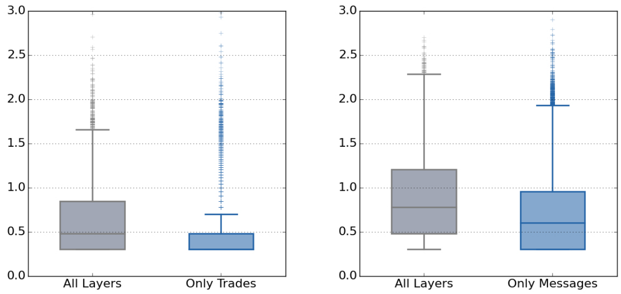
where the parameter $\theta \in [0,1]$ is the smoothing weight for previous time periods. Different values of θ modify the importance assigned to the most or least recent snapshots before current time $t+1$. This procedure generates a composite temporal score matrix for every node similarity metric. $Sim_{(t_0+1)\sim(t_0+T)}$ (shortened to Sim) is used by the algorithm as a summary of network activity, encapsulating the temporal evolution of the similarity matrix.

1.6.10 Rank Aggregation

Before describing the final step of our approach, let us briefly discuss existing methods for *ranked list aggregation/rank aggregation*. List merging or list aggregation refers to the process of combining a number of lists with the same or different numbers of elements in order to get one final list including all the elements. In rank aggregation, the order or rank of elements in input lists is also taken into consideration. The input lists can be categorized as *full, partial, or disjoint lists*. Full lists contain exactly the same elements but with a different ordering, partial lists may have some of the elements in common but not all, and disjoint lists have completely different elements. In this case, we are only dealing with full lists since each similarity metric produces a complete list for the same set of pairs, differing only in ordering.

In rank aggregation, distance metrics are used to find the disagreement between two lists/rankings. In general, any method of rank aggregation is desired to produce an aggregate ranking with minimal total disagreement among the input lists. Two well-known distance measures are:

- **Spearman Footrule Distance:** This computes the distance between



(a) (b)

Figure 1.18 Log scale box-whisker plots for user interactions in different layers of the network: (a) Travian (Trades) (b) Travian (Messages)

two ranked lists by computing the sum of differences in rankings of each element. Formally, it is given by:

$$F(L_1, L_2) = \sum_{i \in n} |L_1(i) - L_2(i)| \tag{1.13}$$

- **Kendall Tau Distance:** This counts the number of pairs of elements that have opposite rankings in the two input lists i.e. it calculates the pairwise disagreements.

$$K(L_1, L_2) = |(i,j) s.t. L_1(i) \leq L_2(j) \& L_1(i) \geq L_2(j)| \tag{1.14}$$

where L_1 and L_2 are the input lists and $L_1(i)$ and $L_2(i)$ represent the ranks of element i in the two lists correspondingly.

Rank aggregation methods can be categorized into two types: order-based and score-based. Order-based methods use the rank information (Liu et al., 2007) while score-based aggregation methods use score information from individual rankers. Several rank aggregation methods are described in (Sculley, 2007), including Borda's, Markov chain, and median rank methods. Borda's method is a *rank-then-combine* method originally proposed to obtain a consensus from a voting system. Since it is based on the absolute positioning of the rank elements and not their relative rankings, it can be considered a truly positional method. For every element

Figure 1.19 Heatmap representing the edge overlap between pairs of layers for the Travian dataset

in the lists, a Borda score is calculated and elements are ranked according to this score in the aggregated list. For a set of complete ranked lists $L = [L_1, L_2, L_3, ..., L_k]$, the Borda score for an element i and a list L_k is given by:

$$B_{L_k}(i) = \{count(j)|L_k(j) < L_k(i) \& j \in L_k\} \tag{1.15}$$

The total Borda score for an element is given as:

$$B(i) = \sum_{t=1}^{k} B_{L_t(i)} \tag{1.16}$$

Borda's method is computationally cheap, which is a highly desirable property for link prediction in large networks.

Algorithm 2 shows our proposed framework which incorporates edge weighting, the temporal decay model, and rank aggregation to produce an accurate prediction of future links in a dynamic multiplex network. The Borda function produces the final output of the MLP framework. Results of the proposed algorithm are compared with other state-of-the-art techniques in the next section.

1.6.11 Experimental Study

To investigate the impact of each component of our proposed method, not only do we compare our results with two other approaches for fusing cross-layer information, but we also analyze the performance of ab-

Algorithm 2 Multiplex Link Prediction Framework (MLP)

```
Input: Weighted edge sets of the target layer for T previous snapshots

Output: Temporal aggregated score matrix S for the target layer
for each node similarity metric u do
    for t ∈ {1, ..., T} do
        Calculate score matrix Sim^u_{t0+t}
    end for
    Calculate temporal similarity matrix Sim^u
end for
Final score matrix S = Borda(Sim^1, ..., Sim^u)
```

Table 1.5 *Dataset Summary: Number of edges, nodes, and snapshots for each network layer*

Dataset		**Travian**
No. of Nodes		2,809
No. of Snapshots		30
Layers/No. of Edges	Trades	87,418
	Messages	44,956

lated versions of our method. The complete method, MLP (Hybrid), is compared with MLP (Decay Model + Rank Aggregation) and MLP (Weighted + Rank Aggregation). All of the algorithms were implemented in Python and executed on a machine with the Intel(R) Core i7 CPU and 24GB of RAM for the purpose of fair comparison. Our implementation uses Apache Spark to speed the link prediction process.

1.6.12 Analysis of Cross-layer Interaction

Figure 1.18 shows log scale box-whisker plots that depict the frequency of interactions between users who are connected across multiple layers. We compare the frequency of interactions in cases where the node pair is connected on all layers vs. the frequency of being connected in a single layer. As expected, in cases where users are connected on all layers, the number of interactions is higher. The heatmap of the number of overlapping edges between different network layers (Figure 1.19) suggests that a noticeable number of edges are shared between all layers. This clearly indicates the potential value of cross-layer information for the link prediction task on these datasets. Our proposed likelihood weighting method effectively captures the information revealed by our analysis.

1.6.13 Performance of Multilayer Link Prediction

For our experiments, we adopted a moving-window approach to evaluate the performance of our temporal multiplex link prediction algorithm. Given a specified window size T, for each time period $t(t > T)$, graphs of T previous periods $(G_{t-T}, ..., G_{t-1})$ (where each graph consists of M layers) are used to predict links that occur at the target layer α in the current period (G_t^α). To assess our proposed framework and study the impact of its components, we compare against the following baselines:

- **MLP (Hybrid)**: incorporates all elements discussed in the framework section. It utilizes the likelihood assignment and edge weighting procedure to extract cross-layer information. Node similarity scores are modified using the temporal decay model and combined with Borda rank aggregation.
- **MLP (Likelihood + Rank Aggregation)**: This method only uses the aggregated scores calculated from the graphs weighted with cross-layer information. It does not consider the temporal aspects of network coevolution.
- **MLP (Decay Model + Rank Aggregation)**: This method does not use the cross-layer weighting scheme and relies on temporal information alone to predict future links. The final aggregated score matrix is calculated based on forecast values at time t for each node similarity metric using the decay model.
- **Likelihood**: Weights generated by the cross-layer likelihood assignment procedure are treated as scores for every node pair. We then sort the pairs based on their score and calculate the AUROC.
- **Rank Aggregation**: This method is a simple aggregated version of all unsupervised scoring methods using the Borda's rank aggregation method applied to node similarity metrics from the target layer.
- **Unsupervised Methods**: The performance of our proposed framework is compared with eight well-known unsupervised link prediction methods described in proposed method under node similarity metrics. All unsupervised methods are applied to the binary static graph from time 0 to $t-1$ in order to predict links at time t. Only the structure of the target layer is used.

- **Average Aggregation**: In order to extend the rank aggregation model to include information from other layers of the network, we use the idea proposed in Pujari and Kanawati (2015). Node similarity metrics are aggregated across all layers. So for attribute X (Common Neighbors, Adamic/Adar, etc.) over M layers the following is defined:

$$X(u,v) = \frac{\sum_{\alpha=1}^{M} X(u,v)^{\alpha}}{M} \tag{1.17}$$

  where $X(u,v)$ is the average score for nodes u and v across all layers and $X(u,v)^{\alpha}$ is the score at layer α. Borda's rank aggregation is then applied to the extended attributes to calculate the final scoring matrix.
- **Entropy Aggregation**: Entropy aggregation is another extended rank aggregation model proposed in Pujari and Kanawati (2015) where $X(u,v)$ is defined as follows:

$$X(u,v) = -\sum_{\alpha=1}^{M} \frac{X(u,v)^{\alpha}}{X_{total}} \log(\frac{X(u,v)^{\alpha}}{X_{total}}) \tag{1.18}$$

  where $X_{total} = \sum_{\alpha=1}^{M} X(u,v)^{\alpha}$. The entropy based attributes are more suitable for capturing the distribution of the attribute value over all dimensions. A higher value indicates a uniform distribution of attribute values across the multiplex layers.
- **Multiplex Unsupervised Methods**: Finally, using the definition of core neighborhood proposed in Hristova et al. (2015), we extend four unsupervised methods (Common Neighbors, Preferential Attachment, Jaccard Coefficient and Adamic/Adar) to their multiplex versions.

Table 1.6 shows the results of different algorithms on the Travian dataset. Bold numbers indicate the best results on each target layer considered; MLP (Hybrid) is the best performing algorithm.

1.6.14 Discussion

In this section, we discuss the most interesting findings:

Does rank aggregation improve the performance of the unsupervised metrics? As shown in Table 1.6, although the aggregated scores matrix produced by Borda's method achieves better results than unsupervised methods in one case (Travian message) and comparable results on Travian trade, it is not able to significantly outperform all unsupervised methods in any of the networks. As discussed before, we are using the simple Borda method for the rank aggregation which does not consider the effect of each ranker on the final performance. While adding weights to the rankers or using more complex rank aggregation models such as Kemeny might achieve better results, it has been shown that those approaches have high computational complexity which makes them less suitable for large real-world networks (Pujari and Kanawati, 2012; Tabourier et al., 2014). Despite the fact that the rank aggregation alone does not significantly improve the overall performance of the link prediction task, it enables us to effectively fuse different kinds of information (edge and node features, nodes similarity, etc.).

On the other hand, the Average and Entropy Aggregation methods, which are designed to consider attribute values from other layers, are able to outperform regular Rank Aggregation and MLP (Decay Model + Rank Aggregation). However, both methods use the static structure of all snapshots from time 0 to $t-1$, while MLP (Decay Model + Rank Aggregation) only incorporates the past T snapshots which makes it more suitable for large networks.

Does the likelihood assignment procedure outperform the unsupervised scores? To study the ability of our likelihood weighting method to model the link formation process, we generate results for two methods: using likelihood explicitly as a scoring method as well as using the values to generate a weighted version of the networks. First, the *Likelihood* method is used in isolation to demonstrate the prediction power of its weights as a new scoring approach. Table 1.6 shows significant improvements on unsupervised scores as well as the aggregated version of them. As expected, the more overlap between the target layer and predictor layers, the more performance improvement *Likelihood* achieves. As an example, Likelihood achieves $\sim 7\%$ of improvement on Travian (Trade) compared with $\sim 5\%$ of improvement on Travian (Message). Not only is there a lower rate of overlapping edges between those layers, but also the number of interactions is higher than the two other layers.

On the other hand, the method introduced in Algorithm 1 generates a weighted version of input graphs which is used to generate a weighted version of unsupervised methods to produce the final scoring matrix. This paired with the rank aggregation method generates significantly better average AUROC performance compared with other proposed methods. Also, when temporal information from previous snapshots of the network is included, MLP (Hybrid) outperforms other variants of MLP as well as well-known unsupervised methods. This indicates

Table 1.6 *AUROC performances for a target layer averaged over all snapshots with a sliding time window of $T = 3$. Variants of our proposed framework are shown at the top of the table, followed by standard unsupervised methods. The algorithms shown in the bottom half of the table are techniques for multiplex networks proposed by other research groups. The best performer is marked in bold.*

Algorithms / Networks	Trade	Message
MLP (Hybrid)	**0.821±0.001**	**0.803±0.002**
MLP (LH/RA)	0.802±0.001	0.790±0.0021
MLP (DM/RA)	0.722±0.002	0.731±0.002
Likelihood	0.770±0.033	0.760±0.041
Rank Aggregation	0.694±0.001	0.712±0.001
Common Neighbors	0.656±0.002	0.667±0.002
Jaccard Coefficient	0.628±0.002	0.680±0.003
Preferential Attachment	0.709±0.002	0.637±0.001
Adamic/Adar	0.635±0.003	0.700±0.003
Resource Allocation	0.625±0.005	0.690±0.003
Page Rank	0.595±0.0016	0.687±0.002
Inverse Path Distance	0.572±0.003	0.650±0.003
Clustering Coefficient	0.580±0.002	0.633±0.003
Average Aggregation	0.744±0.030	0.752±0.020
Entropy Aggregation	0.731±0.004	0.763±0.020
Multiplex CN	0.729±0.0040	0.643±0.013
Multiplex JC	0.666±0.031	0.619±0.012
Multiplex PA	0.722±0.010	0.646±0.012
Multiplex AA	0.671±0.010	0.690±0.031

the power of overlapping links in improving the performance of link prediction in coevolving multiplex networks.

Does including temporal information improve AUROC performance? The importance of incorporating temporal information into link prediction has been discussed in our previous work (Hajibagheri et al., 2016). However, here we are interested in analyzing the impact of this information on improving the performance of MLP. For that purpose, first, the decay model is employed in MLP (Decay Model + Rank Aggregation) to determine whether it improves the results generated by the aggregated score matrix. The final aggregated score matrix is calculated based on forecast values at time t for each unsupervised method using the decay model. As expected, this version of MLP is able to achieve up to $\sim 3\%$ of AUROC improvement using only information from the last three snapshots of the Travian network. On the other hand, we observed the same pattern when the decay model was added to MLP

(Hybrid) along with likelihood and rank aggregation. Using the scores generated by our hybrid approach outperformed all other proposed and existing methods. The results presented here have been obtained using $T = 3$ for the Travian dataset. While for Travian layers, increasing the value of T tends to improve the prediction performance slightly until $T = 3$; higher values of T may decrease the performance. Similarly, the value of θ is set to 0.4.

In summary, MLP (Decay Model + Rank Aggregation) is able to achieve results comparable to other baseline methods except Average and Entropy Aggregation since they benefit from the entire graph structure. Although rank aggregation by itself is not able to significantly improve the performance of unsupervised methods, paired with decay models and taking temporal aspects of the network, it can achieve better performance. On the other hand, the multiplex versions of the neighborhood based unsupervised methods are able to improve average AUROC performance, however the results are inconsistent and they achieve lower performance in many cases. Finally, both MLP (Hybrid) and MLP (Likelihood + Rank Aggregation) achieve higher performance compared with all other methods, illustrating the importance of the cross-layer information created by the network coevolution process. A paired two-sample t-test is used to indicate the significance of the results produced by each method where the p-value is smaller than 0.0001. It is worth mentioning that, even though MLP (Hybrid) is able to outperform all other methods, its performance is not significantly better than MLP (Likelihood + Rank Aggregation) in the case of Travian (Message).

In summary, MLP (Multiplex Link Prediction) employs a holistic approach to accurately predict links in dynamic multiplex networks using a collection of topological metrics, the temporal patterns of link formation, and overlapping edges created by network coevolution. Our analysis on real-world networks created by a variety of social processes suggests that MLP effectively models multiplex network coevolution.

1.7 Conclusion

The Travian massively multiplayer online game has served as a valuable testbed, enabling us to evaluate our social modeling algorithms in a complex and rich environment. Due to the dearth of publicly available data, many of the published prediction models have only been tested on coauthorship networks, such as DBLP and arXiv. However our results show that networks formed through different social processes (e.g., aggression vs. communication) exhibit different characteristics, necessitating experimentation on many types of datasets. Although most of the research described in this chapter has been conducted on the communication and trade layers of the Travian multiplex network, our current work focuses on the problem of leveraging information from the attack layer to improve our models of alliance evolution and link formation. The attack network layer is particularly challenging since it contains fewer interactions, and the standard topological metrics are not good predictors of its future structure. Also we plan to include additional network layers to represent alliance membership and geographic proximity; these relationships are semantically slightly different from the other layers because they are based on long-term relationships, rather than a series of transactions. Using a combination of our three techniques, D-GT, RPM, and MLP, we are able to successfully model the changes in community structure, rate of link formation, and the coevolution of different network layers. In future work we plan to introduce a single unified model, capable of exploiting dependencies between the dynamics of different processes.

1.8 Acknowledgments

This contribution is a culmination of a large, international and multi-year research effort. It is funded in part by the U.S. National Science Foundation (IIS-0838402, IIS-0838231, IIS-1110868, and ACI-1429160), U.S. Office of Naval Research (N000141010091, N000141410489, N0001415P1187, N000141612016, and N000141612412), U.S. Air Force Research Lab, U.S. Army Research Office (W911NF-16-1-0189), U.S. Defense Advanced Research Projects Agency (W31P4Q-17-C-0059), the Deutsche Forschungsgemeinschaft (German National Science Foundation) in Bonn, Germany, research grants, Travian Games GmbH, Munich and the Jerry L. Maulden/Entergy Fund at the University of Arkansas at Little Rock to Nitin Agarwal and Rolf Wigand. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations. The researchers gratefully acknowledge the support.

References

Acar, Evrim, Dunlavy, Daniel M, and Kolda, Tamara G. 2009. Link prediction on evolving data using matrix and tensor factorizations. Pages 262–269 of: *Workshops at IEEE International Conference on Data Mining*.

Adamic, Lada A, and Adar, Eytan. 2003. Friends and neighbors on the web. *Social Networks*, **25**(3), 211–230.

Al Hasan, Mohammad, and Zaki, Mohammed J. 2011. A survey of link prediction in social networks. Pages 243–275 of: *Social Network Data Analytics*. Springer.

Albert, Réka, and Barabási, Albert-László. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74**(1), 47.

Alvari, Hamidreza, Hashemi, Sattar, and Hamzeh, Ali. 2011. Detecting overlapping communities in social networks by game theory and structural equivalence concept. Pages 620–630 of: *Artificial Intelligence and Computational Intelligence*. Springer Berlin Heidelberg.

Alvari, Hamidreza, Hajibagheri, Alireza, Sukthankar, Gita, and Lakkaraju, Kiran. 2016. Identifying community structures in dynamic networks. *Social Network Analysis and Mining*, **6**(1), 77.

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

Barabási, Albert-László, and Albert, Réka. 1999. Emergence of scaling in random networks. *Science*, **286**(5439), 509–512.

Barabási, Albert-László, et al. 2009. Scale-free networks: a decade and beyond. *Science*, **325**(5939), 412.

Benevenuto, Fabrício, Rodrigues, Tiago, Cha, Meeyoung, and Almeida, Virgílio. 2009. Characterizing user behavior in online social networks. Pages 49–62 of: *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*. ACM.

Bennerstedt, U., Ivarsson, J., and Linderoth, J. 2012. How gamers manage aggression: Situating skills in collaborative computer games. *Computer-Supported Collaborative Learning*, **7**, 43–61.

Berlingerio, Michele, Bonchi, Francesco, Bringmann, Björn, and Gionis, Aristides. 2009. Mining graph evolution rules. Pages 115–130 of: *Machine Learning and Knowledge Discovery in Databases*. Springer.

Bianconi, Ginestra. 2013. Statistical mechanics of multiplex networks: Entropy and overlap. *Physical Review E*, **87**(6), 062806.

Blondel, Vincent D, Guillaume, Jean-Loup, Lambiotte, Renaud, and Lefebvre, Etienne. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008.

Brin, Sergey, and Page, Lawrence. 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, **56**(18), 3825–3833.

Bringmann, Björn, Berlingerio, Michele, Bonchi, Francesco, and Gionis, Arisitdes. 2010. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems*, **25**(4), 26–35.

Broder, Andrei, Kumar, Ravi, Maghoul, Farzin, Raghavan, Prabhakar, Rajagopalan, Sridhar, Stata, Raymie, Tomkins, Andrew, and Wiener, Janet. 2000. Graph structure in the web. *Computer Networks*, **33**(1), 309–320.

Buldyrev, Sergey V, Parshani, Roni, Paul, Gerald, Stanley, H Eugene, and Havlin, Shlomo. 2010. Catastrophic cascade of failures in interdependent networks. *Nature*, **464**(7291), 1025–1028.

Buono, Camila, Alvarez-Zuzek, Lucila G, Macri, Pablo A, and Braunstein, Lidia A. 2014. Epidemics in partially overlapped multiplex networks. *PloS One*, **9**(3), e92200.

Cazabet, Rémy, Amblard, Frédéric, and Hanachi, Chihab. 2010. Detection of overlapping communities in dynamical social networks. Pages 309–314 of: *IEEE International Conference on Social Computing*.

Clauset, Aaron, Shalizi, Cosma Rohilla, and Newman, Mark EJ. 2009. Power-law distributions in empirical data. *SIAM Review*, **51**(4), 661–703.

Cook, Diane J, Crandall, Aaron, Singla, Geetika, and Thomas, Brian. 2010. Detection of social interaction in smart spaces. *Cybernetics and Systems: An International Journal*, **41**(2), 90–104.

Danon, Leon, Diaz-Guilera, Albert, Duch, Jordi, and Arenas, Alex. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, **2005**(09), P09008.

Dawes, Robyn M. 1980. Social dilemmas. *Annual Review of Psychology*, **31**(1), 169–193.

De Domenico, Manlio, Solé-Ribalta, Albert, Cozzo, Emanuele, Kivelä, Mikko, Moreno, Yamir, Porter, Mason A, Gómez, Sergio, and Arenas, Alex. 2013a. Mathematical formulation of multilayer networks. *Physical Review X*, **3**(4), 041022.

De Domenico, Manlio, Solé, Albert, Gómez, Sergio, and Arenas, Alex. 2013b. Random walks on multiplex networks. *arXiv preprint arXiv:1306.0519*.

Ding, Ying. 2011. Applying weighted PageRank to author citation networks. *Journal of the American Society for Information Science and Technology*, **62**(2), 236–245.

Faloutsos, Michalis, Faloutsos, Petros, and Faloutsos, Christos. 1999. On power-law relationships of the internet topology. Pages 251–262 of: *ACM SIGCOMM Computer Communication Review*, vol. 29.

Fortunato, Santo. 2010. Community detection in graphs. *Physics Reports*, **486**(3), 75–174.

Getoor, Lise, and Diehl, Christopher P. 2005. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, **7**(2), 3–12.

Gomez, Sergio, Diaz-Guilera, Albert, Gomez-Gardenes, Jesus, Perez-Vicente, Conrad J, Moreno, Yamir, and Arenas, Alex. 2013. Diffusion dynamics on multiplex networks. *Physical review letters*, **110**(2), 028701.

Gómez-Gardenes, Jesús, Reinares, Irene, Arenas, Alex, and Floría, Luis Mario. 2012. Evolution of cooperation in multiplex networks. *Scientific Reports*, **2**.

Hajibagheri, Alireza, Sukthankar, Gita, and Lakkaraju, Kiran. 2016 (June). Leveraging Network Dynamics for Improved Link Prediction. In: *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*.

Hristova, Desislava, Noulas, Anastasios, Brown, Chloë, Musolesi, Mirco, and Mascolo, Cecilia. 2015. A Multilayer Approach to Multiplexity and Link Prediction in Online Geo-Social Networks. *arXiv preprint arXiv:1508.07876*.

Huang, Zan, and Lin, Dennis KJ. 2009. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, **21**(2), 286–303.

Humphreys, M., and Weinstein, J. 2008. Who Fights? The Determinants of Participation in Civil War. *American Journal of Political Science*, **52**(2), 436–455.

Keegan, B., Ahmed, M., Williams, D., Srivastava, J., and Contractor, N. 2010. Dark Gold: Statistical Properties of Clandestine Networks in Massively Multiplayer Online Games. Pages 201–208 of: *IEEE International Conference on Social Computing*.

Kivela, Mikko, Arenas, Alex, Barthelemy, Marc, Gleeson, James, Moreno, Yamir, and Porter, Mason. 2014. Multilayer networks. *Journal of Complex Networks*, **2**, 203–271.

Kolbert, Elizabeth. 2001. Pimps and dragons: How an online world survived a social breakdown. *The New Yorker*, **77**(13), 88–98.

Korsgaard, M., Picot, A., Wigand, Rolf, Welpe, I., and Assmann, J. 2010. Cooperation, Coordination, and Trust in Virtual Teams: Insights from Virtual Games. In: *Online Worlds: Convergence of the Real and the Virtual*.

Kurant, Maciej, and Thiran, Patrick. 2006. Layered complex networks. *Physical Review Letters*, **96**(13), 138701.

Lancichinetti, Andrea, Radicchi, Filippo, Ramasco, José J, Fortunato, Santo, et al. 2011. Finding statistically significant communities in networks. *PloS One*, **6**(4), e18961.

Leskovec, Jure, Backstrom, Lars, Kumar, Ravi, and Tomkins, Andrew. 2008. Microscopic evolution of social networks. Pages 462–470 of: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Liben-Nowell, David, and Kleinberg, Jon. 2003. The Link Prediction Problem for Social Networks. Pages 556–559 of: *Proceedings of the International Conference on Information and Knowledge Management*.

Liben-Nowell, David, and Kleinberg, Jon. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, **58**(7), 1019–1031.

Lichtenwalter, Ryan N., Lussier, Jake T., and Chawla, Nitesh V. 2010. New perspectives and methods in link prediction. Pages 243–252 of: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Liu, Yu-Ting, Liu, Tie-Yan, Qin, Tao, Ma, Zhi-Ming, and Li, Hang. 2007. Supervised rank aggregation. Pages 481–490 of: *Proceedings of the International Conference on World Wide Web*. ACM.

MacKay, David JC. 2003. *Information theory, inference and learning algorithms*. Cambridge University Press.

Min, Byungjoon, and Goh, K-I. 2013. Layer-crossing overhead and information spreading in multiplex social networks. *arXiv preprint arXiv:1307.2967*.

Newman, M. E. J. 2001. Clustering and preferential attachment in growing networks. *Phys. Rev. E*.

Newman, M. E. J. 2002. Assortative Mixing in Networks. *Physical Review Letters*, **89**(20), 208701.

Nicosia, Vincenzo, Bianconi, Ginestra, Latora, Vito, and Barthelemy, Marc. 2013. Growing multiplex networks. *Physical Review Letters*, **111**(5), 058701.

Piraveenan, Mahendra, Chung, Kon Shing Kenneth, and Uddin, Shahadat. 2012. Assortativity of links in directed networks. In: *Fundamentals of Computer Science*.

Potgieter, Anet, April, Kurt A, Cooke, Richard JE, and Osunmakinde, Isaac O. 2009. Temporality in link prediction: Understanding social complexity. *Emergence: Complexity & Organization (E: CO)*, **11**(1), 69–83.

Pujari, Manisha, and Kanawati, Rushed. 2012. Supervised rank aggregation approach for link prediction in complex networks. Pages 1189–1196 of: *Proceedings of the International World Wide Web Conference*.

Pujari, Manisha, and Kanawati, Rushed. 2015. Link prediction in multiplex networks. *Networks and Heterogeneous Media*, **10**(1), 17–35.

Rosvall, Martin, and Bergstrom, Carl T. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, **105**(4), 1118–1123.

Roy, A., Borbora, Z., and Srivastava, J. 2013. Socialization and Trust Formation: A Mutual Reinforcement? An Exploratory Analysis in an Online Virtual Setting. Pages 653–660 of: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

Saumell-Mendiola, Anna, Serrano, M Ángeles, and Boguná, Marián. 2012. Epidemic spreading on interconnected networks. *Physical Review E*, **86**(2), 026106.

Scott, John. 2012. *Social Network Analysis*. Sage.

Sculley, D. 2007. Rank Aggregation for Similar Items. Pages 587–592 of: *SIAM International Conference on Data Mining*.

Snijders, T., van de Bunt, G., and Steglich, C. E. G. 2010. Introduction to Actor-Based Models for Network Dynamics. *Social Networks*, **32**, 44–60.

Soares, Paulo Ricardo da Silva, and Prudêncio, Ricardo Bastos Cavalcante. 2012. Time series based link prediction. Pages 1–7 of: *International Joint Conference on Neural Networks*. IEEE.

Sole-Ribalta, Albert, De Domenico, Manlio, Kouvaris, Nikos E, Diaz-Guilera, Albert, Gomez, Sergio, and Arenas, Alex. 2013. Spectral properties of the Laplacian of multiplex networks. *Physical Review E*, **88**(3), 032807.

Strogatz, Steven H. 2001. Exploring complex networks. *Nature*, **410**(6825), 268–276.

Tabourier, Lionel, Bernardes, Daniel Faria, Libert, Anne-Sophie, and Lambiotte, Renaud. 2014. RankMerging: A supervised learning-to-rank framework to predict links in large social network. *arXiv preprint arXiv:1407.2515*.

Tan, Pang-Ning, Steinbach, Michael, and Kumar, Vipin. 2005. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Thurau, C., and Bauckhage, C. 2010. Analyzing the Evolution of Social Groups in World of Warcraft. Pages 170–177 of: *IEEE International Conference on Computational Intelligence in Games*.

Wang, Chao, Satuluri, Venu, and Parthasarathy, Srinivasan. 2007. Local probabilistic models for link prediction. Pages 322–331 of: *Seventh IEEE International Conference on Data Mining*.

Wigand, R., Agrawal, N., Osesina, O., Hering, W., Korsgaard, M., Picot, A., and Drescher, M. 2012. Social network indices as performance predictors in a virtual organization. Pages 144–149 of: *Computational Analysis of Social Networks*.

Wigand, Rolf T. 2017. Virtual Organization and Online Games. In: Lakkaraju, Kiran, Sukthankar, Gita, and Wigand, Rolf T. (eds), *Social Interaction in Virtual Worlds*. Cambridge University Press.

Xie, Jierui, Chen, Mingming, and Szymanski, Boleslaw K. 2013. LabelrankT: Incremental community detection in dynamic networks via label propagation. *arXiv preprint arXiv:1305.2006*.

Yee, N. 2006. The Labor of Fun: How Video Games Blur the Boundaries of Work and Play. *Games and Culture*, **1**(1), 68–71.

Zhou, Tao, Lü, Linyuan, and Zhang, Yi-Cheng. 2009. Predicting missing links via local information. *The European Physical Journal B*, **71**(4), 623–630.