Link prediction and link detection in sequences of large social networks using temporal and local metrics.

A dissertation submitted to the Department of Computer Science at the University of Cape Town in fulfilment of the requirements for the degree of master of science.

By
Richard Jeremy Edwin Cooke
November 2006

Supervised by
Dr. Anet Potgieter
and co-supervised by
Dr. Kurt April

Abstract

This dissertation builds upon the ideas introduced by Liben-Nowell and Kleinberg in *The Link Prediction Problem for Social Networks* [42]. Link prediction is the problem of predicting between which unconnected nodes in a graph a link will form next, based on the current structure of the graph. The following research contributions are made:

- Highlighting the difference between the link prediction and link detection problems, which have been implicitly regarded as identical in current research. Despite hidden links and forming links having very highly significant differing metric values, they could not be distinguished from each other by a machine learning system using traditional metrics in an initial experiment. However, they could be distinguished from each other in a "simple" network (one where traditional metrics can be used for prediction successfully) using a combination of new graph analysis approaches.

- Defining temporal metric statistics by combining traditional statistical measures with measures commonly employed in financial analysis and traditional social network analysis. These metrics are calculated over time for a sequence of sociograms. It is shown that some of the temporal extensions of traditional metrics increase the accuracy of link prediction.

- Defining traditional metrics using different radii to those at which they are normally calculated. It is shown that this approach can increase the individual prediction accuracy of certain metrics, marginally increase the accuracy of a group of metrics, and greatly increase metric computation speed without sacrificing information content by computing metrics using smaller radii. It also solves the "distance-three task" (that common neighbour metrics cannot predict links between nodes at a distance greater than three).

- Showing that the combination of local and temporal approaches to link prediction can lead to very high prediction accuracies. Furthermore in "complex" networks (ones where traditional metrics cannot be used for prediction successfully) local and temporal metrics become even more useful.

Acknowledgements

I thank my supervisors, Anet and Kurt, for their guidance and advice; the South African National Research Foundation for partly funding this research; my family for providing additional funding; Petter Holme, the owners of *Pussokram* and the owners of *Netcash* for providing the data on which I tested my ideas; and finally the makers of *Weka* and *Jung* for creating and freely distributing useful Java packages.

Contents

List of figures

List of tables

List of Terms

arff	Attribute-relation file format. A comma-separated value file format with header information prefacing the data formatted in a way the *Weka* statistical system can parse it to extract the type of each attribute (column) used.
Common neighbour	A common neighbour of two nodes is one that is adjacent to both the nodes. E.g. a node v_k is a common neighbour of v_i and v_j if and only if $v_k \in \Gamma(v_i)$ and $v_k \in \Gamma(v_j)$, where $\Gamma(v_i)$ is the set of neighbours of v_i. A common neighbour-based metric is a dyadic metric that includes a count of the common neighbours of the two nodes in question.
csv	Comma-separated values. A database text file format where lines in a file represent rows in a table and commas separate the data into columns. It is commonly used by spreadsheet programs.
Degree	The degree of a node is the number of links incident to the node, or the number of neighbours a node has.
Distance metric	A distance metric is a usually a dyadic metric that includes the number of nodes in one or more shortest paths between the two nodes in question.
Dyadic	Dyadic means relating to a dyad, or pair of nodes. Thus a dyadic metric is one that is calculated for two nodes.
Homophily	Homophily means that people tend to make friends with other people who are similar to themselves, in terms of geographic location, interests, culture, age and so on.
Kappa	$\frac{P(A)-P(E)}{1-P(E)}$, where P(A) is the total percentage accuracy of instances predicted by the learning system and P(E) is the total percentage accuracy of instances predicted by random guessing. A measure of prediction reliability greater than chance. It ranges from -100% to 100%.
Metric	A metric is a value calculated for some aspect of a graph that contains information describing that aspect. For instance, the popularity of an individual can be described by the metric *degree*, which is the number of neighbours that individual has.
Monadic	Monadic means relating to a single node. Thus a monadic metric is one that is calculated only for one node.

Pussokram	The Swedish online dating site from which a social network data set was obtained for this research.
Scale-free	This means that the fraction of nodes in a graph with degree d is proportional to $\frac{1}{d^c}$, where c is some constant. In other words, most nodes have the average degree and very few have either very high or very low degrees.
Small world network	A small world network is one where on average every pair of nodes can be connected through a short path within the network; and where the probability that two nodes are linked is greater if they share a neighbour (a high clustering coefficient).
Sociogram	Graph or social network. A mathematical graph (network, sociogram) where people are represented by nodes (vertices) and relationships between people are represented as links (arcs, edges).
Social network	See sociogram.
Social network analysis	The study of social networks. A multidisciplinary combination of graph theory, statistics, sociology, psychology, physics and computer science.
True positive rate	$\frac{TP}{TP+FN}$, where the numerator represents all the instance of the positive class we correctly predicted. The denominator is the sum of these instances, as well as the instances we did not predict. Thus the TP rate is the percentage of all links we discovered.
Weka	A freely available data mining software suite created in Java. It was used for logistic regression in this research.

Chapter 1. Introduction

This introductory chapter overviews the research undertaken, summarises the academic contributions of the dissertation and describes the structure of the discussion.

1.1. Research overview

This report describes research undertaken that builds upon the ideas introduced by Liben-Nowell and Kleinberg in *The Link Prediction Problem for Social Networks* [42] and chapter three from *An Algorithmic Approach to Social Networks*, Liben-Nowell's doctoral thesis [41]. Link prediction is the problem of predicting between which unconnected nodes in a graph (sociogram or network) a link (edge or arc) will form next, based on the current structure of the graph. Link prediction thus far has used simple traditional social network analysis metrics and met with some success. In this research the problem is explored more deeply using a data set obtained from messages exchanged on a dating social networking website. Differences between link detection and link prediction are hypothesised and investigated. Temporal statistical extensions of traditional metrics are proposed and compared against traditional metrics for prediction accuracy. In a further experiment, traditional metrics are defined and computed for different radii to ascertain whether increases in computational speed and prediction accuracy can be achieved. Finally, these extensions to metrics are combined and tested for increases in accuracy using the data set obtained from an email server's log file. This approximates how an intelligence analyst might use the ideas proposed in this dissertation in the real world.

1.2. Motivation

At the time of writing, link prediction in social networks is a relatively new problem, with the classic paper on the problem, [42], being written only in 2003. Thus there is a lot of scope for improving the approaches to solving the problem. Temporal approaches have been completely ignored by researchers and most research has been performed on a network at a single point in time [40]. These networks have also tended to be small, i.e. having few enough nodes to fit into an adjacency matrix in memory. This leads to unrealistic analysis approaches as most real world analysis would be performed on massive criminal intelligence databases, including large email logs collected on the Internet. Researchers have also not distinguished between link detection and link prediction and it seems to be implicitly assumed that the problems are identical. No distinction is made between the problems in nearly all of the papers considered. Only in a few papers is link detection even mentioned. Due to these gaps in research the work presented in this dissertation attempts to show

how temporal statistics in social network sequences can be found and how they contribute to prediction, how metrics can be redefined to be calculated faster and how link prediction and link detection differ. Link prediction is a mathematical and statistical technique that can be applied to many business, social and software problems. Link prediction has many applications. Some examples include:

- Identifying the structure of a criminal network (i.e. predicting missing links in a criminal network using incomplete data) [13].
- Overcoming the data-sparsity problem in recommender systems using collaborative filtering [35].
- Accelerating a mutually beneficial professional or academic connection that would have taken longer to form serendipitously [20].
- Improving hypertext analysis for information retrieval and search engines [31].
- Monitoring and controlling computer viruses that use email as a vector [43].
- Predicting which web pages users will next visit in order to improve the efficiency and effectiveness of a site's navigation [73].
- Helping to predict the spread of an entity through a network [71]. Examples include a disease, such as HIV, or information, such as a clothing fashion or rumour.

1.3. Contributions of this dissertation

This dissertation contributes to the following unresearched areas:

- Highlighting the difference between the link prediction and link detection problems, which have been implicitly regarded as identical in current research. It is shown through empirical testing that the two classes, hidden links and forming links, have very highly significant differing metric values and differing surrounding subgraph patterns. Despite this, they could not be distinguished from each other by a machine learning system using traditional metrics in an initial experiment. However, they could be distinguished from each other in a "simple" network (one where traditional metrics can be used for prediction successfully) using a combination of the approaches described below.
- Defining temporal metric statistics through combining traditional statistical measures with measures commonly employed in financial analysis and traditional social network analysis. These metrics are calculated over time for a sequence of sociograms and the usefulness of different types for link prediction (i.e. information content) are compared and evaluated. It is shown that some of the temporal extensions of traditional metrics increase the accuracy of link prediction.
- Defining traditional metrics using different radii to those at which they are normally

calculated. For instance this includes calculating common neighbour-based metrics by using not only common neighbours but also neighbours of neighbours, or calculating the betweenness of a node not using the whole graph but rather using only an egocentric subgraph of nodes up to three links away from the node under consideration. It is shown that this approach can increase the individual prediction accuracy of certain metrics, marginally increase the accuracy of a group of metrics, and greatly increase metric computation speed without sacrificing information content. It also solves the "distance-three task" (that common neighbour metrics cannot predict links between nodes at a distance greater than three) [42]. Such metrics can be quickly calculated even in graphs too large to fit into an adjacency matrix and could be distributed over multiple processors using software threads.

1.4. Experimental approach

This section presents a brief overview of the way the accuracy of the metrics was computed and evaluated. The data sets used were stored as tables of nodes, links and messages in a *MySQL* database. The data were accessed by a Java program written specifically for this research. It consisted of classes to store sociograms as temporal sequences, display egocentric subgraphs as graph diagrams in a user interface, compute the various traditional metrics and newly invented metrics for unconnected-, forming- and hidden links and output these metrics and the associated links to a comma-separated values (csv) file. These csv files were converted into attribute-relation file format (arff) files, which can be used as inputs for *Weka*, an open source data mining Java suite [70]. Weka was then used to perform logistic regressions on the given instances, using the computed metrics as attributes and the presence or absence of a new link as the variable to be predicted. Different combinations of metrics were tested and various statistics of classification accuracy were recorded. Additionally, the metric values for the classes were analysed in a traditional statistical hypothesis testing manner. The first data set used was obtained from the *Pussokram* Internet social networking portal. The second, email, data set used was constructed by stripping a server's log file (created by the *sendmail* UNIX email server program) into the bare essentials, discarding bad data (such as missing or unreachable addresses), matching the sender and recipient of emails sent and mapping the email addresses used to a unique positive integer to mask them and protect the privacy of individuals prior to analysis.

1.5. Evaluation criteria

The objective of this research is not only to present and discuss link prediction, but also to invent and implement new prediction techniques that maximise the number of forming links correctly predicted.

Since link prediction is a relatively new field there are few studies with which to compare accuracy. Furthermore, these studies differ in the data sets and prediction techniques used and the way the prediction accuracy is reported statistically. Thus the accuracy of the new techniques presented in this work had to be evaluated in relative isolation, but in a way that could be compared to future work. The traditional metrics and the new metrics were computed on the same data set and used for regression in the same statistical program. In this way the techniques can be compared to those that would have been used by other researchers. The accuracy of the predictions are also reported as an overall percentage of the total number of correct instances, the true positive rates for each class and as a kappa value, which compensates for the accuracy of a random prediction and allows for universal comparison in future research (these statistics are explained in the research methodology chapter).

1.6. Scope and limitations

The prediction accuracy of the system used relied both on the usefulness of the metrics and the statistical classification software. The statistical system used, Weka, may have influenced the outcome of the experiments in unforeseen ways. For instance, the learning- and classification algorithms used by Weka may be superior or inferior to those used by other researchers, thereby affecting the accuracy of these results in a way that is not due to the new ideas presented here. This possibility was mitigated by using the standard logistic regression classifier implemented by Weka. Additionally, and more unlikely, Weka may have software bugs that, for the given data, generated incorrect results that were in the expected accuracy range, and hence went undetected. However, no inexplicable or unexpected results appear to have been generated. The initial experiments were performed using only one data set. It is an established one that has been analysed in previous research [32][33], and the positive results obtained might not be repeatable on other data sets. However, the final chapter uses a real world email data set and combines the metrics to reassess the usefulness of the ideas proposed. Finally, all available sources, including websites, conference outputs, journal articles and books, were searched to find similar research. Although none was found, this does not preclude the possibility that similar techniques have been used previously. If this is so, then these previous techniques would have to be compared to the techniques presented in this research in a future study to compare the effectiveness of both.

1.7. Dissertation outline

This report is divided into eight chapters, starting with this short introductory chapter. Chapter two is a literature review of social network analysis and summarises important ideas in the fields, including link analysis, link prediction, metrics and applications of social network analysis. Chapter three

summarises statistical methods used in artificial intelligence for learning and inference and explains the statistical methodology used in the experiments presented here, based on the methodologies used by previous researchers. Chapter four is the first of the four experiments discussed. It presents the work done on the differences between link detection and link prediction. Chapter five is another experimental chapter and presents the work done on the use of temporal metrics in link prediction. Chapter six is the third and last experiment to present new ideas and presents the work done on local metric computations. Chapter seven presents the final experiment, which draws together aspects of temporal and local analysis using an email data set. Chapter eight is a short final chapter that concludes the discussion.

Chapter 2. Social network analysis background

This chapter gives a brief introduction to the theory and practice of social network analysis, with an emphasis on link prediction. It describes the theory and applications of social network analysis, describes current link prediction techniques and introduces metrics and other concepts that are used later in the discussion.

2.1. Social networks

A social network consists of a group of people and connections between them [69]. These connections can be any type of social link that implies a relationship between two people. There is an existing field of mathematics called graph theory that deals with any structure, like a social network, that can be represented by nodes and links [14][18]. The discipline of graph theory has been extended by researchers into the field of social network analysis. This discussion of some of the problems in social network analysis begins with an introduction to graph theory.

2.2. Graph theory

A social network is represented by a mathematical structure called a graph. A graph G is a structure consisting of a set of nodes V (also called vertices), and a set of links E (also called arcs or edges). Thus we can write $G=\langle V, E \rangle$. In this discussion nodes represent people and links represent a relationship between people. A link e (where $e \in E$) is a set of two nodes from the set V. For instance, we could have that $V=\{v_1, v_2, v_3, v_4, v_5\}$ and $E=\{e_1, e_2\}$. E could also be written as $E=\{\{v_1, v_3\}, \{v_3, v_5\}\}$, assuming that $e_1=\{v_1, v_3\}$ and $e_2=\{v_3, v_5\}$. In other words, this graph consists of five people named v_1, v_2, v_3, v_4, and v_5. v_1 and v_3 know each other and v_3 and v_5 know each other. This graph can be drawn as a diagram, where nodes are represented by dots and links are represented by lines linking nodes:

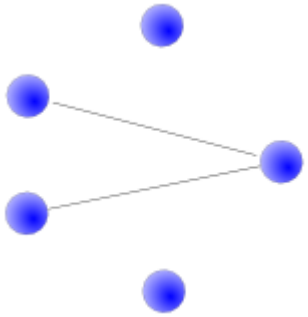

Figure 1. A graph with five nodes and two edges

These diagrams have been called sociograms by sociologists. In this discussion the terms graph, network, social network and sociogram are used interchangeably. The sociogram above may be easier to understand if we label the nodes:

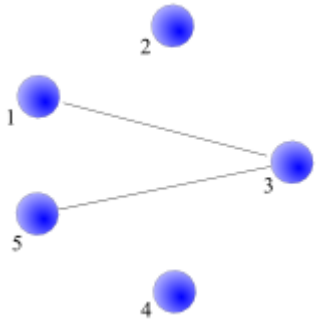

Figure 2. A graph with five labelled nodes and two edges

These graphs are bidirected[1], meaning that we are not concerned about the order of the nodes in a link. Either we are assuming two people who are linked know each other equally well or we do not mind that the relationship is one sided. With a directed graph however, we are concerned about the direction of a link between two nodes. For instance, if the graph discussed above were directed, it would look like this:

1 Also called "undirected" [69], but bidirected is used in this paper to emphasise that messages and emails are directed and links go in both directions.

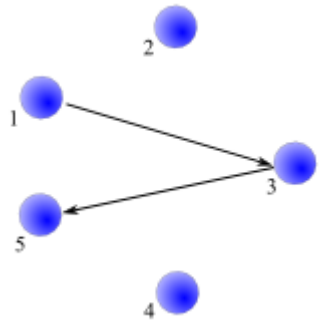

Figure 3. A directed labelled five node graph

An example of a one-sided relationship or directed link might be when one person has emailed another but the recipient of the email has not reciprocated. There exist other variations and sub-classifications of graphs that are used in various fields. One type that is commonly used in social network analysis is the bimodal graph [69]. A bimodal graph is a graph in which the nodes are divided into two distinct sets, where each set represents a different type of entity. For instance, we could create a bimodal graph where the two sets of nodes represent researchers and papers published. Links represent that a researcher was an author or co-author of a paper with another researcher. Alternatively, such an arrangement can be represented by a bipartite graph, where nodes are all of the same type, but are partitioned into two sets [69]. Links may join a node in one set only with a node in the other. A bipartite or bimodal graph is shown below, where the blue nodes could represent papers published and the orange nodes could represent authors.

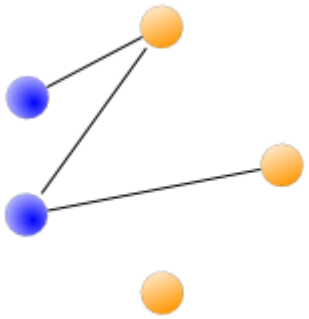

Figure 4. A bimodal or bipartite graph

For an introduction to graph theory [66] is recommended. And for a more comprehensive discussion [14], [18], [29], and [59] are recommended.

2.3. Links in social networks

Examples of a relationship between people studied in social network analysis include friendship,

having exchanged emails, or belonging to the same social club. In other disciplines nodes and links can be used to represent towns and highways, computers and cabling, or companies and ownership. In this research the relationship we are considering is a single message sent from one person to another. For instance, an email sent from one address to another address names and connects the two nodes in a directed link. To illustrate this, consider that all we know about a certain hypothetical group of people is that an email was sent from bill@mail.com to ted@mail.com. Then for this graph $V = \{bill@mail.com, ted@mail.com\}$ and $E = \{\{bill@mail.com, ted@mail.com\}\}$. Although there are more individuals in this group, we can record only those of which we have evidence.

A sociogram represents a complex system – a human society, or at least part of it. Complexity theory has found that complex systems occur on the edge of chaos [47]. For example, life cannot exist in a vacuum (complete order), as there are no environmental forces that cause change and adaptation. But nor can it exist in a sun (complete chaos), as a system ceases to function amidst total chaos. A compromise of semi-chaos is needed for adaptation and development to flourish [7]. In the same way, since social networks are representations of complexity, they are neither totally ordered lattices, nor random distributions of nodes and links, but rather have special properties of their own. These properties are described in the following subsections.

2.3.1. Small world networks

A small world network is one where on average every pair of nodes can be connected through a short path within the network; and where the probability that two nodes are linked is greater if they share a neighbour (a high clustering coefficient) [4]. The idea of a small world network was created by the American psychologist, Stanley Milgram, in the 1960's [65]. He found that randomly selected individuals could send a letter to a chosen recipient in another state through a remarkably short (approximately six people) chain of acquaintances. This led to popular constructions such as "six-degrees of separation", six-degrees of Kevin Bacon (actors linked to Kevin Bacon by chains of film performances) and the similar Erdös number (the number of scientists removed from co-publishing a paper with mathematician Paul Erdös). Social networks have the small world property. The Internet does too, if we consider web pages to be nodes and hypertext links to be links. Even artificial networks can be small worlds. An example is the Marvel comic book universe, consisting of 6486 comic characters (nodes), where links were co-appearances in comics [4]. It is also sometimes useful to design networks to be small world. For instance, Roumeliotis and Mataric experimented with the idea of creating small world networks for robots to avoid data traffic overload [62]. The robots communicated with nearby neighbours only and avoided broadcast messaging to create a small world communication system.

2.3.2. Scale-free networks

Small world networks usually (and almost always for large social networks) follow a power law distribution [3]. This means that the fraction of nodes with degree d (the number of links a node has) is proportional to $\frac{1}{d^c}$ where c is some constant. In other words, most nodes have the average degree and very few have either very high or very low degrees (humanly speaking, most of us are neither exceptionally popular or unpopular). Such networks are said to be scale-free or power-law graphs. Shown below on the left is a graph that is scale-free. The orange nodes have a far higher proportion of links than the blue nodes. The graph on the right is not scale-free as nearly all the nodes have an equal degree.

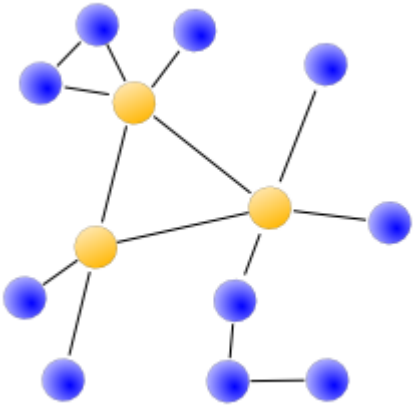

Figure 5. A scale-free graph

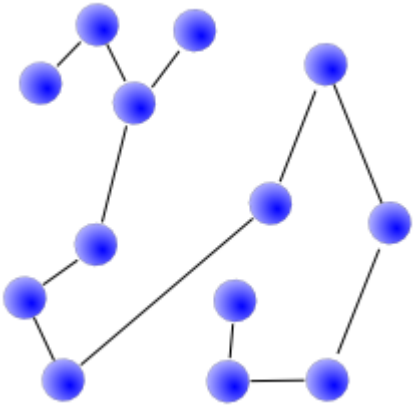

Figure 6. A non scale-free graph

A related term is preferential attachment – meaning that a new link is likely to be incident (or preferentially attach) to an existing node of high degree [3]. This is a property that social networks share with the Internet. Preferential attachment leads to the existence of scale-free networks. Most websites have the same small number of incoming and outgoing links, but a few are sites with large numbers of both types of links. The rise of blogging (web-logging) in the past few years has accelerated until approximately 80 000 blogs were being created every day in August 2005 [44]. As each blog is generally maintained by one individual the community of bloggers forms a classic social network. Other structures can also be modelled as social networks. For instance, academic papers and citations can be considered as nodes and links. The Gnutella peer to peer networked file sharing system is an example of a scale-free network [2]. The properties of such networks have both benefits and flaws. Crucitti, Latora, Marchiori and Rapisarda found that scale-free networks are highly error tolerant but vulnerable to intelligent attack [17]. When up to two percent of nodes are removed

randomly network communication will be completely unaffected. However, a network can be quickly crippled by removing only a few highly connected nodes.

2.3.3. Homophily and assortative mixing

Homophily means that people tend to make friends with other people who are similar to themselves, in terms of geographic location, interests, culture, age and so on. This psychosocial principle influences the structure of sociograms. For instance, researchers have found that sociograms exhibit assortative mixing, meaning that nodes of high degree tend to have neighbours of high degree [41]. Popular birds of a feather flock together, if you will. Newman [49] found that social networks (actor and academic publication collaboration networks) exhibit assortative mixing whereas technological networks (the Internet and the World Wide Web) and biological networks (protein interactions, food webs and neural networks) exhibit disassortative mixing. Disassortative mixing means that nodes of high degree tend to have neighbours of low degree. In terms of resilience, Newman notes that removing high degree nodes in a social network (for instance, to vaccinate against a sexually transmitted disease) will have little effect as high degree nodes are redundant. However, removing a high degree node in a technological network (such as the failure of an important Internet server) will be more likely to have a crippling effect as the node is not redundant.

2.4. Social network analysis

The study of sociograms, social network analysis, mainly involves computing various measures of graphs (called metrics), which provide useful sociological information. Modern researchers combine disciplines as diverse as sociology, anthropology, psychology, geography, mathematics, statistics, and computer science [46]. There is one standard textbook on social network analysis, [69], cited by the large majority of papers in this field. Although social network analysis has existed for over fifty years, most analysis techniques have been designed for static data. For example, [69] contains no mention of temporal metrics, even though it was written in 1994 when electronic networks were well established. It is difficult to collect social data for numerous individuals by hand using survey techniques. However, with the increase in the use of computers, collecting enough data to create numerous graphs over fixed time intervals becomes possible. An example is creating a graph per week from email data, using a server's email log of to, from, and date fields [10]. This series of graphs can be used to study the evolution of the network and the change over time of various metrics.

2.4.1. Equivalence

Equivalence is a way of comparing the similarity of two individuals based on the similarity of their

position in a graph [30]. There are three types of equivalence: structural, automorphic and regular. If two nodes are structurally equivalent they are equivalent in location, identical or substitutable. They have exactly the same links to exactly the same nodes. This is a very rare situation in real life as no two people are identical. In the graph shown below the orange nodes are structurally equivalent.

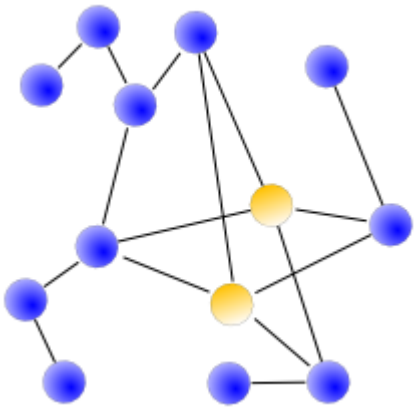

Figure 7. Structural equivalence

If two nodes are regularly equivalent they are equivalent in role. They have the same links to nodes that are also regularly equivalent. This is a recursive definition that makes finding regularly equivalent nodes in a graph a non-trivial problem. However, understanding the concept is far easier. Two doctors in two different hospitals are not structurally equivalent because they know totally different people. But they are regularly equivalent as they have similar ties to nurses, administrators and patients who are also regularly equivalent. Mathematically, if x_i and x_j are equivalent, and x_i is adjacent to x_a, then x_j must be adjacent to x_b, where x_b and x_a are equivalent. In the graph shown below the orange nodes are regularly equivalent.

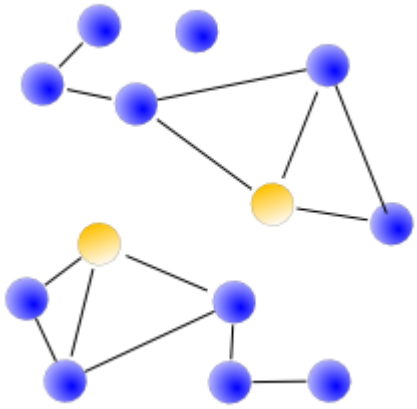

Figure 8. Regular equivalence

Automorphic equivalence is not as strict as structural equivalence but more strict than regular equivalence. It is concerned with finding subgraphs of individuals that can be moved in the network to replace a similar module somewhere else. Two nodes in such a substitutable module are automorphically equivalent.

2.4.2. Web link analysis

Since websites form networks, tools that have been used to analyse the structure of the Web are also useful for analysing the relationships between people. Most of these tools have been developed by search engine firms. In this field, sites with a high in-degree are called authorities, as other sites trust them to be the authoritative expert on a certain topic. Sites with a high out-degree are called hubs, as they are like telephone directories that reference a large number of the most authoritative experts on a certain topic. Two algorithms used to discover authorities are HITS and PageRank (used by Google) [50]. The importance of hubs and authorities is calculated by looking at the importance of sites to which they link. Thus the definition is recursive and the ranking is not trivial to compute.

2.5. Metrics

A metric is a value calculated from a graph that describes the graph in some way. This section provides only an overview of metrics that are used by social network analysis researchers. Their precise mathematical definitions are given in chapter three. Many metrics have been defined, only some of which are useful for link prediction. Out of that subset, only some have been used in this research. Most traditional social network analysis metrics are described and defined in [69] and summarised in an online book by Hanneman [30]. New metrics are invented occasionally by researchers who wish to concentrate on a particular area of social network analysis.

Two examples of metrics are *degree* and *common neighbours* [69]. These are monadic and dyadic metrics respectively [69]. In other words, degree is a value calculated for a single node and common neighbours is a value calculated for a pair of nodes (a dyad). The degree of a node is the number of connections to other nodes that it has. It therefore represents the popularity or sociability of a node. The number of common neighbours of a dyad is the number of mutual nodes a dyad shares. It therefore represents how much overlap two nodes' social circles have. Both these metrics are neighbour-based metrics, because they are based on calculations involving the number of neighbours of a node. Other neighbour-based metrics include Jaccard's coefficient, the Adamic\Adar number and preferential attachment [35][1]. Another commonly used type of metric is the distance-based metric. This involves calculating the shortest path (or many shortest paths) between two nodes. Variations of

the shortest path metric include the Katz measure (where paths of different lengths are weighted differently) [41] and betweenness [69]. Betweenness is a type of centrality metric. These metrics are an indication of how popular or prestigious a node is, i.e. how many paths between other nodes pass through it, or how close it is to other nodes. It is one of the most commonly used social network analysis tools.

2.6. Link prediction

The nodes in a sociogram are linked in a complex web of relationships that change over time. These relationships emerge, strengthen and decay as a result of individuals' positions in the network, their behaviour and the influence of the environment. Predicting changes to a social network is called the link prediction problem. Liben-Nowell and Kleinberg [42] explain it as:

> Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t'.

The problem is illustrated in the graphs below, where a link is forming between the orange nodes.

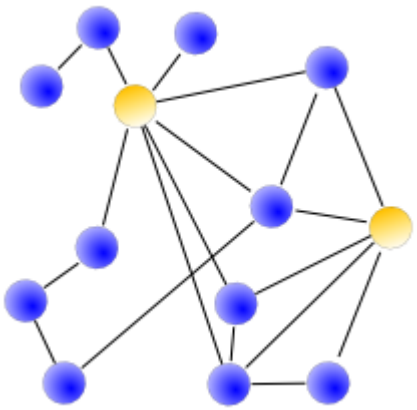

Figure 9. A graph at time step t where a new link is forming

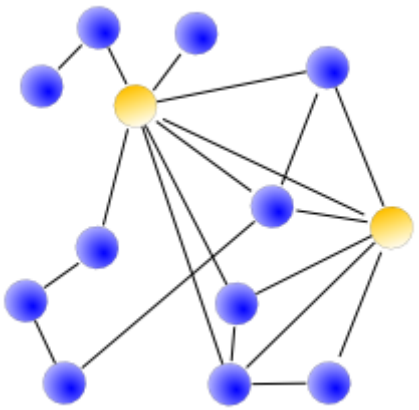

Figure 10. A graph at time step t+1 where a new link has formed

Link prediction is generally approached in two separate (but complementary) ways: relational analysis (or topological analysis) and feature analysis. The first way examines a sociogram for unbalanced social structures that should tend towards a state of equilibrium (e.g. two people who have a lot of mutual friends should eventually meet). The second way does not involve graph theory at all, but rather looks at the content of communications between individuals to search for common interests (e.g. two people who discuss both fly-fishing and abstract algebra in emails should eventually meet). This dissertation concentrates only on the former method – predicting links using graph theory – and disregards the content of communications. This method was chosen so as to concentrate on only one

part of the problem and perform detailed research on it alone. The purely graph theoretical technique will almost certainly not be as accurate as one that uses both graph theory and content analysis combined, but it means that it is suitable for analysing networks where no content is available (such as email logs, which are numerous and easily accessible to intelligence and business analysts). Additionally, a purely graph theoretical approach to link prediction can later be augmented with content analysis.

2.6.1. Existing link prediction techniques

Existing link prediction techniques can use the values of metrics in a graph instance to determine where new links are likely to arise. For instance, it is more likely that a new link will be incident to a node with a high degree than a node with a low degree. However, though link prediction has been used for many applications, there have been very few investigations of link prediction itself. This section lists papers that discuss link prediction per se, and the next section lists papers that discuss its use in an application. In 2003 Popescul and Ungar made citation prediction systems using statistical learning that extended inductive logic programming [54]. Their system learnt link prediction patterns from queries to a relational database, including joins, selections and aggregations. Taskar, Abbeel and Koller used relational Markov models to learn patterns of cliques and transitivity in web pages and hyperlinks [64]. Both these prediction systems included node attributes (e.g. web page text) in addition to relational features. This makes them more powerful than prediction systems using only topological metrics, but also more domain specific. In 2004 Popescul and Ungar enhanced link prediction of author\document bipartite networks by using clustering [55]. They clustered documents by topic and authors by research community in order to generate new entities that were used in logistic regression of features and relations. Their system was tested on data consisting of an equal number of positive and negative cases. They managed to increase accuracy over models not using clustering by roughly four percent on average. Zhou and Scholkopf approached three related graph problems (classification, ranking and link prediction) in a new way [72]. They defined discrete calculus for graphs and then shifted classical regularization from the continuous case to graph data. Though mathematically interesting, their paper did not include empirical testing.

Liben-Nowell and Kleinberg tested the predictive power of only proximity metrics, including common neighbours, the Katz measure and variants of PageRank [42]. They found some of these measures had a predictive accuracy of up to 16% (compared to a random prediction's accuracy of less than a percent). A third of Liben-Nowell's doctoral thesis was a chapter on link prediction in social networks [41]. It supersedes his 2003 paper [42]. His hypothesis was that link prediction could be performed from topology alone. This was found to be true since his system outperformed random

predictions by a factor of 50 at times. However, the collaboration networks he extracted from www.arxiv.org, the online physics paper archive, were quite small (between 486 and 1790 nodes only). As such, he would have been able to store all his networks in an adjacency matrix, making shortest path calculations and similar computationally expensive metrics quick to calculate. Medium size social networks consisting of fifty thousand nodes or so cannot be stored in adjacency matrices because of random access memory size limitations. This is because an adjacency matrix has to have an entry for all pairs of nodes, making the total memory required quadratically proportional to the number of nodes in the network. Thus his shortest path based prediction methods are somewhat impractical for large-scale analysis. Luckily, he found that common neighbour methods "perform surprisingly well". Unfortunately by definition common neighbour approaches cannot predict links between nodes at a distance greater than three (since nodes with a common neighbour have a distance of at most two). This is called "the distance-three task" [41]. This difficulty might be partially alleviated by comparing shared topological features. For instance, by the principles of assortative mixing and homophily two nodes of unusually high degree are more likely to form a link, despite being at a distance of more than three.

Despite the computational obstacles involved with large networks, they might in fact be more suited to link prediction. Liben-Nowell notes that the more diverse a network is, the easier it is to separate nodes into groups of common interest (e.g. research interests in publication networks). With smaller graphs, people are likely to form links more randomly, as all the nodes tend to be similar. This brings up an interesting point about the dating network data used in this research: if we assume a link forms mainly between a male and a female then prediction could be improved vastly by separating all the nodes into a male group and a female group. However, this grouping would involve finding the regular equivalence [30] between nodes, a procedure that is computationally expensive and hence unusable for large graphs. It requires a complex algorithm since finding regular equivalence involves finding nodes that have similar links with other nodes that in turn have similar links to the original nodes and other nodes. Grouping the data set into males and females would need to use this type of approach. Finding the groups of women in the network would involve finding nodes (women) that have similar links with other nodes (men) that in turn have similar links the original nodes (women) and other nodes (both men and women).

A few other link prediction papers are summarised in Getoor and Diehl's 2005 survey of link analysis [25].

2.6.2. Link prediction application papers

Huang, Li and Chen investigated the use of link prediction to improve collaborative filtering in recommender systems [35]. They found that the Katz measure was the most useful, followed by preferential attachment, common neighbours and the Adamic\Adar measure. These path-based and neighbour-based measures outperformed simpler metrics. They found that the distance between nodes was not useful, probably because most node pairs in their set could be linked within a short path. Farrell, Campbell and Myagmar used link prediction to design a system that recommended new academic links for researchers at a computer science conference and received feedback through a survey [20]. They found that established researchers found little use for the system but newer researchers found it useful in recommending potential colleagues and talks of interest. Farrell et al are proponents of relation-oriented computing and believe that social network systems will be useful to help humans cope with the huge number of professional contacts they need to maintain at present. Their systems maintain information on contacts and their links by analysing data in bimodal graphs (such as papers published and academics, or meetings attended and businessmen). In his doctoral thesis Zhu [73] used link prediction to determine what web page a user was next likely to visit in order to improve the navigation and efficiency of a site. This was done by storing lists of web pages visited as a Markov chain. Normally when predicting links in social networks we assume links are independent. In other words, a person forms new links one at a time, and not in a common sequence of new people. This is clearly different from how Zhu predicts visits to web pages. This technique would be useful only if we could assume that forming a new link to a person A would imply that it was probable the next link formed would be to person B, which is not how links form in social networks.

2.6.3. Link completion

Link completion is also a link analysis problem and is almost identical to link prediction [28]. It differs in that the problem is to determine the missing node in a pair of nodes defining a link. In other words, if we have a data set with some partial links we need to determine to which node another particular node links. Thus link completion is subsumed in the harder and more general problem of link prediction; instead of trying to determine which pair of nodes are most likely to link next, we are trying to determine only to which node another node is trying to link. The node is already known to have a link, we are just not sure to what other node the link is attached. An example of the problem is when a user buys five books online and the name of one book is corrupted in transfer. A link completion algorithm could infer the name of the missing book based on the user's name and the other books she bought.

2.7. Anomalous link discovery

In 2005 Rattigan and Jensen [58] presented the anomalous link discovery problem in response to the massive difficulties of link prediction. They reasoned that link prediction has inherent insurmountable difficulties because the number of dyads that need to be evaluated increases quadratically in proportion to the number of nodes in a network and social networks are very sparse, leading to extremely few positive cases. This makes it nearly impossible for prediction systems to learn significant differences between metrics in positive and negative cases. There are so few positive cases that they are "swallowed up" by the negative cases that have similar metrics. Rattigan and Jensen therefore recommend focussing on anomalous link discovery. This involves detecting which links in a network are surprising - e.g. two linked nodes that had very few common neighbours or were a large distance apart in the previous time step. Examples where anomalous link discovery could be used include discovering surprising links arising between individuals that could indicate criminal collaboration or discovering surprising links between web pages that indicate their content is somehow related. Anomalous link discovery is complementary to link prediction in the sense that they both use the same metrics to evaluate which links are surprising and which are expected. Thus research on improving either problem should benefit the other.

2.8. Link detection

Link detection is similar to link prediction, but involves determining where an established link exists that is not shown in the sociogram, perhaps because of missing data. An example is inferring the influence (a type of hidden link) of one researcher on another, based on publication networks. Often criminal analysts have only partial network information to analyse, obtained through phone records, email records and the observations of human intelligence gathering agents. Criminals would like all their links to others not to be known and might intentionally hide their links as much as possible. Similarly, marketers undertaking surveys for use in activities such as viral marketing might wish to infer connections between people based on incomplete information. Link detection would be useful in all these circumstances. Liben-Nowell [41] and Taskar et al [63] mention the problem as an application of link prediction in the introduction to their research. Komarek used the link detection problem as a test application for his work on fast logistic regression in his 2004 thesis [39]. Similarly, Popescul and Ungar used the link detection problem as a test application for their work on structural logistic regression [54]. Link detection in social networks using topological metrics must not be confused with other types of link detection. These include story link detection [15], where news articles are compared to determine if they are about the same event, and hardware link detection, which is involves determining the state of a computer's network card's link to other computers.

2.9. Dynamic network analysis

The analysis of the changes in social networks over time is called dynamic network analysis. It is currently a popular avenue of research for law enforcement and intelligence agencies, given the rise in the global activities of terrorists and other organised criminal groups [16]. Such groups have been labelled dark networks, and their structure and behaviour differs widely from normal social networks. For example, they trade efficiency for secrecy in structure and have unusual patterns of communication [21]. Carley is one of the most prolific researchers in the modelling of dark networks using dynamic techniques. She has created a dynamic network program, *DyNet*, where multiple agents model the social behaviour of human beings, with access to resources and organisations [12]. This program is used to understand network evolution and the best way to destabilise terrorist networks. These techniques are powerful, but relatively domain specific and complex. There have also been a few purely theoretical studies done on the change of the structure of networks over time. Holme's work has focused on this, including studies on the changing metrics of a Swedish Internet dating network called Pussokram [32][33]. Differing from this research, Holme's work investigated the trends of aggregate graph measures, such as the average path length and average degree.

2.9.1. Temporal analysis

Leskovec, Kleinberg and Faloutsos [40] state that little work has been done on analysing long-term graph trends:

> Many studies have discovered patterns in static graphs, identifying properties in a single snapshot of a large network, or in a very small number of snapshots; these include heavy tails for in- and out-degree distributions, communities, small-world phenomena, and others. However, given the lack of information about network evolution over long periods, it has been hard to convert these findings into statements about trends over time.

Their study of trends found that over time graphs increase in density and the average distance between nodes decreases. This was contrary to the existing beliefs that average nodal degree remains constant and average distance slowly increases. They claimed that existing graph generation models are not realistic and proposed a new “forest-fire” generation model. Desikan and Srivastava studied the change in metrics of a set of web pages over time for the graph as a whole and for single nodes (subgraphs are their current research) [19]. They found that temporal metrics, such as their Page Usage Popularity, “can be effectively used to boost ranks of recently popular pages to those that are more obsolete.” This seems to indicate that temporal metrics are a useful addition to traditional static metrics in the study of some networks.

2.10. Applications of social network analysis

Social network analysis has many real world applications, especially in the fields of marketing and crime prevention. This section discusses some of these applications.

2.10.1. Search in social networks

Adamic, Lukose and Huberman investigated searching in scale-free networks, such as a peer-to-peer system or social network [2]. They assumed that searching a node meant that all its neighbours had been included in the search. This idea is similar to how the participants in Milgram's experiment tried to pass a message through a social network, knowing only their immediate neighbours. It was found that a random-walk traversal of a scale-free network will cover a large fraction of nodes of the network since an edge is more likely to lead the random walker to a node of high degree, and such nodes are linked to many other nodes in a network. Thus, when searching for a particular node (an expert, or a computer with a certain file), the search request message can be passed from node to node, rather than broadcast to an entire network, and still quickly find the required node in most cases.

2.10.2. Dark networks

A dark network is one that operates covertly and illegally. Examples include suppressed political organisations in countries run by dictators, terrorists, smugglers and drug trafficking organisations. A bright network is one that operates overtly and legally. Examples include the police, military, government and most large companies. Raab and Milward found in their analysis of many instances of dark networks that the topologies of dark networks are as varied as those of bright networks. Though the networks are each faced with similar problems of operating covertly, they cope in diverse ways [57]. Government intelligence analysts use combinations of link and group prediction, past criminal activities, potential sites of risk and visualisation techniques to predict the details of future criminal activities [53].

2.10.3. Content recommendation systems

Huang et al investigated the use of link prediction in collaborative filtering recommender systems [35]. An example of a recommender system would be Amazon.com, which recommends books similar to the one someone is buying, which other buyers of the book have enjoyed. Amazon.com is also an example of collaborative filtering, because by their purchases users are collaboratively changing the preference rankings of books in the system. Collaborative filtering first clusters users into similar groups based on their preferences and then recommends to a user items preferred by the user's neighbours. However, the system suffers from two problems. The bootstrapping problem is

that nothing can be recommended when the system is first used as no one has chosen any items. The sparse data problem is that few items can be recommended when only a few users have chosen a few items. This is addressed by Huang by performing link prediction on the graph to predict which items users should choose and thereby increase the density of the graph. Golbeck has also investigated content recommendation [26]. Her system used the trust users assign to each other to enhance film recommendations. She has also used social networks to improve mail sorting and classification, including collaborative spam filtering [27].

2.10.4. Marketing

Companies from Amazon to Yahoo are trying to figure out ways of advertising to customers that include references from a trusted source. For instance, an advertisement would be displayed on a web page that endorses a local restaurant using a review written by someone close by in the reader's social network. This application is rapidly gaining importance as analysts believe advertising will be worth about 11 billion dollars in 2009 [44].

2.10.5. Ecology

McMahon, Miller and Drake recommend that biological ecologists and social scientists could use each others' tools to better understand their own discipline [46]. They suggest that ecologists could use the concepts of betweenness and distance to better understand the interaction of species and that regular equivalence is useful for understanding relationships between groups of predator and prey. And in return, social network analysis researchers can use molecule visualisation techniques to visualise social networks and can compare the complexity of social networks using the biological concept of “connectance”.

2.10.6. Specific applications of link prediction

The capability to predict changes in relationships before they occur is highly beneficial to an organisation. Examples of the advantages of such social clairvoyance include:

- Identifying the structure of a criminal network (i.e. predicting missing links in a criminal network using incomplete data) [13].
- Overcoming the data-sparsity problem in recommender systems using collaborative filtering [35].
- Accelerating a mutually beneficial professional or academic connection that would have taken longer to form serendipitously [20].
- Improving hypertext analysis for information retrieval and search engines [31].

- Monitoring and controlling computer viruses that use email as a vector [43].
- Predicting which web pages users will next visit in order to improve the efficiency and effectiveness of a site's navigation [73].
- Helping to predict the spread of an entity through a network [71]. Examples include a disease, such as HIV, or information, such as a clothing fashion or rumour.

Link prediction might also be useful in ecology, though interdisciplinary sharing between these two fields is still new [46].

2.11. Social data sources

Most social network analysis is focused on analysing a single graph – static analysis. This is because of the static nature of the data most researchers have used. Traditionally, social network data is collected by sociological surveys, analysing historical archives or other time consuming processes that lead to the creation of only one sociogram. However, the proliferation of the Internet has made collecting far larger data sets, that provide many sociograms over time, an easy task. Examples include search engine records of websites, social networking Internet communities (such as *hi5*, *LinkedIn* and *Orkut*) and email logs. These temporal sequences of sociograms provide a richer description of the underlying network – showing both structure and behaviour. However, because of the novelty of temporal networks, there has been almost no adaptation of static techniques to techniques more appropriate for them. Additionally, the massive increase in volume of data on which to perform calculations that is gleaned from sequences of sociograms, as contrasted to a single sociogram, has implications for the practicality of computational analysis. Whereas previously researchers could calculate all possible metrics for every node and subgraph in a single sociogram, now using many sociograms the same detailed analysis would take unacceptably long (many weeks or months). Thus, for the purposes of link prediction, we need to find metrics that are very quick to calculate and that also provide useful information for training whatever prediction system is being used.

2.12. Computational complexity of social network analysis

Perhaps the biggest problem when trying to understand a sociogram is the amount of time it takes to calculate metrics for a graph, and the amount of space it takes to store them. Let us consider two examples, one requiring a large amount of time to calculate and one requiring a large amount space to store. For the first example, a commonly used monadic metric is betweenness, the percentage of shortest paths between all nodes that contain a specified node. The calculation of this metric requires the discovery of every shortest path between every pair of nodes in the graph. The time taken to

compute these paths increases exponentially as the number of nodes in the graph increases (i.e. over each successive time step, as more people join a network). The storage of this metric is less intensive, requiring one value to be stored per node. In the second instance, consider the task of storing that a link exists between two nodes. This is normally represented by an edge table for sparse graphs (where each vertex has a list of vertices to which it is linked), or an adjacency matrix for dense graphs (where a one at $m_{i,j}$ in the matrix implies node i and node j are linked). Since nearly all large social networks are sparse by nature, an edge table is an efficient storage mechanism. An adjacency matrix is not efficient as it increases quadratically in size in proportion to the number of nodes it contains. However, an edge table is not suitable for many statistical calculations when performing link prediction. A large number of network computations that are simple to implement using matrix operations (such as multiplication to find paths of certain lengths) are far more complex to implement using an edge table. Furthermore, storing dyadic metrics that we have calculated requires space quadratically proportional to the number of nodes in the graph. If there are n nodes in a graph we can calculate dyadic metrics for up to n^2 of them. This ultimately means that link prediction using most artificial intelligence techniques requires a huge amount of database space. Ulrik Brandes vastly increased the computational speed of centrality calculations from $O(n^3)$ to O(nm), using a new algorithm[2] in 2001 [9]. He invented a method of recursively computing shortest paths for all nodes simultaneously. This increase in computational speed is a certainly a leap forward. Unfortunately, even using this algorithm, the sheer number of nodes and edges in a large graph means that calculating several metrics using a single processor can take a few hours. Carpenter, Karakostas and Shallcross note that computing betweenness for an undirected unweighted 6000 node graph takes about fifteen minutes [13].

2.13. Decentralisation (distributed intelligence) theory

A dominant theme in the artificial intelligence community currently is distributed intelligence, or the decentralisation of decision making capabilities. Resnick describes how decentralisation is the only way to understand certain complex systems, such as: ant colonies, flocks of birds, traffic, economic markets, evolution and immune systems [60]. He also notes that society is currently increasingly adopting the decentralisation paradigm in diverse ways, and especially in the computer modelling of systems. These ways include democracy, free markets, flat company management hierarchies and self contained business units, the Internet, object orientated programming, evolutionary programming and agent-based design.

A decentralised approach to model any group of objects is usually one in which there is no leader

2 Where n is the number of nodes, and m is the number of edges.

object (or central planner and commander), and where every object in the model has some degree of simple intelligence and contributes to the operation of the model, which emerges from the interactions of all the objects. The advantages of this approach over a traditional centralised approach often are:

- that the model is more realistic, i.e. a better approximation to real behaviour, as objects in reality usually have information only on their immediate surroundings,
- that the algorithm is faster, due to performing a larger number of simpler operations,
- that the algorithm is easier to design, due to the simple nature of the individual objects' operations, but harder to tune for optimum performance, due to the erratic nature of emergent behaviour.

2.14. Dealing with complexity in sociograms

Carpenter et al [13] proposed several ways to overcome the prohibitive computational times discussed in the previous sections. They suggest that graphs should be separated into components so that separate shortest paths can be calculated for each component and then recombined into longer shortest paths for the graph as a whole. Alternatively, when working with temporal graphs, computations can be done that alter existing metrics only by noting the changes that new vertices and edges in the latest time step induce (as opposed to recalculating the metrics from scratch at every time step). They also suggest that it may be more useful to use metrics that calculate localised information from a graph (i.e. from a small neighbourhood of nodes, as opposed to the entire graph). This approach seems intuitively sound as people interact usually only with their immediate social group, or one social group removed. Also this approach fits well into the emergence paradigm in complex adaptive systems, where localised interactions combine to create sophisticated global patterns [23].

Kempe and McSherry investigated a decentralised distributed algorithm for spectral analysis in graphs [37]. This algorithm treats each node as a computational entity – i.e. assuming hundreds of thousands of multiprocessors (such as computers on the Internet). Thus it cannot be used for social network analysis performed using a single processor, or even a dozen processors, due to the massive multiprocessor requirement. Their paper was related to recent work that tries to infer global properties from local analysis, such as Benjamini and Lovasz, where certain global topological properties of a graph were determined from random walks [8]. Kleinberg noted that it is surprising that people in Milgram's experiment were able to construct short paths to the target based only on local information [38]. He proposed a decentralised algorithm to find short paths based only on local knowledge but found it works only in graphs with certain properties. These techniques are not directly applicable to link prediction but show that local analysis can be a useful tool for analysing graphs. In an unpublished draft Pattison and Robins propose neighbourhood-based models for

understanding social networks [52]. They consider Markovian neighbourhoods and variations as a way to understand the structure of social networks. A pair of possible ties belong to a Markovian neighbourhood (and so are conditionally dependent) whenever they have a node in common. This is a rare instance of research into local neighbourhoods in the field of social network analysis. Though their work is unrelated to link prediction and computational problems it could prove useful if applied.

2.15. Conclusions

This chapter presented a broad overview of social network analysis. It introduced the concept of metrics, which are used by researchers to obtain useful sociological information about individuals, dyads and groups of people. The problem of link prediction and various solutions were described. Finally, some of the inherent challenges of link prediction and social network analysis were presented. These challenges and the deficit in temporal research discussed in this chapter are addressed in the rest of this dissertation. The next chapter describes the previous statistical and research methodologies that have been used by other researchers, the methodology used in this research and the software system used to conduct the research.

Chapter 3. Research methodology

This chapter explains the research methodology common to the experiments presented in the following four chapters. It presents an overview of the statistical analysis and data mining tools used and describes the Java software system created to analyse the data and calculate variations of metrics defined by past researchers. The first part of the chapter reviews common practice in the field of social network analysis and statistics. The discussion is structured in the same way data flows through a program: starting with raw data, proceeding to metric calculations, input transformations and computations, and ending with statistical analysis techniques. This first half of the chapter compares various methodological options. The second part of the chapter builds on this review to describe the exact methodology used in this research. Also included is an overview of the software system designed to store and analyse sequences of social networks.

3.1. Data format

Social network data is simple to store and understand. At the most basic level all that is needed is the names of the nodes involved in a link or message. This research also made use of temporal information, which was obtained from the date on which the message was sent. An example of a social network data set is shown below. It was taken from the Pussokram data set message file:

```
34215;8936;2001-2-13 13:54:00
34215;8936;2001-2-13 14:01:00
123154;34215;2001-2-13 16:58:00
34215;42183;2001-2-13 17:07:00
8560;42172;2001-2-13 17:35:00
```

The columns of the data set are separated by semicolons and the rows are separated by line breaks. The first column names the node that sent the message. The second column names the node that received the message. The last column specifies the date and time the message was sent. This data was used by Holme in [32] and [33]. Another source of social network data is the log files created by email servers. An example extract from a logfile created by the sendmail program is shown below. Fictional addresses have been used to protect privacy.

```
Dec 30 00:00:00 mail newsyslog[30554]: logfile turned over

Dec 30 00:01:57 mail sendmail[30563]: jBTM1qA30563:
from=<kaauqeffjtnfy@yahoo.com>, size=20548, class=0, nrcpts=1,
msgid=<200512292201.jBTM1qA30563@mail.audiobiz.co.za>, proto=SMTP, daemon=MTA,
relay=213-235-66-135.web.star.ps [223.235.66.135]

Dec 30 00:02:01 mail sendmail[30565]: jBTM1qA30563:
to=<lindy@jamesonediting.co.za>, delay=00:00:08, xdelay=00:00:00,
```

```
mailer=local, pri=50231, relay=local, dsn=2.0.0, stat=Sent

Dec 30 00:05:21 mail sendmail[30577]: jBTM5LA30577:
<jerry@gooseelectric.co.za>... User unknown

Dec 30 00:05:21 mail sendmail[30577]: jBTM5LA30577: lost input channel from
server.garten.com [88.51.253.17] to MTA after rcpt

Dec 30 00:05:21 mail sendmail[30577]: jBTM5LA30577: from=<>, size=0, class=0,
nrcpts=0, proto=SMTP, daemon=MTA, relay=server.garten.com [86.70.250.16]
```

The extract above illustrates some of the challenges in using logs to generate social data. Firstly, the receiving and forwarding of an email by a server are two separate actions completed at different times. Thus we need to match every email that contains "from=<", with emails that contain "to=<", where all the emails have the same unique message identifier. Furthermore, bad information as shown in the extract, such as empty addresses, unknown users, lost input channels and other errors, must be discarded.

3.2. Metric calculation

The social network data described in the previous section forms the base on which researchers perform metric calculations. These metrics are typically stored in csv files, where commas separate attributes and rows separate instances. An instance is a set of related metric values (usually a set of metrics relating to one or a pair of nodes). An attribute is usually a metric. The words instance and attribute are general terms that are used in machine learning [70]. There is social network analysis software available on the Internet that will perform standard metric calculations. Researchers that are testing new metrics have to program their own metric computations. These metrics are in turn statistically analysed for patterns. In the following subsections the precise definitions of metrics are given.

3.2.1. Basic terminology

Before the metrics themselves can be defined, a few basic graph theory definitions must be given. The following list of mathematical definitions used in this research provide the basic elements that will be used to define metrics in the next section. The time step, written in subscript, of a set of nodes, links or other elements can be omitted if it is implicit or irrelevant. For the sake of brevity most of the definitions and metrics are defined only for bidirected graphs (i.e. using the set U_n). However, many can trivially be changed to apply to directed graphs (i.e. using the set E_n). In fact, metrics can generally be defined in four ways:

- for bidirected links,
- for links leaving a node,

- for links coming into a node,
- and for links either leaving a node or coming into it (in which case a bidirected link would be counted twice).

Note that the first and last type of links are not the same. For instance, the concept of degree (i.e. bidirected degree) can be expanded to in-degree, out-degree and inout-degree for each of the four ways given above, respectively. If a node has two bidirected links and one non-reciprocated outgoing link its bidirected degree would be two, its out-degree would be three, its in-degree would be two and its inout-degree would be five. This is illustrated for the orange node in the graph below.

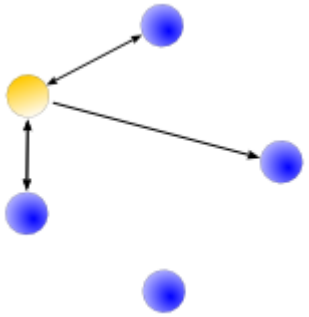

Figure 11. A node with two bidirected links and an outgoing directed link

Metric definitions that can be used in either of these four ways include strength, recency, distance and the set of all shortest paths. This applies to many of the metrics defined in tables 2, 3 and 4, as well as strength and recency, defined in table 1. Nodes can link to themselves. Finally, nodes are never removed over time, and links never decrease in strength. This was a simplifying assumption adopted in this research.

Table 1. Basic mathematical definitions used in metric definitions

Name	Description
$G_n=\langle V, E\rangle$	A graph of nodes and links at a given time step n (a point in time).
V_n	The set of nodes of the social network at time step n. A node in this set is notated as v_i. V_n contains all nodes from previous time steps (e.g. $v_i \in V_2 \Rightarrow v_i \in V_{49}$).
E_n	The set of links of the social network at time step n. A link between nodes v_i and v_j is notated as $e_{i,j}$. Note that this link is directed and is a link from v_i to v_j. E_n contains all links from previous time steps.
U_n	The set of bidirected links of the social network at time step n. A bidirected link between nodes v_i and v_j is notated as $u_{i,j}$ and $u_{j,i}$. Note that this link exists if and only if $e_{i,j} \in E_n$ and $e_{j,i} \in E_n$. U_n contains all bidirected links from previous time steps. Any node contained in a link in a given time step will also be contained in the set of nodes for that time step.
#	The number of elements in a set. E.g. $\#(V_n)=6$, if the set V_n contains six nodes.
max	The maximum element in a set. E.g. $max(\{2,4,1,3\})=4$.
min	The minimum element in a set. E.g. $min(\{2,4,1,3\})=1$.
mean	The mean of a set. E.g. $mean(\{2,4,1,3\})=\frac{10}{4}=2.5$.
$f_{m,n}(v_i)$	Given a function f, $f_{m,n}(v_i)$ is the sequential set of all values of $f_t(v_i)$ over the time step variable t, ranging from m to n. E.g. $degree_{1,365}(v_3)=\{degree_1(v_3), degree_2(v_3), \ldots, degree_{364}(v_3), degree_{365}(v_3)\}$.
$\Gamma(v_i)$	The set of neighbours of v_i ,i.e. the set $\{v_j : u_{i,j} \in U\}$.
$str(e_{i,j})$	The strength of a link from node i to node j. Strength is defined as the number of messages sent between the two nodes.
$rec(e_{i,j})$	The recency of a link from node i to node j. Recency is defined as the number of time steps that have elapsed since the last message was sent or received. If a message was sent in the last time step the recency of the link would be 1.
$rec(v_i)$	The recency of node i. Recency is defined as the number of time steps that have elapsed since the last message was sent to or received by any other node. If a message was sent in the last time step the recency of the node would be 1.
$dist(v_i, v_j)$	The distance between nodes v_i and v_j. In other words the path length (number of links) in the shortest path between v_i and v_j. If v_j is unreachable from v_i then $dist(v_i, v_j)=0$. The distance from a node to itself is undefined.

Name	Description
$P(v_i, v_j)$	The set of all shortest paths from v_i to v_j. The elements of $P(v_i, v_j)$ are sequences of nodes, beginning at v_i and proceeding along the path to v_j. If $p \in P(v_i, v_j)$ then the distance from v_i to v_j is $\#(p) = dist(v_i, v_j) + 1$.
$P(v_i, v_j, v_x)$	The set of all shortest paths from v_i to v_j that pass through node v_x.

3.2.2. Metric definitions

Metrics can be generally be separated into three categories:

- Monadic metrics: calculated for a single node, e.g. degree.
- Dyadic metrics: calculated for a pair of nodes, e.g. number of common neighbours.
- Graph metrics: calculated for an entire graph, e.g. graph size.

These are the categories into which the metrics listed in the tables below have been separated. Metrics can also be calculated on groups of nodes. These metrics are simply graph metrics as computed on a given subgraph and do not warrant their own category. Note that the word "mean" in the name of a metric does not imply that it is a temporal metric, but rather that it is a metric for that time step as a whole. For instance, *mean number of messages received* is the mean calculated over all nodes in the graph in one time step, not the average number of messages received of a given node over several time steps. Some of the metrics listed below are complex to calculate and their algorithm is not given if they are not used in this research. The interested reader can refer to the reference given after the metric name for a complete discussion. The monadic metrics below are defined for the node of focus v_i at time step n and the dyadic metrics are defined for the pair of nodes v_i and v_j.

Table 2. Monadic metric definitions

Name	Definition	Description
Degree	$\#(\{u_{i,j}: u_{i,j} \in U_n\})$ or $\#(\Gamma(v_i))$	The number of links from v_i to any node at time step n.
Normalised degree [69]	$\frac{\#(\Gamma(v_i))}{\#(V)-1}$	A standardised degree that ranges from 0 to 1. A normalised metric is one where the metric definition has been altered to ensure that the values fall within a standard range.
Recency	Defined in terminology section	The number of time steps elapsed since the node last communicated.
Eccentricity [30]	$max(\{dist(v_i, v_j): v_j \in V\})$	The length of the shortest path to the node furthest away.
Jordan centrality [69]	$\begin{cases} 1 & if\ eccentricity(v_i) = min(\{eccentricity(v_j): v_j \in V_n\}) \\ 0 & otherwise \end{cases}$	Has a value of 1 if v_i is a central node (node of minimum eccentricity) in the graph, otherwise 0.
Closeness [69]	$\frac{1}{\sum_{v_j \in V_n, v_j \neq v_i} dist(v_i, v_j)}$	How close the node is to all other nodes. This ranges from 0 (far away) to $\frac{1}{\#(V)-1}$ (very central).
Betweenness [69]	$\sum_{v_j \in V} \sum_{v_k \in V, v_k \neq v_j} \frac{\#(P(v_j, v_k, v_i))}{\#(P(v_j, v_k))}$	The sum of all shortest paths between all nodes that contain v_i as a percentage of all shortest paths between all nodes. It ranges from 0 to $\frac{(\#(V)-1)(\#(V)-2)}{2}$.

Name	Definition	Description
PageRank [56]	The fixed point of the following recursive definition: $PageRank(v_i) = \sum_{v_j \in V, u_{i,j} \in U} \frac{PageRank(v_j)}{\#(\{u_{j,k} : u_{j,k} \in U\})}$	This is a very basic definition of the web page ranking algorithm used by Google. Intuitively, a node has a high importance if other nodes of high importance link to it.

Table 3. Dyadic metric definitions

Name	Definition	Description
Distance	Defined in table 1	The distance between the two nodes.
Link strength	Defined in table 1	The strength (number of messages sent) of the link between the two nodes.
Recency	Defined in table 1	The number of time steps elapsed since the link was last used.
Common neighbours [35]	$\#(\{v_k : u_{i,k} \in U_n, u_{k,j} \in U_n\})$ or $\#(\{\Gamma(v_i) \cap \Gamma(v_j)\})$	The number of nodes linked to both focus nodes (i.e. mutual friends).
Jaccard's coefficient [35]	$\frac{\#(\{\Gamma(v_i) \cap \Gamma(v_j)\})}{\#(\{\Gamma(v_i) \cup \Gamma(v_j)\})}$	The number of neighbours of the focus nodes divided by the number of nodes that are neighbours of either focus node.
Adamic\ Adar similarity [1]	General case: $\sum_{z : a\ shared\ feature} \frac{1}{\log(frequency(z))}$ Common neighbours case: $\sum_{v_z \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{1}{\log \#(\{\Gamma(v_z)\})}$	The number of features shared by the nodes, divided by the log of the frequency of the features. This metric rates rarer features more heavily.
Preferential attachment [35]	$\#(\{v_k : u_{i,k} \in U_n\}) \cdot \#(\{v_k : u_{j,k} \in U_n\})$ or $\#(\{\Gamma(v_i)\}) \cdot \#(\{\Gamma(v_j)\})$	The product of the number of edges incident to the two nodes.
Katz measure [41]	$\sum_{l=1}^{\infty} \beta^l \cdot \#(\{paths_{v_i, v_j}^{\langle l \rangle}\})$, where $paths_{v_i, v_j}^{\langle l \rangle}$ is the set of all paths of length l from v_i to v_j	The sum of all paths between the nodes exponentially damped by length to weight short paths more heavily. $0 < \beta < 1$.
Hitting time [41]	The expected number of steps it would take for a random walk to reach v_j from v_i	An asymmetric distance measure.

Name	Definition	Description
Commute time [41]	hitting time(v_j , v_i) + hitting time(v_i , v_j)	A symmetric version of hitting time.
Rooted PageRank [41]	The stationary distribution of v_j in a random walk from v_i ,where on every step there is a probability α that we return to the root, v_i .	A version of hitting time that weights closer nodes far more heavily.
SimRank [41]	The fixed point of the following recursive definition: $SimRank(v_i, v_j) = \begin{cases} 1 & \text{if } v_i = v_j \\ \gamma \cdot \frac{\sum_{v_a \in \Gamma(v_i)} \sum_{v_b \in \Gamma(v_j)} SimRank(v_a, v_b)}{\#(\{\Gamma(v_i)\}) \cdot \#(\{\Gamma(v_j)\})} & \text{otherwise} \end{cases}$	Two nodes are similar in the extent that they are joined to similar neighbours.

Table 4. Graph metric definitions

Name	Definition	Description
Density [69]	$\frac{\sum_{v_j \in V} degree(v_j)}{\#(V)(\#(V)-1)}$	How complete a graph is. Ranges from 0 (each node is isolated) to 1 (a complete graph). It is equal to the average standardised degree and computed as such.
Diameter [30]	$max(\{eccentricity(v_i) : v_i \in V\})$	The length of the longest shortest path in the graph.
Size	$\#(V_n)$	The number of nodes in the graph.
Number of edges	$\#(U_n)$	The number of edges in the graph.
Degree centralisation [69]	$\frac{\sum_{v_j \in V} [degree(v_x) - degree(v_j)]}{(\#(V)-1)(\#(V)-2)}$, where v_x has the maximum degree value in V.	How distributed the degree values are. Has a value of 0 in a circle graph (equal degree for all nodes) and a value of 1 in a star graph (completely unequal).

Name	Definition	Description
Betweenness centralisation [69]	$\frac{\sum_{v_j \in V} [betweenness(v_x) - betweenness(v_j)]}{(\#(V)-1)}$, where v_x has the maximum betweenness value in V.	How between on average all nodes are to each other. Has a value of 0 in a circle graph (equal betweenness for all nodes) and a value of 1 in a star graph (completely unequal).
Closeness centralisation [69]	$\frac{\sum_{v_j \in V} [closeness(v_x) - closeness(v_j)]}{2(\#(V)-1)(\#(V)-2)/(\#(V)-3)}$, where v_x has the maximum closeness value in V.	How close on average all nodes are to each other. Has a value of 0 in a circle graph and a value of 1 in a star graph.

3.2.3. Derived metrics

In addition to the metrics listed above, we can derive other metrics related to these. We can create graph metrics by taking the mean, median, mode, etcetera, of a group of monadic metrics. For example, $mean(\{indegree(v_i) : v_i \in V_4\})$ (the mean in-degree per node at time step four) is a graph metric. This is also an internodal but intratemporal graph metric – meaning that it is calculated over many different nodes but within a single time step. This is in contrast to a metric such as $mean(\bigcup_{n \in [1..40]} indegree(v_{61}) : v_{61} \in V_n)$, the temporal mean of the in-degree of node 61 in the time range from time step one to time step forty, which is an intertemporal metric. Finally, do not forget that most of the metrics defined in the tables can be calculated for any of the four types of links (in, out, bi and inout) and can use weighted or unweighted links.

3.2.4. Metrics useful for link prediction

The most useful of metrics for link prediction of those given in section 3.2.2 are dyadic metrics. This is because they usually represent the connective strength of a potential link, either by the distance between the nodes, or by their structural similarity. The next most useful set of metrics are monadic metrics. Though they do not take into account similarities between nodes, they generally give an indication of how important or prestigious an individual node is. By the principle of preferential attachment new links are likely to be formed incident to nodes of high degree. Graph and subgraph metrics are generally not useful for link prediction. This is because they provide no information pertaining to individual nodes and hence provide no information on potential links. At most these metrics can indicate when a graph or subgraph is ready for a new link to occur (for instance, when the ratio of nodes to links becomes unusually high), after which monadic and dyadic links could be used

to find where the link would occur. This might be useful in sequences of specialised networks in which links arise very seldom. However, many links arose in each time step in the data set used in this research. Finally, we recall from the previous chapter that Huang et al [35] found that the Katz measure was the most useful, followed by preferential attachment, common neighbours and the Adamic\Adar measure.

3.3. Input transformations

Input transformations are operations performed on the metric input data that prepare it for being learnt. Attribute selection is one example. Attributes in a learning model that have no predictive value cause the performance of most models to deteriorate [70]. We can therefore improve the performance of a model by selecting only relevant variables to be part of a data set and removing others. Removing attributes that seem irrelevant based on our knowledge of the concept in question is called the filter method. Removing attributes that would confound a specific learning model that has been chosen is called the wrapper method. Generally, attribute selection aims to remove redundant attributes (those that are highly correlated to other attributes), and irrelevant attributes (those that do not contribute much to prediction). Techniques used to achieve this include forward selection (where attributes are added one at a time to an empty set and then evaluated), backward elimination, beam searches and the use of genetic algorithms. These processes are all computationally expensive as they require performing multiple tests on each attribute multiple times. Other common transformations that were not necessary for this research include discretisation of numeric attributes, filtering noisy instances and creating synthetic attributes from existing attributes that better suit the chosen model.

3.4. Data modelling

Data modelling involves analysing attributes to find statistically significant relationships between them. For link prediction specifically, it involves finding relationships between the y-variable (whether two nodes are forming a new link) and all the other monadic and dyadic metrics (attributes) of the nodes. In other words, we are trying to find quantitative rules that classify dyads into two classes, or groups, based only on the values of their attributes. Fitting models to a data set is called data mining, part of the broader field of statistics [70]. There are many techniques that can accomplish this, including linear regression, logistic regression, Bayesian networks [51] and clustering. These techniques all consist of two phases: learning (or training) a model and using (testing) the model for prediction on new data.

3.5. False positives and instance weighting

Once a model has been learnt it can be used for prediction on new data. Given an instance to classify either positively or negatively (for a new link forming or not, respectively), four outcomes are possible. The instance can be classed as positive, and actually be a link forming. This is called a true positive, "TP". The instance can be classed as positive, but not be a link forming. This is called a false positive, "FP". In statistics this is called a type I error, i.e. rejecting a null hypothesis when it is true [67]. The instance can be classed as negative, and not be a link forming. This is called a true negative, "TN". The instance can be classed as negative, but actually be a link forming. This is called a false negative, "FN". In statistics this is called a type II error, i.e. accepting a null hypothesis when it is false [67]. If this seems counter-intuitive, remember that accepting a link prediction null hypothesis usually would mean accepting that a dyad is normal, negative, or not forming a new link. For the link prediction problem we are less concerned with false positives than false negatives. In other words, we are interested in any potential new links being predicted by a model even if they have a very low probability. We would be upset to miss any new links that form. This is because social networks are incredibly sparse. There are extremely fewer new links forming in a time step than the number of possible links that could form, which are quadratically proportional to the number of nodes in the graph. Thus an analyst studying a network would like to have any potential links highlighted by an artificial intelligence program. He or she can then examine the two nodes suggested in more detail manually. This would be true in crime, business or marketing analysis. However, because the ratio of forming links to potential forming links is so incredibly small almost no forming links would be predicted using a system trained on a random sample of social network instances. Instead we have to perform instance weighting. This means that the number of positive instances in the training set is not representative of the number of positive instances we would find in the whole social network. Rather, we increase the number of positive instances to be equal to the number of negative instances. This 50\50 split of classes that was chosen for this research is different from that used in most previous research methodologies. This is because the networks mentioned in the previous chapter were very small compared the networks used in this research. Thus most of the researchers mentioned in the previous chapter were able to use their entire network as training data, including all the positive instances. This is not possible using networks consisting of many thousands of nodes. Although finding true positives is important, the number of false positives cannot be allowed to become too large or the system will become useless. This is discussed later in the section on model evaluation.

3.6. Regression

Regression analysis is a statistical technique used to find an equation that describes a quantitative relationship between a dependent variable, y, and its causes, x_1, x_2, etcetera [67]. It is one of the many

possible learning techniques a machine learning system can use. Linear regression is discussed first to introduce the concept simply. Logistic regression is then discussed. It is a specialised form of regression that is not normally taught in basic statistics courses and is appropriate for link prediction.

3.6.1. Linear regression

Linear regression is a commonly used form of regression that assumes the relationship between the variables can be described by the equation of a straight line: $y = b_0 + b_1 x$. We can extend this model to include a series of x variables by using multiple regression. The regression equation then describes a plane in many dimensions. Linear multiple regression can be implemented simply by using matrix algebra [68]. This is described in the following paragraphs:

Let $\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ be the vector containing the y-values (what we are trying to predict) in n observations of the variables.

Let $\boldsymbol{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$ be the vector containing the coefficients of the x-variables (metrics) that we are trying to learn.

Let $\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$ be the n by p+1 matrix containing the values of the x-variables in the observation set. The first column consists only of ones to allow for the calculation of the constant coefficient, b_0. We want $\boldsymbol{Y} = \boldsymbol{X}\,\boldsymbol{B}$ to be a reasonable approximation for the actual relationship between x and y. Thus the values of $\boldsymbol{B}$ are found by minimising the residuals (difference between the predicted values of y given x, and the actual values of y in the sample observations) using the least squares method. This is called the maximum likelihood approach. The equation that gives a best fitting line for the coefficients is: $\boldsymbol{B} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$, where ' indicates the transpose of a matrix (in some texts the ' is replaced by T) and -1 indicates the inverse of a matrix. If $\boldsymbol{X}'\boldsymbol{X}$ does not have full rank and an inverse cannot be calculated for it, any generalised inverse will suffice [45].

3.6.2. Updating linear regression

There are times when one might wish to update a regression equation when new evidence becomes available, without recomputing the equation using all the original data. This might be the case when there is simply too much data to store in memory at one time and invert, or when a model changes over time and the coefficients of an equation must be altered to reflect this new situation. Both these cases are true when working with a sequence of sociograms: there are literally billions of possible combinations nodes to use as evidence points and the network changes as each time step passes. It is relatively trivial to separate the regression process into separate parts and store temporary values that are updated when new data becomes available. All that has to be stored to update an equation is the small matrix $\boldsymbol{X'X}$, though the values it contains will become quite large as they are sums of squares. We can separate the equation $\boldsymbol{B}=(\boldsymbol{X'X})^{-1}\boldsymbol{X'Y}$ into matrices for different ranges of n (observation sets or instances). This equation given for different values of n is obtained by simply chopping the original matrices off at a given row and putting the chopped off part in a new matrix. We now have $\boldsymbol{B_n}=(\boldsymbol{X_1'X_1}+\boldsymbol{X_2'X_2}+\ldots+\boldsymbol{X_n'X_n})^{-1}(\boldsymbol{X_1'Y_1}+\boldsymbol{X_2'Y_2}+\ldots+\boldsymbol{X_n'Y_n})$. So if we compute an equation at time n-1, we need to store only $\boldsymbol{X_{n-1}'X_{n-1}}$ and $\boldsymbol{X_{n-1}'Y_{n-1}}$. Then the coefficients at time n can be calculated using $\boldsymbol{B_n}=(\boldsymbol{X_{n-1}'X_{n-1}}+\boldsymbol{X_n'X_n})^{-1}(\boldsymbol{X_{n-1}'Y_{n-1}}+\boldsymbol{X_n'Y_n})$ [24].

3.6.3. Logistic regression

Linear regression is the most commonly used form of regression but it is not suitable for the purposes of link prediction. This is because the range of y using a linear equation is $(-\infty;\infty)$, but we are trying to predict a binary or dichotomous variable, i.e. one or zero, whether a link will arise or it will not. Hosmer and Lemeshow recommend logistic regression as the suitable method for binary data [34]. It is a standard social network analysis technique and has been used by many researchers in the past, including [39][36][52][28][54] and [64]. Instead of basing our predictions of y on the equation $y=b_0+b_1x_1+b_2x_2+\ldots+b_nx_n$, logistic regression is based on the equation $y=\pi(x)=\frac{e^{b_0+b_1x_1+b_2x_2+\ldots+b_nx_n}}{1+e^{b_0+b_1x_1+b_2x_2+\ldots+b_nx_n}}$. The advantages of this technique is that the range of y is $[0;1]$ and that it uses similar principles to linear regression. Linear regression uses the least squares approach to minimise residuals. This is simply a specific application of the maximum likelihood approach. The maximum likelihood approach can be used with logistic regression but a new maximum likelihood function must be defined to suit logarithms. The derivation of a suitable function is given on page nine of [34]. It is proposed that a likelihood function for the vector of

coefficients, **B**, of an equation should be $l(\boldsymbol{B}) = \prod_{i=1}^{n} \zeta(x_i)$, where $\zeta(x_i) = \pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i}$.

Because logistic regression is non-linear the coefficient calculation process is not a simple matrix combination, but rather an iterative process in which the coefficients are gradually converged upon. This makes it more complex and far slower than linear regression.

3.7. Model evaluation

A link prediction model is evaluated based on how accurate its predictions are, given a sample of test data. It is common practice to separate a large data set into two parts: one for training a model and one for testing the trained model [70]. The quality of a model can be found by calculating the percentage of test data instances classed correctly, using only the given test metrics. This is the success rate of the model, which we are aiming to get as close to 100% as possible. The accuracy of a model depends on the statistical system chosen and the variables that are chosen to train it. In the same way a linear model would not be suitable for quadratic data, if a model is chosen with metrics that have no correlation with the class types, the accuracy of the model will be low. As mentioned in the previous section on true positives, we are more interested in finding all true positives, even if a number of false positive are included in our predicted results. The true positive rate is defined as $\frac{TP}{TP+FN}$ and is given as a percentage. The numerator represents all the forming links we correctly predicted. The denominator is the sum of these links, as well as the links we did not predict. In other words the denominator represents the total number of new links that formed in a time step. Thus the TP rate is the percentage of all links we discovered. The aim of link prediction is to maximise the TP rate, rather than the overall success rate, which is defined as $\frac{TP+TN}{TP+TN+FP+FN}$. However, we would like to minimise the number of false positives we find, to make a human analyst's job easier, but this is not as important as maximising the TP rate. In other words we would like to minimise the number of false positives, as a percentage of all the positive classifications the system makes. We would like to minimise $\frac{FP}{TP+FP}$, the percentage of predicted forming links that are incorrect. This is different from the standard data mining definition of the false positive rate, defined as $\frac{FP}{FP+TN}$, which represents the number of negatives we predicted as positives as a percentage of all the negatives in the set. It is in fact the true positive rate for the negative class. In other words we are trying to find the true positive rates of both classes, forming links and unconnected links.

Another way to evaluate the accuracy of a model is to use the kappa statistic, defined as $\frac{P(A)-P(E)}{1-P(E)}$, where P(A) is the total percentage accuracy of instances predicted by the learning system and P(E) is the total percentage accuracy of instances predicted by random guessing [11]. If an equal number of positive and negative instance are used P(E) = 0.5 and 1 – P(E) = 0.5. The kappa statistic ranges from -100% to 100% and tells us how much more useful a model is, compared to a random guess [22]. However, it does not take the increased value of true positives into account, and weights true positives and true negatives as equally desired [70]. Values higher than 40% are said to indicate "good agreement beyond chance" [22]. This statistic has not been widely used in social network analysis, but is extremely useful as it allows researchers working on data sets with different numbers of new links forming to compare their predictive accuracy. This cannot be done with the simple percentage total accuracy statistic, as it does not take into account the number of positive and negative instances.

3.8. The methodology used in this research

This section discusses the exact methodology used in this research. The methodology is placed in the theory framework created by previous researchers.

3.8.1. Data source

The data set used for this research is the Pussokram Internet dating site message exchange log. Pussokram is a Swedish dating site that translates roughly to "hug and kiss" in English. Users must be registered with the site to exchange messages. It was studied by Petter Holme for his studies on temporal trends [32][33]. It consists of 500 time steps starting with a graph consisting of 24 nodes, 21 directed links and 6 bidirected links. By the final time step the graph consists of 21541 nodes, 20003 directed links and 6863 bidirected links. The time range used in this research was from time step 50 to time step 150. The respective nodes, directed links and bidirected links are 5736, 5286 and 1713 for time step 50 and 12265, 11328 and 3813 for time step 150. The total number of new links formed over this time range is 9939. This gives an average of 99 forming links per time step. The experiments were conducted using the specified time range for two reasons. Firstly, by starting at time step 50 the metrics are calculated on a graph that is already sizeable and therefore realistic and interesting. Secondly, computing metrics for one hundred time steps instead of the available 450 time steps saves time (as metric calculations can take several days) without affecting the worth of the results.

The final experimental chapter uses the Netcash email log data set in addition to the Pussokram data set. The performance of the Netcash data set is compared to that of the Pussokram data set to see if the classification system used performed equally well on a real-world email database. The Netcash data were extracted by this researcher from an email log consisting of email records collected over a period of ten months during 2005 and 2006. It consists of 268 time steps starting with a graph consisting of 840 nodes, 174 directed links and 16 bidirected links. By the final time step the graph consists of 40756 nodes, 1180 directed links and 252 bidirected links. The time range used in this research was from time step 50 to time step 150. The respective nodes, directed links and bidirected links are 14287, 595 and 131 for time step 50 and 26654, 906 and 227 for time step 150. The total number of new links formed over this time range is 4765. This gives an average of 48 forming links per time step. The ratio of nodes to links in the Netcash data set is vastly greater than the ratio in the Pussokram data set because email can have multiple recipients. We could say that in time step 150 there is a core set of only 227 nodes. The other email recipients have very weak one-sided ties to the other users.

3.8.2. System overview

This section gives an overview of the system, *Tesa* (a contraction of the words "temporal sociogram analysis"), programmed for this research in Java to analyse sociogram sequences. As this research focusses on the mathematics of link prediction only a brief description of the system is given. A diagram of the most important classes is shown below.

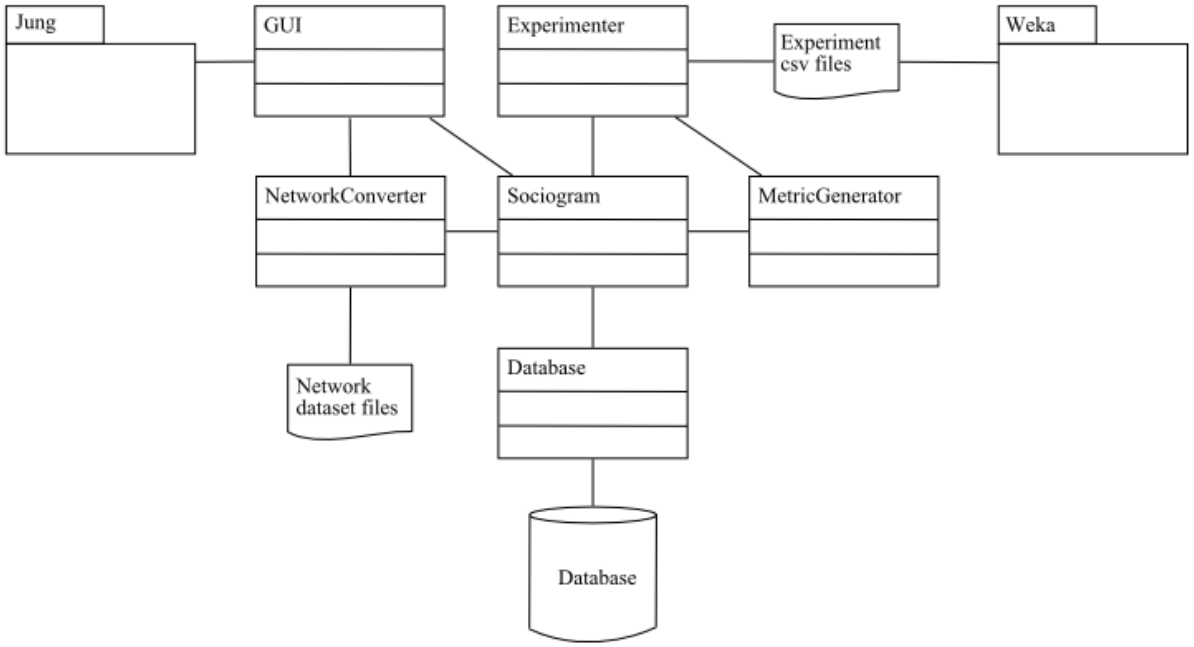

Figure 12. Sociogram sequence analysis system overview

The system consists of the following classes:

- GUI and Experimenter, which allow the user to interact with the system,
- NetworkConverter, Sociogram and MetricGenerator, which perform the mathematical and graph theoretic computations and conversions, and
- Database, which stores and retrieves sociogram sequences from the MySQL database.

The system uses two external Java packages, Jung and Weka. Jung (Java Universal Network\Graph framework) is used by GUI to display graphs on the screen. Weka is used to perform statistical analysis on the data files generated by Experimenter. Each class is now discussed in turn.

The GUI was used to display the graphs in the data set at various time steps. A screen shot is shown below.

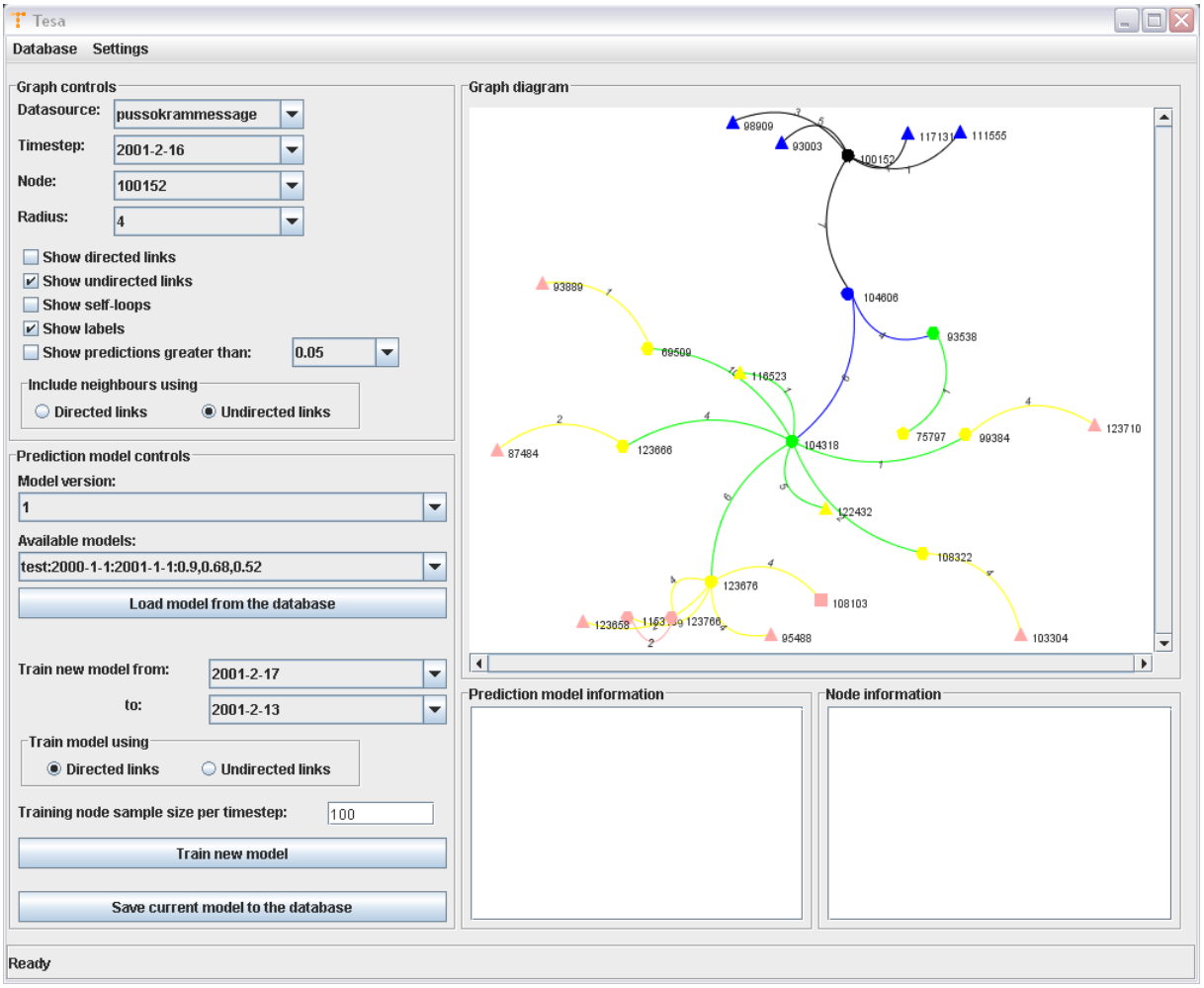

Figure 13. GUI screen shot

The graph is shown in panel at the top right of the screen. Since sociograms are generally very large it is practically impossible to show an entire graph at one time. Instead, in keeping with the local analysis focus of this research, only egocentric subgraphs were shown at any given time. An egocentric graph is one that is centred around a single individual. For instance, when researchers speak of constructing an egocentric graph they mean that they asked only a single individual to describe his social network. For example, in the figure above the black node at the top of the screen is the node of focus. Its neighbours are shown in blue, its neighbours' neighbours are shown in green, its neighbours' neighbours' neighbours are shown in yellow and its neighbours' neighbours' neighbours' neighbours are shown in pink. There is an exponential increase in the number of nodes shown in a subgraph as we increase the radius from the node of focus. Thus it becomes slow and impractical to show subgraphs at a radius greater than five for popular nodes. Different colours were used at each radius to clearly highlight the structure of the subgraph. The nodes and the strength of the links (number of messages exchanged between nodes) could be labelled on the graph. Finally the number of sides of the polygon illustrating the node was proportional to the degree of the node. Thus nodes

with a degree of three or less are shown as triangles, nodes of degree four are shown as squares and so on, up to nodes with a degree of twenty, where after the shape is indistinguishable from a circle. From a system perspective, GUI stored graphs as Sociograms, which were in turn loaded into memory using the Database class to access the actual database stored on disk.

The Experimenter class was used to perform the various experiments discussed in subsequent chapters. It loaded Sociograms into memory for a given time range, selected random instances of unconnected- and forming links and had the MetricGenerator class calculate various metrics for these instances. The instances and their associated metrics were saved to csv files. Experimenter performed higher level functions than the MetricGenerator class. For instance, Experimenter aggregated and then calculated temporal statistics from all the metrics calculated by MetricGenerator at various individual time steps.

NetworkConverter is a seldom used class that converts graphs from one form to another. For instance, the data set stored by Pussokram in one specific format was converted to a format usable by the Database class so that it could be stored in the database. Additionally NetworkConverter is able to output Sociograms as Pajek (a popular social network analysis program) files and convert email logs into Sociograms.

Sociogram represented social networks and stored information in Node and Link classes. Specifically, Java *TreeMaps* were used to ensure an access time of constant order. Directed links were stored from the origin node to the destination node and from the destination node to the origin node. This made certain calculations that involve finding common neighbours extremely fast, as links both to and from a node are immediately available. Bidirected links were stored in a separate TreeMap.

The MetricGenerator class computed the various metrics described in the experiments. This class implemented the standard metric definitions that were altered to allow for computation at any given radius for the local subgraph of a given node. It implemented Brandes' algorithm [9] for fast computation of betweenness (also altered to accommodate local subgraphs). Metrics can also be calculated for any given link type. E.g. the common neighbours or distance of two nodes can be calculated using only links pointing towards the given nodes, pointing out from them, or pointing in both directions.

The Database class has almost no noteworthy elements and is simply a way to interface with the MySQL database holding the social data sets. Its only other use is to calculate the metric *recency*.

Monadic recency is the number of time steps that have elapsed since a node last sent or received a message. It was calculated by the Database class rather than the MetricGenerator class, as might be expected. This is because calculating recency using MetricGenerator would require loading many Sociograms into memory, which is relatively slow. Instead, the process was accomplished more quickly by using Database to search for recent links using SQL queries on individual nodes in the database itself.

3.8.3. Metric computations

One general computational methodology was used for the experiments described in the following chapters. One experiment is described per chapter, but they all use the same data set and similar computations. They differ slightly and these differences are described in the chapters themselves. The general methodology is simple and described now. A y-variable was chosen for a pair of nodes, for instance whether a new link is forming in the next time step, and an equal number of positive and negative cases were chosen for one hundred time steps. Dyadic and monadic metrics were calculated for each dyad instance in each time step. All these values were written to a csv text file. An example extract of such a file is shown below, where columns are lined up neatly and the commas separating the values in each row into columns have been removed.

Table 5. Example csv file

TS	From	To	CF	NDegreeF	NDegreeT	Size2F	Size2T	CN	JC	AA
50	101033	33161	0	0	3.49E-004	68	12	0	0	0
50	104447	123807	0	5.23E-004	3.49E-004	11	4	1	0.33	2.1
50	106156	125313	0	1.74E-004	3.49E-004	2	2	0	0	0

Each column stores an attribute. The first row of the file contains the attribute names (column headings). Every subsequent row in the file represents an instance. Relational data, such as that found in the relationships between people in social networks, needs to be converted from attributes contained in a single instance for each person to a single instance for a relation between two people. Converting this information into a flat file format, such as the one shown above, is called denormalisation. These values were then analysed by Weka, a set of artificial intelligence learning models [70]. Liben-Nowell observed that we cannot possibly hope to predict links arising incident to new nodes in the current time step [41]. Thus in this research, similar to his methodology, we exclude all links in the next time step that form incident to nodes that are not in the graph in the current time step from our set of positive cases. The metrics used in the following experiments were chosen as they were the metrics having the highest usefulness for prediction, according to Huang et al [35]. Their standard metrics definitions were used, as given in [1], [35], [41], [69] and [30]. Some original

metrics were invented for this research and are listed in the metric definitions tables in this chapter. Differing trivially from previous research, the distance from one node to another was attempted to be found only up to a distance of thirty nodes. A review of the data used revealed that it is highly unlikely any nodes will be connected at a distance greater than twenty; thus searching beyond this distance is unhelpful. Previous researchers searched the entire graph to find distance because their graphs were small enough to make this feasible, or because they were using only one time step.

This paragraph describes the general procedure that was used to calculate a set of metrics. For every time step the node set was searched to find 100 dyads where a link is forming. This requires checking that the nodes are unconnected in the current time step and connected in the next time step. In certain time steps, less than 100 forming links were available. Thus the sample size of most of the experiments is 9939 instances per class, rather than 10000. After the forming links were chosen an equal number of unconnected dyads (or dyads with a hidden link as the case may be) were chosen. Metrics were calculated for each instance and immediately written to an output file. The exception to this procedure is the temporal statistics experiment. In this case metrics had to be computed for up to twenty time steps at a time in order to calculate temporal values over all the time steps.

3.8.4. Statistical analysis

This research follows the standard practice discussed earlier, separating the dyad instances using a 70\30 split. 70% of the data is used for training and the remaining 30% of instances are used to test the model after training. A statistical learning system was used to classify positive and negative cases according to their metrics. The accuracy of the system was given, both as an overall percentage, true positive rates for both classes and as a kappa value. Additionally, the difference in the metric means of the positive and negative cases was evaluated for traditional statistical significance. The statistical evaluation of the results used in this research is similar to the standard approach used in most related studies (e.g. [54]). Taskar et al [64] used an eight-fold train-test split and Popescul and Ungar [54] used a ten-fold split, which was not necessary in this research due to the abundance of data. Popescul and Ungar used an equal number of positive and negative cases, the same method as used in this research. Since there is an equal number of positive and negative instances in the data set a random prediction would have an accuracy of 50%. Thus any trained model has to have an accuracy of above 50% to be of any worth at all. In 2003 Liben-Nowell and Kleinberg's study of link prediction used two years worth of training data and test data from the subsequent two years [42]. They evaluated the accuracy of their method using the true positive rate only, but gave the accuracy as a factor improvement over random accuracy. This makes it difficult to compare their results with other studies, which may use different quantities of positive and negative cases. Additionally, Liben-

Nowell took his predictions to be the dyads scoring highest on certain metrics (common neighbours, Katz, etc) and did not use a separate data mining system, as was done in this research.

The individual predictive accuracy of each metric was evaluated, as well as the combined accuracy of sets of multiple metrics. An attribute selection method in Weka was used to discover the most useful combinations of metrics to use in each experiment. Weka allows a user to combine a variety of attribute evaluators and search methods to discover useful sets. Specifically, the *ClassifierSubsetEval* attribute evaluator was used to evaluate the usefulness of each attribute when using logistic regression as the classification scheme. Two different search methods were used to generate sets, the *BestFirst* and *GeneticSearch* methods. *BestFirst* starts with an empty set and adds nodes with a high classification ranking until the usefulness of the set starts to decline. Then the algorithm backtracks and tries to include different nodes in the set to see if the accuracy can be increased. *GeneticSearch* uses a simple genetic algorithm to create random attribute sets, evaluate them, and breed new generations of more accurate sets. Additionally, standard statistical hypothesis testing was performed to test the difference between the mean metric values for each class (e.g. links that are hidden, forming or unconnected dyads). The test statistic used was a normal distribution two sided difference of means test. The standard formula for this test statistic is $z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$, where $\bar{X}$ is the mean metric value, σ^2 is the variance of the metric, n is the sample size, the subscript 1 denotes a value related to the negative cases and the subscript 2 denotes a value related to the positive cases [67]. Although nodes in social networks are not normally distributed (it was discussed how nodes and links obey a power law relationship in chapter two) the central limit theorem of statistics allows us to use this simple statistical test of normal distribution to determine the significance of the results [67]. Simply put, the central limit theorem states that if we have a very large number of instances for which we are calculating means, it does not matter what the underlying distribution was and we can treat it as if it is were a normal distribution.

3.8.5. Comparing these results to others

Liben-Nowell's sociograms were a lot smaller than the ones used in this research [41]. Thus he was able to work with all arising links when performing prediction. Because of the huge size of the networks used in this research all possible links could not be investigated. Instead, a sample of all possible links had to be used. For every positive case (a link arising in the next time step) one

negative case (a pair of nodes where a new link does not arise in the next time step) was randomly selected from the graph. This gives a random prediction an accuracy of 50%, in contrast to the random predictions in Liben-Nowell's networks, which had an accuracy of roughly 0.2%. His best predictor in his best data set was correct on only 16.1% of predictions. This leaves us with a statistical comparison problem since most link prediction research presents accuracy in terms of factor improvements over random predictions. An improvement over random predictions in a sparse graph where perhaps only 0.2% of cases are positive will be much higher than using test data where there is a 50% split between positive and negative cases. However, even though using factor improvements as a benchmark will be unhelpful in this case, Liben-Nowell's figure of 16.1% can be used for comparison.

3.9. Conclusions

This chapter described the previous statistical and research methodologies that were used by other researchers, the methodology used in this research and the software system used to conduct the research. It explained how the metrics computed from social networks are used in data mining experiments. A general experimental methodology was presented that forms the basis for the experiments described in the following chapters. Each of the experiments has peculiarities that slightly deviate from or augment the general methodology. These differences are discussed in the experiment chapters. The next chapter begins the experimental section of the discussion. We start with an investigation into link prediction and link detection.

Chapter 4. Link prediction versus link detection

This chapter addresses the differences between link prediction and link detection. It defines each problem, suggests intuitively how the graph structure surrounding new links might differ from the structure surrounding hidden, but existing, links and describes the experiment undertaken to verify these intuitions. It uses definitions and concepts discussed in detail in the earlier background chapters on social network analysis, including link prediction and link detection. Only methodology and motivations peculiar to this experiment are given in this chapter; the general statistical methodology and motivation in terms of previous research for the methodology used in this experiment is given in the previous background chapter on statistical methodology.

4.1. A definition of link prediction and link detection

Link prediction is defined as determining whether a link will arise in the next time step between two nodes that are unlinked in the current time step [42]. In other words, if and only if two nodes are not adjacent in the current time step but are adjacent in the next time step, we say there is a link forming in the current time step. For computational purposes we create a binary variable called LinkForming, representing the state of this possible new link. For example, $LinkForming_8(93,110)$ equals one if nodes 93 and 110 are not adjacent at time step 8, but are adjacent at time step 9. This variable would equal 0 if either nodes 93 and 110 are adjacent at time step 8, or if they are not adjacent at time step 9. We now define the related problem of link detection. Link detection is defined as determining whether a link exists between two nodes in the current time step, without a link between the nodes being present in the graph. This would occur when graph data is incomplete and a link between two nodes has been hidden intentionally or through negligence. This idea has received hardly any individual attention, though it has been used or mentioned in [41][39][54] and [63]. To simulate this in a graph we need to choose a connected dyad and then remove its link to create an unconnected dyad. This acts as a hidden link, while the surrounding graph structure remains untouched.

4.2. Motivation for the investigation

Until now no research has been conducted into whether detection is similar to prediction and whether identical techniques can be used to solve the problems. Link detection should be as useful a problem to solve as link prediction, espcially for criminal analysis. This problem is similar to link prediction in that it attempts to find missing links, but differs in that the links are hidden in the current time step, either through missing data or intentional subterfuge, and may have existed for long periods of time.

It has to be assumed that researchers believe the two problems of detection and prediction are similar, and that techniques that work on prediction can be used for detection. In other words, researchers have implicitly assumed that the values of metrics indicating that a link is forming in the current time step will be the same as the values of metrics indicating that a link is hidden in the current time step. This may be true, but intuitively it seems as though there might be a difference between the two situations. One would think that the graph structure surrounding two individuals who have known each other for a long time would be different from that surrounding two individuals who are about to meet. For instance, two connected nodes might have more mutual friends than unconnected nodes, they might have fewer non-mutual friends, and their neighbours at a distance of two links away might themselves be at a closer distance and might have more mutual friends. This leads to three possibilities: that the metric values of connected links and forming links are identical, that the values are similar and differ only in magnitude, or that the values are significantly different in structurally important ways. These possibilities are illustrated below. In the following sets of figures, the orange link in the diagram on the left represents a hidden link between the two orange nodes (a link detection problem) and the orange link in the diagram on the right represents a link that will form in the next time step (a link prediction problem).

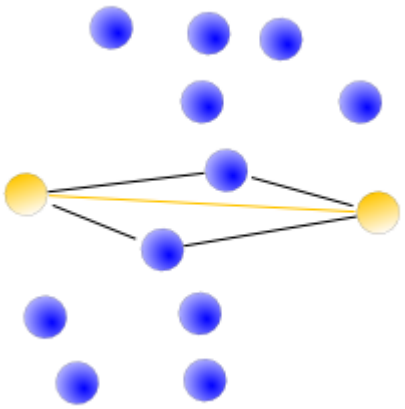

Figure 14. A hidden link between two nodes with two common neighbours

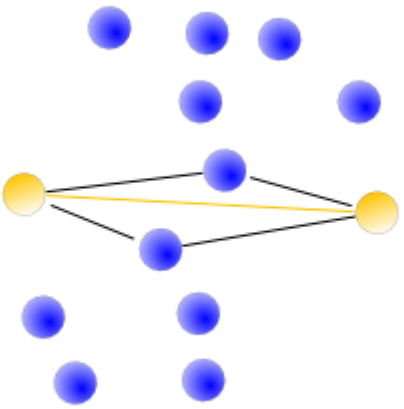

Figure 15. A forming link between two nodes with two common neighbours

The diagrams above illustrate the first possibility (identical problems). The dyad on the left, having the hidden link, and the dyad on the right, having the forming link, will have the same metrics. This is because they both have two common neighbours. This is the view that researchers probably currently have of the problems, that they are identical.

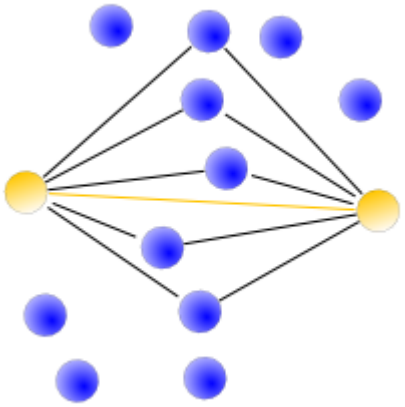

Figure 16. A hidden link between two nodes with many common neighbours

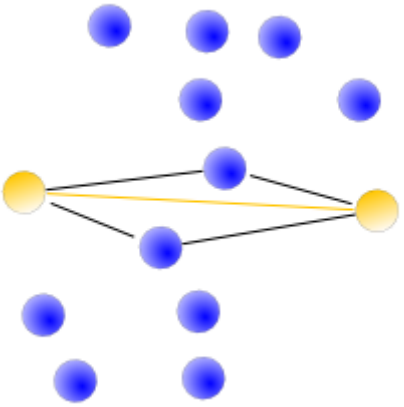

Figure 17. A forming link between two nodes with two common neighbours

The diagrams above illustrate the second possibility (difference in magnitude). The dyad on the left, having the hidden link, and the dyad on the right, having the forming link, will have similar but not identical metrics. The dyad on the left has more common neighbours and thus will have higher metric values than the dyad on the right. This might occur since people who have links that are hidden but well established might have far more mutual friends than people who about to meet.

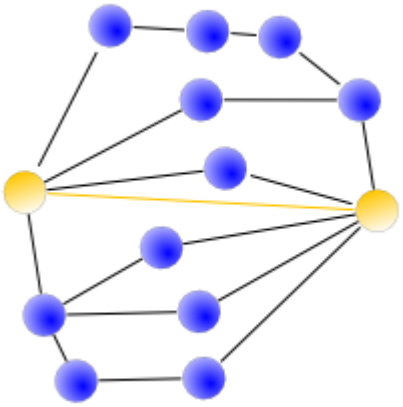

Figure 18. A hidden link between two nodes with few common neighbours at small radii

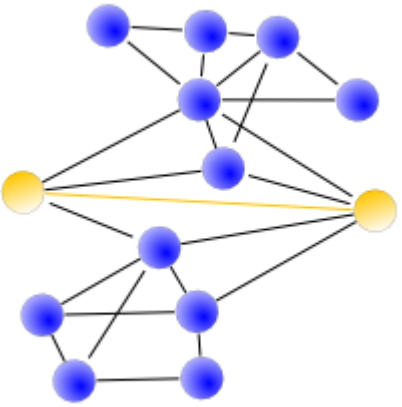

Figure 19: A forming link between two nodes with many common neighbours of high degree

The diagrams above illustrate the third possibility (difference in structure). The dyad on the left, having the hidden link, and the dyad on the right, having the forming link, will have very different metrics. The dyad on the left has only one common neighbour whereas the dyad on the right has

three. Furthermore, the dyad on the left has many more paths of distance two and few neighbours of high degree. This might occur if people trying to hide that they know each other communicate through long chains of acquaintances who do not know each other. The dyad on the right has many common neighbours of high degree. This might occur when two people who belong to two highly connected social groups are about to meet. Both this third possibility and the preceding one illustrate cases where a different classification system will have to be used for hidden and forming links. In other words, though both problems can be solved through analysing the values of the same metrics, the metric values themselves might be very different.

4.3. Hypothesis statement

The null hypothesis of this experiment is that link prediction can use the same techniques in the same way to predict links forming between nodes as link detection does to detect hidden links. More mathematically, the null hypothesis is that the kappa value of a regression performed on hidden and forming links will be less than 40%.

4.4. Methodology

This section describes the experiment undertaken to find if there are differences between prediction and detection. It follows the general methodology described in the research methodology chapter. Sample metric values were taken for three classes, 19878 instances, over 100 time steps. The y-variable chosen equalled zero if two nodes under consideration were unconnected, equalled one if the two nodes had a hidden link and equalled two if the nodes had a forming link. The x-variables included the monadic metrics: normalised degree and the neighbourhood size at a radius of two[3]; and the neighbour-based dyadic metrics: common neighbours, Jaccard's coefficient, Adamic\Adar similarity and preferential attachment. The distance metrics used were: the simple unweighted distance and the Katz measure. These metrics were defined in the tables in chapter three. The column headings for the data set calculated are:

NDegreeF, NDegreeT, Size2F, Size2T, CN, JC, AA, PA, Dist, Katz.

The metric names are abbreviations or initialisms of the corresponding metric names mentioned in the previous paragraph. The monadic metric names are suffixed with *F* (from) and *T* (to) to indicate which node in the dyad they represent. Statistics were calculated for each metric for the positive and negative cases and are displayed in the results section below. If there is indeed a structural difference in the networks surrounding hidden and forming links then a statistical system should be able to

3 Equivalent to the local metric *degree at a radius of two*, discussed in the chapter on local metrics.

correctly classify each type of link by the associated metric.

4.5. Results

This results section presents both the statistics of the metrics calculated and their predictive accuracy individually and in sets. These results are presented in three tables. The first table, 6, shows the statistical means test for the hidden and forming classes using every metric. The second table, 7, shows detailed results of the accuracy attained by logistic regression using each metric individually. The third table, 8, shows the accuracy attained using sets of metrics. Thereafter two more similar tables are presented. These tables, 9 and 10, hold results for three classes, unconnected links, hidden links and forming links. This is perhaps a better simulation of reality, where we would have to distinguish not only between hidden and forming links, but also between them and links that are completely unconnected and will remain that way.

Each row in the following table gives the statistics associated with a certain metric. The first column in the table gives the name of the metric. The second and third columns give the average of the values for the metric for the hidden and forming dyads respectively. The fourth and fifth columns give the standard deviation of the values for the metric for the hidden and forming dyads respectively. The mean difference column shows the value of the hidden mean minus the forming mean. Thus, it is negative when metrics associated with a forming dyad are larger than those associated with an hidden dyad. The test statistic column gives the value of the Normal distribution test statistic, calculated using the mean and standard deviation values as described in the research methodology chapter. The subsequent column gives the significance level associated with the test statistic. The final column shows the kappa statistic for the metric, as given by Weka using logistic regression. It is not related to the standard hypothesis testing statistics given in the previous column.

Table 6. Mean metric values

Metric	Hidden mean	Forming mean	Hidden standard deviation	Forming standard deviation	Mean Difference	Test statistic	Significance level	Kappa
NDegreeF	0.0002	0.0009	0.0005	0.0022	-0.0006	-27.6	0.01%	24.30%
NDegreeT	0.0011	0.0008	0.0026	0.0021	0.0003	9.9	0.01%	6.60%
Size2F	19.9522	44.9479	41.9858	93.6341	-24.9957	-24.3	0.01%	17.11%
Size2T	60.4137	41.9855	117.1428	88.4634	18.4282	12.5	0.01%	10.45%
CN	0.0405	0.0231	0.2271	0.1791	0.0174	6.0	0.01%	1.44%
JC	0.0102	0.0044	0.0748	0.0474	0.0059	6.6	0.01%	1.51%
AA	0.0611	0.0329	0.3666	0.2701	0.0283	6.2	0.01%	1.44%
PA	18.5685	33.2047	75.8562	114.0258	-14.6363	-10.7	0.01%	9.30%
Dist	4.8046	5.0648	1.5948	1.5800	-0.2603	-11.6	0.01%	-1.80%

Metric	Hidden mean	Forming mean	Hidden standard deviation	Forming standard deviation	Mean Difference	Test statistic	Significance level	Kappa
Katz	0.0323	0.0348	0.0517	0.0467	-0.0025	-3.6	0.05%	7.79%

The table below lists each metric individually in separate rows and describes their usefulness (contribution to accuracy) in a logistic regression. The first column shows the kappa statistic, which is a measure of how much more accurate a prediction is, compared to a random prediction. The metrics are ranked in descending order of their kappa value. The total accuracy can be described in different ways. The last column shows the overall accuracy of the regression. As explained in chapter three, this value can be deceptive and is not as useful as the kappa statistic. The third and fourth columns are important and show the true positive rate for each class.

Table 7. Metric predictive accuracy, ranked by kappa

Metric	Kappa	Hidden TP rate	Forming TP rate	Overall accuracy
NDegreeF	24.30%	79.2%	45.1%	62.2736%
Size2F	17.11%	79.7%	37.4%	58.719%
Size2T	10.45%	32%	78.5%	55.0302%
PA	9.30%	81.4%	27.9%	54.8793%
Katz	7.79%	60.8%	47%	53.9571%
NDegreeT	6.60%	22.8%	83.9%	53.0349%
JC	1.51%	3.5%	98.1%	50.3186%
CN	1.44%	3.6%	97.9%	50.285%
AA	1.44%	3.6%	97.7%	50.285%
Dist	-1.80%	17%	81.2%	48.7928%

The following table uses the same columns as the one above, but instead of showing the accuracy of each metric individually it shows the accuracy of regressions performed with different sets of metrics. The metric subset column describes the type of metrics used in italics and then lists all the metrics in the set. Weka's best first search found the same set of metrics as the genetic search (it was stated in chapter three that both searches would be used in the experiments). Thus the regression for those metrics is shown only once.

Table 8. Metrics set predictive accuracy

Metric subset	Kappa	Hidden TP rate	Forming TP rate	Overall accuracy
All metrics	27.61%	75.5%	52.0%	63.8833%
Weka's logistic subset classifier genetic attribute selection search: NDegreeF, NDegreeT, Size2T, AA, PA,Dist	27.5%	75.5%	51.9%	63.833%

The table below has the same format as the second table in this section but lists the results for three classes, not just two. It compares the classification accuracy of a logistic regression for unconnected-, hidden- and forming dyads.

Table 9. Metric predictive accuracy, ranked by kappa

Metric	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
NDegreeT	18.15%	80.8%	32.7%	22.7%	45.5064%
Size2T	15.02%	80.2%	36.9%	12.9%	43.4049%
PA	14.88%	87.1%	18.8%	22.9%	43.0807%
NDegreeF	12.76%	70%	11.7%	43.6%	41.9405%
Katz	12.1%	80.3%	5.9%	37.8%	41.5269%
Size2F	9.34%	71.4%	12.1%	35%	39.6714%
CN	1.63%	99.8%	3.5%	0%	34.6188%
AA	1.63%	99.8%	3.5%	0%	34.6188%
JC	-1.64%	85.1%	11.7%	0%	32.3944%
Dist	-3.31%	17.1%	32.9%	43.5%	31.1424%

Like the previous table, the following table has a format that has already been described. It also displays three classes rather than two. Once again Weka's best first search found the same set of metrics as the genetic search. Thus the regression for those metrics is shown only once.

Table 10. Metrics set predictive accuracy

Metric subset	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
All metrics:	28.84%	73.9%	44.4%	39.3%	52.5933%
Weka's logistic subset classifier genetic attribute selection search: NDegreeF, NDegreeT, Size2T, Size2F, Dist, CN	29.4%	75.6%	47.2%	36%	52.9622%

4.6. Conclusions

This section draws conclusions from the data presented in the tables and graphs above. Firstly, a note on interpreting the mean differences: bear in mind that a negative difference means that the forming link metrics were larger than the connected (hidden) link metrics, and vice versa for positive differences.

We first examine the table of means, Table 6. The mean differences are found to be very highly significant. This should negate the null hypothesis. However, looking at the individual kappa values for these metrics we see that none of them are higher than 40%. This is the minimum level for "good agreement beyond chance". However, when used in combination these metrics may prove more useful. We examine that possibility later in this section. Before that we examine the metrics themselves to see if they indicate any structural differences between hidden and forming links. The normalised degree and size at radius two metrics (the first four rows) indicate the popularity of a node and the popularity of the node's neighbours respectively. We can see that for hidden links (connected links) there tends to be a disparity in the from node's and the to node's metric values. However for forming links the values tend to be equal. The common neighbour-based metrics (CN, JC and AA) all have higher values for the hidden links than for the forming links. They tend to be approximately double, indicating that people who know each other have roughly twice as many mutual friends as people who are about to meet. The preferential attachment value is higher for forming links than for existing links, again being roughly double. This indicates that a dyad with a high number of total neighbours has more chance of forming a link than a dyad with fewer. The distance and Katz metrics are roughly equal at approximately five, indicating that distance is not a distinguishing factor for

hidden and forming links.

The second table, Table 7, does not give us much more information. It shows us that a node's popularity and a dyad's preferential attachment are the most useful metrics for classification and that common neighbour-based metrics and distance metrics are the least useful. The third table, Table 8, showing predictive accuracy for metrics used in combination, shows equal accuracies for the rows displaying all metrics and a selected lesser number. Both these regressions have total accuracies of only 64%, with kappa values of only 28%. This is barely better than using *normalised degree from* by itself, which had a kappa of 24%. The last two tables, 9 and 10, contain another class, unconnected dyads, in addition to the hidden links and forming links we have been considering. We can see that the metrics that were previously useful now become less so. Size at radius two becomes far less useful, but preferential attachment and the Katz number have higher kappas than before. However, there is almost no change in the predictive accuracy when using all the metrics together. Table 10 shows us that the predictive accuracy of these metrics remains with a kappa of 29%. This is barely one percent higher than the prediction using the two classes by themselves. Thus it appears that although unconnected-, hidden- and forming dyads have different structural subgraph patterns, these patterns cannot be used to accurately distinguish between the dyads and the null hypothesis cannot be completely refuted. Researchers are thus mostly correct in treating the two problems as essentially identical.

To summarise, we have found that:

- There is a difference in structure between hidden and forming links: hidden links have twice as many common neighbours, half as large a preferential attachment and more disparate nodal degrees than forming links. Distance is not a distinguishing factor.
- Although all metrics differences are very highly significantly different no individual metric is useful for regression in accurately classifying the two classes.
- Furthermore even when using all the metrics together it is not possible to successfully distinguish hidden links from forming ones (a kappa of only 29%).

Chapter 5. Temporal link analysis

This chapter describes an investigation of whether and how temporal analysis techniques can aid link prediction. It explains how current link prediction techniques completely ignore temporal information by using only static metrics, suggests statistics that would quantify temporal information and describes the experiment undertaken to determine the usefulness of these statistics. Only methodology peculiar to this experiment is given in this chapter; the general statistical methodology is given in the background chapter on research methodology.

5.1. Deficiencies in static link prediction

I believe the definition of link prediction by Liben-Nowell and Kleinberg is deficient [42]. Their approach to the problem is limited as it attempts to predict the evolution of a complex entity over time from a snapshot – the previous time step. Consider the analogy of trying to predict the position of a thrown ball a second from now, given only a photograph of the ball when it was released from the thrower's hand. It is true that the ball's position can be approximately predicted, but it would be better to have seen the ball move through the air. In other words, we need to know the velocity of a social network, not just its position – we need to examine more time steps than just the previous one. To continue the analogy, the position of a social network is given by traditional social network analysis metrics calculated from a snapshot, but velocity can be determined only by calculating temporal statistics (metrics) using the history of changes to a network (i.e. an animation of the network over discrete time intervals). This idea is explained using the diagrams below. First consider trying to predict links from a single graph. In the diagram below assume that a link that we would like to predict is forming between the orange nodes. We can see that the orange nodes have neither the highest degrees, the shortest path lengths or the highest number of common neighbours. Thus we might guess a link is forming between them, but we would not have much confidence in our prediction.

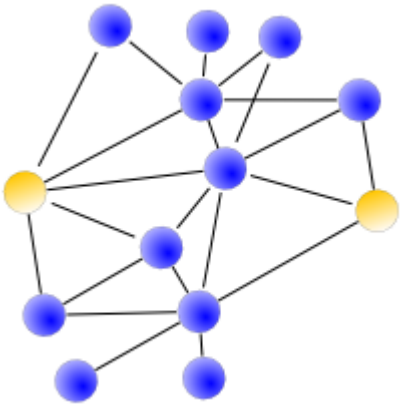

Figure 20. Prediction from a single sociogram

However, now consider the same graph shown as part of a temporal sequence of four time steps. This is illustrated below.

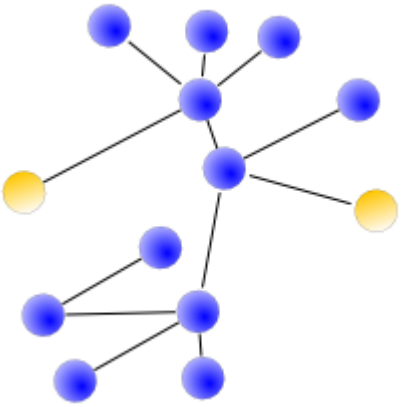

Figure 21. Time step one of four

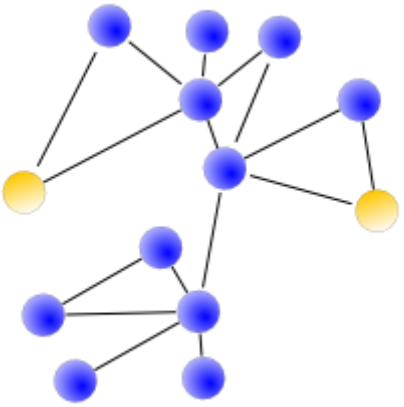

Figure 22. Time step two of four

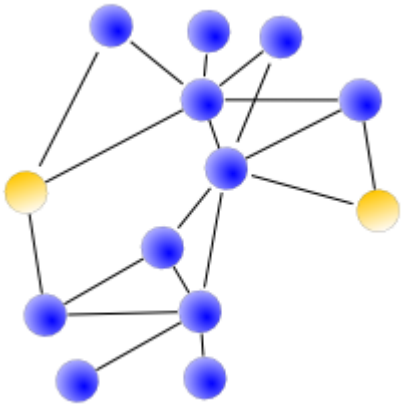

Figure 23. Time step three of four

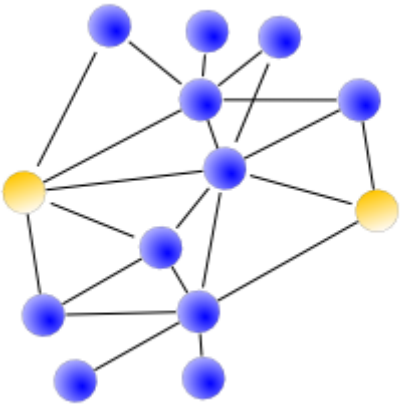

Figure 24. Time step four of four

These graphs provide far more information than the final time step alone. It is now apparent that orange nodes are highly likely to form a new link. We can see that these two nodes have been far more active over the past three time steps than any of the other nodes. Their degree has a higher daily average increase than any of the other nodes. The overall increase in degree from the first to the last time step has also been far greater than the other nodes. If we examined how many messages were exchanged between various nodes we might find that the orange nodes were also far more active communicators with existing neighbours than other nodes. In this chapter we describe ways of measuring these ideas and testing for their actual usefulness in a real world network. Though these ideas seem both simple and powerful it is easy to see how they could be overlooked by researchers to date. Ignoring the temporal information inherent in a sequence of social networks was due to the lack of data availability until recently, as explained in chapter two. Researchers were not interested in, and did not have available, large sequences of networks. This situation has changed and needs to be addressed by modern social network analysis practitioners. The obvious question now is: "How do we quantify temporal behaviour and trends into relevant metrics or statistics that can be used for link analysis?" Though there are many possibilities, in this research a few temporal statistics were defined that seemed to represent a broad selection of ways to quantify temporal trends.

Before the definitions are given, it must be pointed out that some classification systems do use temporal information, but not in the sense we mean it here. For instance, dynamic Bayesian networks include time steps as another set of nodes and links in the graph structure of their classification system [48]. This allows the system to mine temporal relationships – but only for the metrics it has been given to observe. Such a network can be used to perform tasks such as predicting links in a certain number of future time steps (rather than just the next time step). However it cannot discern temporal information from the sociogram structure other than by temporally analysing traditional (static) social

network metrics it has in its given evidence data set.

5.2. Temporal statistics

In order to illustrate how existing temporal statistical ideas can be incorporated into the study of link prediction we use finance as an analogy. Finance is a useful analogy as the study of financial markets has led to helpful ways of incorporating trends for prediction. One of the most basic trend quantifications is the concept of return. Return is the percentage increase or decrease of a value over a period of time [61]. For instance, if we are considering the degree of node v_i from time step one to time step fifty, the degree return would be $\frac{degree_{50}(v_i) - degree_1(v_i)}{degree_1(v_i)}$. Return is used in finance to calculate how much a share has increased in value over the time an investor has held it. In other words it can be a measure of profit or loss. In a social network, return quantifies how the metric value of a node, dyad or graph has increased or decreased over time. For example, it allows us to compare the rates of decrease of distance of two dyads over time. We might expect that the dyad that is getting closer faster would be more likely of forming a link first. Similarly, a node with a very low degree return is unlikely to form new links in the present.

Finance also uses moving averages to extract long-term trends from short term noise [61]. A moving average is the average of the values of a metric calculated for every time step surrounding a point. For instance, if we are considering the degree of node v_i from time step one to time step fifty, the degree average would be $\frac{\sum_{t=1}^{t=50} degree_t(v_i)}{50}$.

Social network researchers also seemed to have ignored the timing of communications between nodes. Although they have defined the concept of strong and weak ties (often representing how many times a pair of nodes has communicated), they have not focused much on the frequency of communications. Frequency might also be an important indication of trends. Thus the recency of two nodes is defined as one plus the number of time steps that have elapsed since they last communicated (which can be defined in any of the four possible combinations of directed and bidirected ways). Recency can also be defined for a single node, and is the number of time steps elapsed since the node last communicated. One is added to the number of elapsed time steps so that if the node communicated in the current time step (giving an elapsed value of zero) its recency will be one and will have an effect

on the statistical process. These ideas were first proposed in an earlier form in two articles stemming from the research described in this dissertation [5][6].

5.3. Hypothesis statement

The null hypothesis of this experiment is that using temporal metrics in addition to static metrics will not increase the accuracy of link prediction. More mathematically, the null hypothesis is that the kappa value of a regression performed using temporal and traditional metrics will be equal to the kappa value of a regression performed using traditional metrics.

5.4. Methodology

This section describes the experiment undertaken to find if including temporal information in addition to static metric values can enhance link prediction. It follows the general methodology described in the research methodology chapter. As this is a totally new area of research I was forced to invent temporal statistics that may or may not embody useful temporal information. This implies that if these statistics are found to be useful then temporal analysis warrants further investigation. However if the experiment finds no extra value in the temporal statistics it may simply be because the statistics defined were unhelpful, and not that temporal analysis is useless in itself. The experiment conducted using these statistics is explained below. Sample metric values were taken for 9939 instances per class for two classes over 100 time steps. The y-variable chosen equalled zero if two nodes under consideration were unconnected and equalled one if the nodes had a forming link. Thus the negative case is the unconnected class and the positive case is the link forming class. The x-variables used include the dyadic metrics past research has shown to be best predictors of links. The monadic metrics used were degree and recency, which were calculated for both nodes in the dyad. The dyadic metrics used were: the Katz measure, preferential attachment, common neighbours and the Adamic\Adar number. In addition to these static metrics their temporal variants were also stored as metrics in the sample data set. For every x-variable listed above, the return and the average were calculated for the last twenty time steps, the last ten time steps and the last two time steps. This range of time steps was chosen to see if the size of the window of observations makes any difference to the usefulness of the temporal statistics. All metrics were calculated using bidirected links. If a metric could not be calculated for a given instance it was classed as a missing value. This happens frequently when computing temporal metrics as a node is often missing in a previous time step. This occurs when a node is chosen as part of a sample in the current time step but only joined the network five time steps ago. Thus when calculating the temporal average for ten time steps in the past the metric will have a value of “NaN (“not a number”in Java). Missing values can be handled by most

learning systems, including Weka.

The column headings for the data set calculated are:
DegreeF, DegreeT, RecF, RecT, Katz, PA, CN, AA,
DegreeFR20, DegreeFR10, DegreeFR2, DegreeTR20, DegreeTR10, DegreeTR2, KatzR20, KatzR10, KatzR2, PAR20, PAR10, PAR2, CNR20, CNR10, CNR2, AAR20, AAR10, AAR2, RecFR20, RecFR10, RecFR2, RecTR20, RecTR10, RecTR2,
DegreeFA20, DegreeFA10, DegreeFA2, DegreeTA20, DegreeTA10, DegreeTA2, KatzA20, KatzA10, KatzA2, PAA20, PAA10, PAA2, CNA20, CNA10, CNA2, AAA20, AAA10, AAA2, RecFA20, RecFA10, RecFA2, RecTA20, RecTA10, RecTA2.

The suffix *F* stands for From (the first node in a dyad), *T* stands for To (the second node in a dyad), *R20* stands for return for the past twenty time steps, *R10* stands for return for the past ten time steps, and *R2* stands for return for the past two time steps, *A20* stands for average for the past twenty time steps, *A10* stands for average for the past ten time steps, and *A2* stands for average for the past two time steps. The temporal values for all the metric types were calculated by loading the sociograms for the past twenty time steps into memory and calculating the metrics needed for the required nodes. These values were stored in an array so that temporal statistics such as average and return could be calculated for them. Thus calculating temporal metrics is approximately twenty times slower than calculating static metrics, as the same metrics have to be calculated for twenty time steps rather than just one.

5.5. Results

This results section presents both the statistics of the metrics calculated and their predictive accuracy individually and in sets. These results are presented in three tables. The first table, 11, shows the statistical means test for both classes using every metric. The second table, 12, shows detailed results of the accuracy attained by logistic regression using each metric individually. The third and last table, 13, shows shows the accuracy attained using sets of metrics.

Each row in the table below gives the statistics associated with a certain metric. The metrics are grouped into all the variations of the base statistics. The thick black horizontal lines delineate the different metric groupings. The first column in the table gives the name of the metric. The second and third columns give the average of the values for the metric for the unconnected- and forming dyads respectively. The fourth and fifth columns give the standard deviation of the values for the metric for the unconnected- and forming dyads respectively. The mean difference column shows the

value of the unconnected mean minus the forming mean. Thus, it is negative when metrics associated with a forming dyad are larger than those associated with an unconnected dyad. The test statistic column gives the value of the Normal distribution test statistic, calculated using the mean and standard deviation values as described in the research methodology chapter. The subsequent column gives the significance level associated with the test statistic. The final column shows the kappa statistic for the metric, as given by Weka, using logistic regression. It is not related to the standard hypothesis testing statistics given in the previous column, but allows the reader to compare the general usefulness of each temporal variation of the statistic.

Table 11. Metric statistics, grouped by category

Metric	Unconnected mean	Forming mean	Unconnected standard deviation	Forming standard deviation	Mean Difference	Test statistic	Significance level	Kappa
DegreeFA20	1.9438	8.6818	3.5317	21.5584	-6.7379	-30.75	0.0001	35.48%
DegreeFA10	1.9152	8.3771	3.5181	22.4411	-6.4618	-28.36	0.0001	35.12%
DegreeFA2	1.8927	7.9833	3.5052	22.1150	-6.0906	-27.12	0.0001	29.85%
DegreeF	1.8876	7.8111	3.5006	21.9536	-5.9235	-26.56	0.0001	27.13%
DegreeFR20	0.1361	0.7550	0.5419	2.2471	-0.6189	-26.69	0.0001	10.26%
DegreeFR10	0.0626	0.4599	0.3011	1.8352	-0.3973	-21.3	0.0001	5.86%
DegreeFR2	0.0089	0.0505	0.0952	0.2291	-0.0417	-16.74	0.0001	-8.33%
DegreeTA20	1.9810	8.0030	3.4860	19.1849	-6.0220	-30.79	0.0001	42.46%
DegreeTA10	1.9606	8.0086	3.5178	20.3095	-6.0480	-29.25	0.0001	41.54%
DegreeTA2	1.9256	7.5180	3.5184	20.7846	-5.5923	-26.45	0.0001	32.30%
DegreeT	1.9223	7.0698	3.5223	20.2884	-5.1475	-24.92	0.0001	26.47%
DegreeTR20	0.1426	0.6271	0.5724	1.6312	-0.4844	-27.94	0.0001	15.68%
DegreeTR10	0.0658	0.4169	0.2848	1.5329	-0.3510	-22.45	0.0001	12.97%
DegreeTR2	0.0066	0.0795	0.0734	0.3285	-0.0729	-21.59	0.0001	-2.55%
KatzA20	0.0111	0.0427	0.0225	0.0473	-0.0316	-60.08	0.0001	41.78%
KatzA10	0.0110	0.0415	0.0229	0.0475	-0.0305	-57.7	0.0001	44.92%
KatzA2	0.0107	0.0373	0.0227	0.0471	-0.0267	-50.8	0.0001	37.32%
Katz	0.0106	0.0348	0.0227	0.0467	-0.0241	-46.31	0.0001	28.17%
KatzR20	0.2177	0.5243	0.6967	1.4813	-0.3065	-18.67	0.0001	-2%
KatzR10	0.1344	0.3634	0.6071	1.2972	-0.2290	-15.94	0.0001	-7%
KatzR2	0.0094	0.0846	0.1194	0.6620	-0.0752	-11.15	0.0001	-23.1%
PAA20	3.8428	38.6423	17.7252	97.9615	-34.7995	-34.85	0.0001	45.46%
PAA10	3.7283	37.9504	17.0146	106.8425	-34.2221	-31.54	0.0001	48.36%
PAA2	3.6188	35.7246	16.5470	117.9152	-32.1059	-26.88	0.0001	41.61%
PA	3.6043	33.2047	16.4837	114.0258	-29.6005	-25.61	0.0001	33.27%
PAR20	0.2813	1.6457	0.8481	4.4977	-1.3645	-29.72	0.0001	5.31%
PAR10	0.1404	1.0239	0.5073	3.2094	-0.8835	-27.11	0.0001	1.36%
PAR2	0.0151	0.1359	0.1213	0.4344	-0.1209	-26.72	0.0001	-10.6%
CNA20	0.0015	0.0306	0.0477	0.2007	-0.0290	-14.02	0.0001	32.2%
CNA10	0.0019	0.0302	0.0507	0.2012	-0.0283	-13.58	0.0001	29.17%
CNA2	0.0019	0.0253	0.0497	0.1865	-0.0234	-12.1	0.0001	12.52%
CN	0.0019	0.0231	0.0501	0.1791	-0.0212	-11.38	0.0001	1.92%
CNR20	0.0000	0.0920	0.0000	0.3282	-0.0920	-27.93	0.0001	-0.87%

Metric	Unconnected mean	Forming mean	Unconnected standard deviation	Forming standard deviation	Mean Difference	Test statistic	Significance level	Kappa
CNR10	0.0000	0.0794	0.0000	0.2639	-0.0794	-29.98	0.0001	-1.18%
CNR2	0.0000	0.0112	0.0000	0.1054	-0.0112	-10.57	0.0001	-0.00%
AAA20	0.0015	0.0442	0.0656	0.3151	-0.0427	-13.22	0.0001	32.2%
AAA10	0.0019	0.0417	0.0656	0.2989	-0.0399	-12.99	0.0001	29.17%
AAA2	0.0017	0.0357	0.0619	0.2785	-0.0340	-11.87	0.0001	12.52%
AA	0.0017	0.0329	0.0615	0.2701	-0.0311	-11.21	0.0001	1.92%
AAR20	-0.0393	-0.0091	0.0481	0.2333	-0.0301	-12.62	0.0001	-0.40%
AAR10	-0.0163	0.0539	0.0376	0.2920	-0.0702	-23.78	0.0001	-1.18%
AAR2	-0.0092	0.0066	0.0346	0.1109	-0.0158	-13.59	0.0001	0.07%
RecFA20	38.1636	12.5866	30.5582	16.1022	25.5771	73.82	0.0001	28.38%
RecFA10	38.2149	11.9455	32.0390	16.3050	26.2694	72.85	0.0001	35.75%
RecFA2	38.7039	11.0723	33.3012	16.5778	27.6316	74.05	0.0001	44.52%
RecF	38.7928	10.8518	33.4656	16.6218	27.9410	74.55	0.0001	46.34%
RecFR20	1.5342	1.4653	3.1617	3.5274	0.0688	1.45	NS	-2.92%
RecFR10	0.7021	0.8002	1.5502	1.9719	-0.0981	-3.9	0.0001	1.82%
RecFR2	0.0814	0.1413	0.2480	0.4357	-0.0599	-11.91	0.0001	8.46%
RecTA20	37.3356	15.1974	30.3941	18.7683	22.1382	61.78	0.0001	14.80%
RecTA10	37.5869	14.0277	31.8041	18.7829	23.5592	63.59	0.0001	22.61%
RecTA2	37.7987	11.8050	33.1000	18.2863	25.9937	68.53	0.0001	37.67%
RecT	37.8514	11.1834	33.2870	18.0862	26.6681	70.18	0.0001	42.41%
RecTR20	1.5430	1.5128	3.1033	3.5408	0.0302	0.64	NS	0.00%
RecTR10	0.6878	0.8434	1.5252	2.0300	-0.1556	-6.11	0.0001	8.01%
RecTR2	0.0797	0.1669	0.2564	0.4539	-0.0872	-16.67	0.0001	12.87%

The table below lists each metric individually in separate rows and describes their usefulness (contribution to accuracy) in a logistic regression. The first column shows the kappa statistic, which is a measure of how much more accurate a prediction is, compared to a random prediction. If the value is negative it means that using the given metric gives a prediction model that is even worse than guessing randomly. The metrics are ranked in descending order of their kappa value. The total accuracy can be described in different ways. The last column shows the overall accuracy of the regression. This value can be deceptive and is not as useful as the kappa statistic. The third and fourth columns are important and show the true positive rate for each class.

Table 12. Metric predictive accuracy, ranked by kappa

Metric	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
PAA10	48.36%	80.0%	68.3%	74.2119%
RecF	46.34%	73.4%	72.9%	73.1724%
PAA20	45.46%	67.2%	78.3%	72.7029%
KatzA10	44.92%	70.2%	74.7%	72.4514%
RecFA2	44.52%	73.2%	71.4%	72.2669%
DegreeTA20	42.46%	70.7%	71.8%	71.2274%
RecT	42.41%	71.9%	70.5%	71.2106%
KatzA20	41.78%	58.7%	83.2%	70.8249%

Metric	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
PAA2	41.61%	89.0%	52.4%	70.9088%
DegreeTA10	41.54%	77.5%	64.0%	70.8082%
RecTA2	37.67%	70.5%	67.2%	68.8464%
KatzA2	37.32%	79.9%	57.3%	68.7290%
RecFA10	35.75%	72.9%	62.8%	67.9074%
DegreeFA20	35.48%	70.6%	64.8%	67.7565%
DegreeFA10	35.12%	77.5%	57.6%	67.6224%
PA	33.27%	90.3%	42.9%	66.7840%
DegreeTA2	32.30%	75.9%	56.3%	66.2140%
AAA20	32.20%	73.2%	58.9%	66.1469%
CNA20	32.20%	73.2%	58.9%	66.1469%
DegreeFA2	29.85%	82.4%	47.3%	65.0402%
CNA10	29.17%	87.2%	41.9%	64.7384%
AAA10	29.17%	87.2%	41.9%	64.7384%
RecFA20	28.38%	75.6%	52.7%	64.2689%
Katz	28.17%	81.1%	47.0%	64.2019%
DegreeF	27.13%	83.1%	44.0%	63.6989%
DegreeT	26.47%	76.5%	49.9%	63.3300%
RecTA10	22.61%	69.6%	53.0%	61.3682%
DegreeTR20	15.68%	45.7%	70.0%	57.7465%
RecTA20	14.80%	71.6%	43.2%	57.5117%
DegreeTR10	12.97%	52.6%	60.4%	56.4554%
RecTR2	12.87%	57.0%	55.8%	56.4386%
CNA2	12.52%	98.5%	13.9%	56.6063%
AAA2	12.52%	98.5%	13.9%	56.6063%
DegreeFR20	10.26%	47.1%	63.1%	55.0637%
RecFR2	8.46%	55.5%	52.9%	54.2421%
RecTR10	8.01%	48.5%	59.5%	53.9571%
DegreeFR10	5.86%	53.4%	52.5%	52.9343%
PAR20	5.31%	23.6%	81.8%	52.3977%
CN	1.92%	99.8%	2.10%	51.4085%
AA	1.92%	99.8%	2.10%	51.4085%
RecFR10	1.82%	44.9%	56.9%	50.8551%
PAR10	1.36%	30.0%	71.4%	50.4863%
AAR2	0.07%	0.0%	99.9%	49.4970%
DegreeTR2	-2.55%	60.8%	36.7%	48.8431%
RecTR20	0.00%	0.0%	100.0%	49.5305%
CNR2	0.00%	0.0%	100.0%	49.5305%
AAR20	-0.400%	0.0%	99.6%	49.3293%
CNR20	-0.87%	0.10%	99.0%	49.0946%
CNR10	-1.18%	0.2%	98.6%	48.9437%
AAR10	-1.18%	0.2%	98.6%	48.9437%
RecFR20	-2.92%	8.9%	88.2%	48.1556%
KatzR20	-7.56%	22.1%	70.3%	45.9759%
KatzR10	-7.56%	22.1%	70.3%	45.9759%
DegreeFR2	-8.33%	59.6%	32.1%	45.9759%
PAR2	-10.61%	35.5%	53.9%	44.6009%
KatzR2	-23.15%	27.5%	49.3%	38.2964%

The graph below shows how the kappa of each metric increases as we take moving averages over

different numbers of time steps prior to the current one for the given metric.

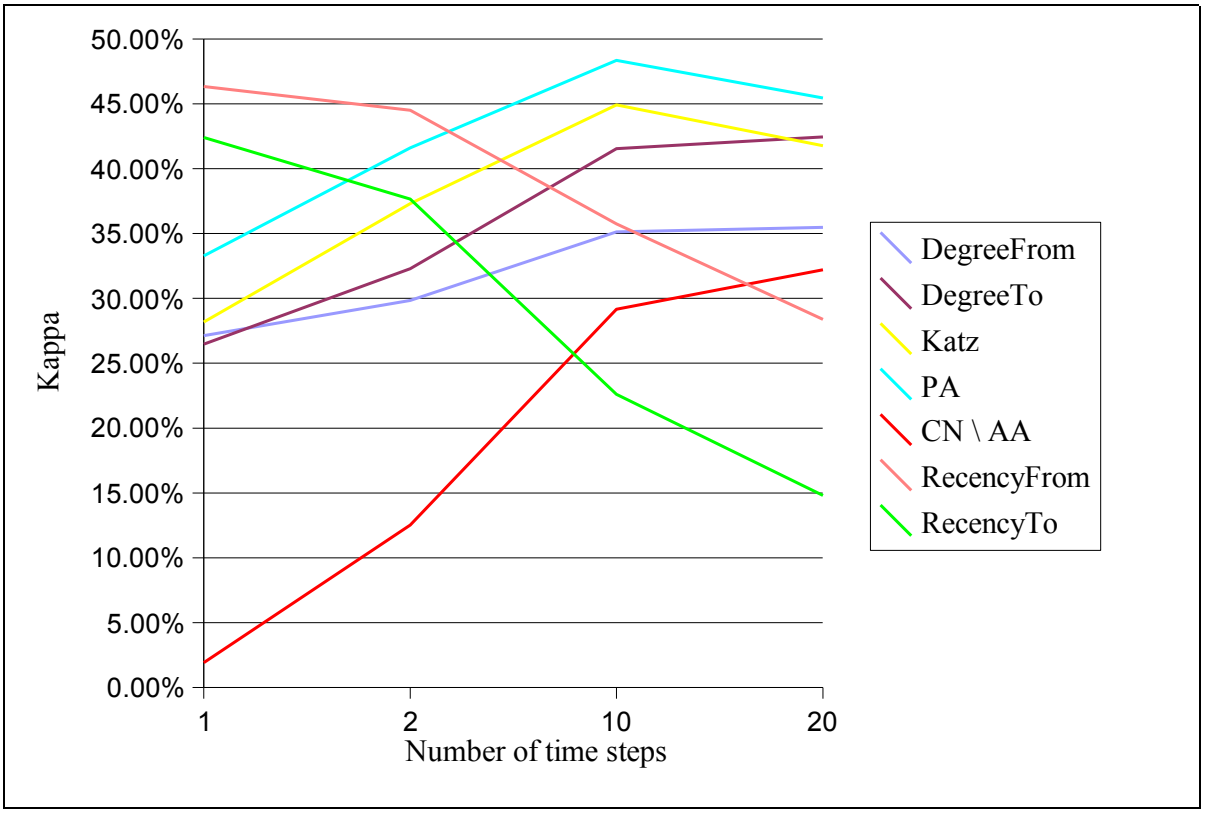

Figure 25. Increase in kappa per time step for average metrics

The table below uses the same columns as the one above, but instead of showing the accuracy of each metric individually it shows the accuracy of regressions performed with different sets of metrics. The metric subset column describes the type of metrics used in italics and then lists all the metrics in the set.

Table 13. Metrics set predictive accuracy

Metric subset	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
Static metrics: DegreeF, DegreeT, Katz, PA, CN, AA	39.83%	79.8%	59.9%	69.9698%
Static metrics with average 10: DegreeF, DegreeT, Katz, PA, CN, AA, DegreeFA10, DegreeTA10, KatzA10, PAA10, CNA10, AAA10	51.13%	80.4%	70.1%	75.5869%
Static metrics with average 20: DegreeF, DegreeT, Katz, PA, CN, AA, DegreeFA20, DegreeTA20, KatzA20, PAA20, CNA20, AAA20	53.62%	80.6%	73.0%	76.8276%

Metric subset	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
Static metrics with average 10 and recency: DegreeF, DegreeT, Katz, PA, CN, AA, RecF, RecT, DegreeFA10, DegreeTA10, KatzA10, PAA10, CNA10, AAA10	63.62%	81.3%	82.4%	81.8075%
Metrics with highest individual accuracy: RecF, RecT, DegreeFA20, DegreeTA20, PAA10, KatzA10	63.59%	80.3%	83.3%	81.7907%
Weka's logistic subset classifier best first attribute selection search: RecF, RecT, CN, AA, DegreeFA20, DegreeFA10, DegreeTA20, DegreeTA2, KatzA10, PAA10, PAA2, RecFA10	64.02%	81.2%	82.8%	82.0087%

5.6. Conclusions

This section draws conclusions from the results presented in the three tables in the previous section. The first table, Table 11, shows that there is a highly significant statistical difference between the two classes for all the metrics except for a few of the recency returns. Some metrics that are significantly

different from zero are not useful for distinguishing the two classes. This can be seen by the low kappa values for some of the metrics with high test statistics, such as common neighbours. However, the metrics with better predictive power than their sibling variants do indeed have higher test statistics than them. Thus the kappa statistic is a better indicator of usefulness for these metrics than their test statistic. As expected, we can see that the unconnected means and standard deviations are far lower than the forming means and standard deviations. This is because the metric values of unconnected nodes are likely to be very low due to the low number of common neighbours and long distances between the nodes. Grouping metrics according to their base type (e.g. Katz, for metrics like KatzA20, KatzR2, etcetera) allows us to see how useful that type of metric is overall. It also allows us to detect which, if any, temporal variants of a metric are the most useful.

Firstly, we can see that for all the metrics either their base type or one of their temporal variants had kappa values indicating that the metric is useful for prediction. Huang et al found the Katz measure was the most useful, followed by preferential attachment, common neighbours and the Adamic\Adar measure [35]. This research is similar, finding that the most useful static metrics are: preferential attachment (kappa of 33%), the Katz measure (kappa of 28%), the from degree (kappa of 27%) and the to degree (kappa of 26%). The following metrics were found not to be useful: common neighbours (kappa of 2%) and Adamic\Adar number (kappa of 2%). The temporal metrics paint a slightly different picture. The most useful metrics overall (including all base metrics and their temporal variants) are: the preferential attachment moving average over ten time steps (kappa of 48%), the from recency (kappa of 46%), the Katz measure moving average over ten time steps (kappa of 45%), the to degree moving average over twenty time steps (kappa of 42%), the to recency (kappa of 42%), the from degree moving average over twenty time steps (kappa of 35%), the common neighbours moving average over twenty time steps (kappa of 32%), and the Adamic\Adar measure moving average over twenty time steps (kappa of 32%). It is interesting to see that the simple common neighbours metric performs just as well as the more complex Adamic\Adar measure.

Secondly, we notice that the kappa of a metric can increase tremendously from one variant to another. For instance, the common neighbours metric was found to be no better than a random guess (with a kappa of only 2%). However its average over the past twenty time steps was found to have a kappa of 32%. There is a pattern that the moving average of a metric is more useful than the metric calculated at a point. Furthermore, the increase in accuracy of the moving average of a metric appears to approach zero as the time step span of the average reaches twenty time steps. We can observe this also in the graph of the kappa values over each number of time steps. This indicates that we need to calculate useful moving averages only between ten and twenty time steps prior to the current time step. Thirdly, the return of a metric over any time range was found to decrease the kappa of a metric.

This indicates that returns are not useful additions to the link prediction toolbox. Both the return and the average of the recency metric decrease the metric's kappa, indicating that we need never calculate temporal variants of the recency metric. This is unsurprising as the recency metric itself is a temporal metric.

The second table, Table 12, shows not only the kappa value of a given metric, but also its contribution to the true positive rate of both classes. All the metrics above preferential attachment are useful (with a kappa of at least 35%). Preferential attachment is the first metric in the list to have a true positive rate of less than 50% for either class. All the useful metrics (those above preferential attachment) generally classify over 60% of both classes correctly.

The third table, Table 13, shows us the predictive power of groups of metrics. This is the most important test of temporal metrics. Even though temporal metrics may have higher individual kappa values than their static counterparts they may not be more useful when used in combination with other metrics. Thus the third table shows us the contribution of temporal metrics to link prediction where it matters most: choosing a set of metrics that provides for the highest possible prediction accuracy. To start with we examine the predictive power of the standard static metrics that would be used by a traditional social network analysis practitioner: DegreeF, DegreeT, Katz, PA, CN and AA. This set has a kappa of 40%, with a true positive rate for forming dyads of only 60%. If we include the moving averages of the metrics the kappa rises to 51%, with both true positive rates above 70%. Using the moving average over twenty time steps rather than ten increases the kappa by just over a percent. This minute increase in accuracy, compared to the large increase from using just the static metrics (effectively an average over one time step), indicates that taking an average over more than twenty time steps would be of little use. The sets we have discussed so far prove that using temporal metrics (both returns and averages) have led to a large improvement in link prediction. Now we address the question: “What is the maximum accuracy we can attain using any combination of metrics?” Using a set comprising the six metric variants with the highest rated kappas from the second table we attain a kappa of 64%, with an overall accuracy 82%. The last metric set used was obtained by running an attribute selection process using a subset logistic regression classifier search in Weka. There was a marginal increase in accuracy to an overall accuracy of 82%. Overall, we can say that the null hypothesis was rejected.

To summarise, it has been found that:

- Static metrics that are not useful for prediction for a given data set, may become more useful when converted to a temporal variant of the metric.
- Metrics that are not useful individually may become useful when used in combination with

other metrics in a set.

- A metric's moving average is more useful than the static metric alone (except in the case of the recency metric). The increase in accuracy of the moving average of a metric appears to approach zero as the time step span of the average reaches twenty time steps.
- The most useful individual metrics (including all base metrics and their temporal metrics) are: the preferential attachment moving average over ten time steps (kappa of 48%), the from recency (kappa of 46%), the Katz measure moving average over ten time steps (kappa of 45%), the to degree moving average over twenty time steps (kappa of 42%), the to recency (kappa of 42%), the from degree moving average over twenty time steps (kappa of 35%), the common neighbours moving average over twenty time steps (kappa of 32%), and the Adamic\Adar measure moving average over twenty time steps (kappa of 32%).
- Metric returns are not useful for prediction.
- The recency metric is a useful new contribution to link prediction, but its temporal variants are useless and can be ignored.
- Using temporal metrics enhances link prediction significantly. The maximum accuracy attained using temporal metrics was 82% (compared to the static metric set's accuracy of 70%), with true positive rates of 81% and 83%, using the following metrics: RecF, RecT, CN, AA, DegreeFA20, DegreeFA10, DegreeTA20, DegreeTA2, KatzA10, PAA10, PAA2 and RecFA10.

Chapter 6. Local link analysis

This chapter describes an investigation into local metrics – metrics that are calculated in the local neighbourhood of a given node. Traditional metrics are calculated globally, i.e. calculated using every node in the graph. This investigation considers whether local metrics are as useful for link prediction as global metrics. The speeds of the types of metric calculations are also compared.

6.1. Global metrics deficiencies

As discussed in the background chapter, most link prediction research has been conducted on small graphs (ones that fit into adjacency matrices in a computer's random access memory). Similarly, the metrics used for prediction (and nearly every other social network application) have been computed globally. For instance, betweenness is computed using shortest paths found between every node in the graph. However, large networks (larger than a few thousand nodes, depending on the computer in question) cannot fit into matrices in a computer's memory as the space requirement increases in quadratic proportion to the number of nodes. Unfortunately, these large networks are the type that are most likely to be used for marketing, criminal analysis and epidemiological applications. They have to be stored in more efficient data structures, such as lists of linked nodes. The smaller space requirement for these lists is proportional only to the density of the network stored, since only linked nodes are stored and not unlinked pairs. This type of data structure makes the computation of shortest path-based metrics unacceptably slow. A new way of calculating metrics, or a new type of metric, is now required for real-world graphs. Thus, the introduction of local metrics is now proposed. A local metric is identical to its traditional counterpart, except that it is calculated using only the nodes within a small radius of the node, or nodes, in question. It is hoped that using fewer nodes in metric computations will be speed up the process to a level acceptable for real-world applications. Furthermore, computing common neighbour metrics at a radius of more than one might help solve what Liben-Nowell calls “the distance-three task”[41]. He states “...nodes separated by a graph distance of more than two have no neighbors in common ...and hence this ...rules out the use of methods based on common neighbors”. This problem is illustrated in the graphs below.

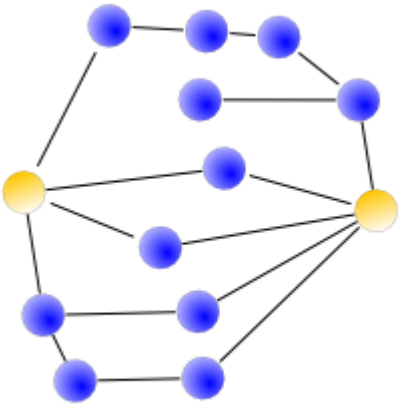

Figure 26. A forming link with two common neighbours at radius one

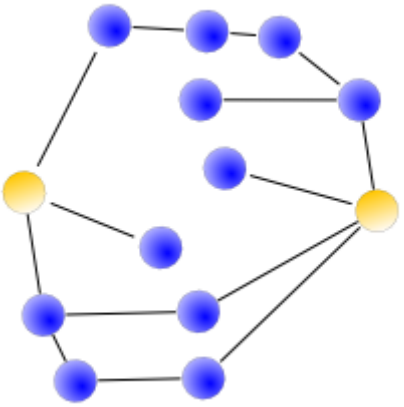

Figure 27. A forming link with no common neighbours at radii less than two

Suppose in the graph above on the left a link is forming between the orange nodes. Since this dyad has two common neighbours, i.e. it has a shortest path length of two, we might be able to predict the link. However the dyad in the graph on the right has no common neighbours, thus we would not be able to predict this link using common neighbour-based metrics. It is therefore an example of a distance three task. Instead of using common neighbours for prediction we would have to use the more computationally complex distance-based metrics.

6.2. Definition of local metrics

Before local metrics are defined the concept of neighbourhoods must first be explained. The following diagram is used in this explanation.

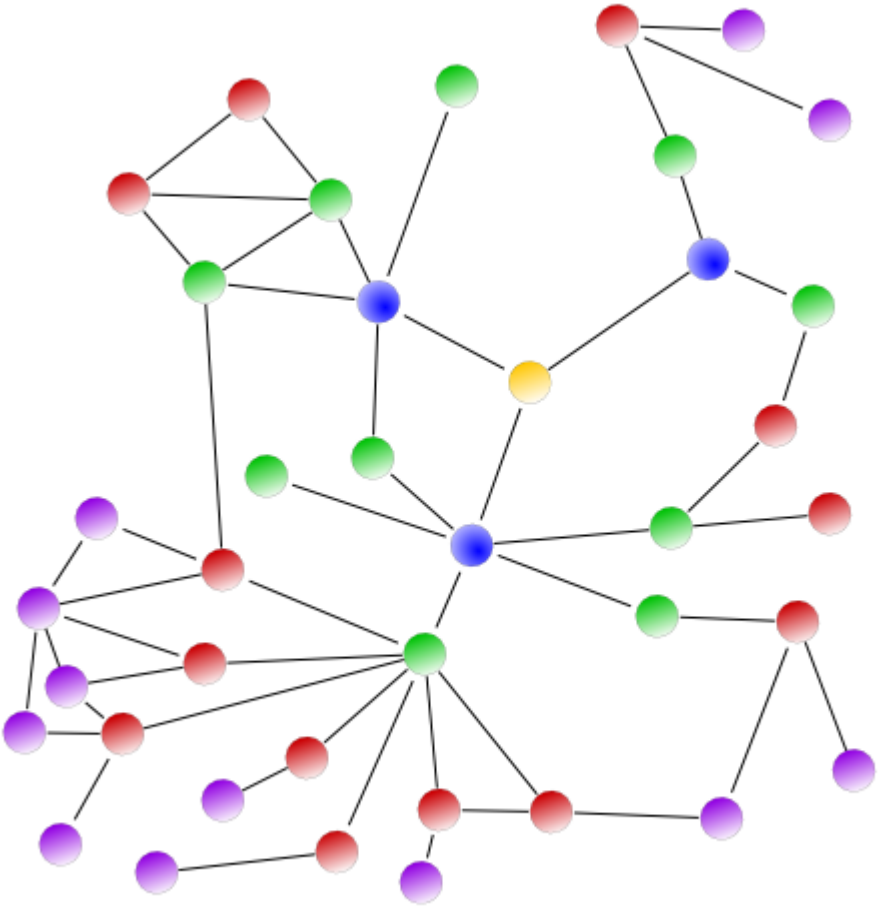

Figure 28. An egocentric subgraph of radius four centred on the orange node

The diagram above is a subgraph of a sociogram. It is therefore a sociogram in itself. It is a local egocentric sociogram for a radius of four centred on the orange node in the centre of the diagram. In other words it shows the focus node (drawn in orange) along with all nodes that are connected by at most four bidirected links. Nodes at a radius of one are coloured in blue, nodes at radius two are coloured in green, nodes at radius three are red and nodes at radius four are purple. The blue nodes therefore represent friends of the orange node, and the green nodes represent friends of friends. The orange (focus) node and the blue nodes can be called the orange node's neighbourhood. More specifically the nodes are the neighbourhood for a radius of one. The subgraph formed by the focus node, the blue nodes and the green nodes would be the node's neighbourhood at a radius of two. If the radius is not specified, we assume a neighbourhood is for a radius of one.

A local version of a metric has the original definition of its global counterpart, but is calculated as if the chosen neighbourhood was the entire graph. For instance, the radius three betweenness of the

node in the previous figure would be its betweenness as calculated for nodes only up to three links away from the focus node (i.e. the blue, green and red nodes). For a dyadic metric the local version is defined in the usual way except that is calculated using only the nodes in the neighbourhoods surrounding the two nodes in question. For instance, the radius three common neighbours of two nodes is calculated using the nodes within three links from the focus nodes. Thus if we return to figure 27 we can see the common neighbours value (at radius one) of the orange dyad is zero. However, the common neighbours value at radius two is four (the bottom four blue nodes). Given these intuitive examples the local definitions are now formally defined.

6.2.1. Distance-based monadic local metric definitions

Given a graph $G_n = \langle V_n, E_n \rangle$ at time step n and a monadic metric λ applied to a node v_i that is a member of the set of nodes of the graph, we traditionally calculate the metric using the formula $\lambda_{G_n}(v_i)$. In other words we calculate the local metric for the node v_i using the whole graph at time step n. If we wished to use a local metric at a radius of r instead the metric would be calculated as $\lambda_{\langle \{v_x : dist(v_i, v_x) \leq r\}, \{e_{v_m, v_n} : v_m \in \{v_x : dist(v_i, v_x) \leq r\} \wedge v_n \in \{v_x : dist(v_i, v_x) \leq r\}\} \rangle}(v_i)$. An equivalent version that might be easier to read is $\lambda_{\langle H, \{e_{v_m, v_n} : v_m \in H \wedge v_n \in H\} \rangle}(v_i)$, where $H = \{v_x : dist(v_i, v_x) \leq r\}$. Simply put, instead of using the entire graph G_n to calculate the metric we are using only the local egocentric subgraph (H) surrounding v_i up to a radius of r. In other words, we are using the subgraph formed by the nodes (the set $\{v_x\}$) that have a shortest path to v_i of at most length r ($dist(v_i, v_x) \leq r$,where r is a positive integer) and the nodes' incident links ($\{e_x\}$) where one of these links connects two nodes ($\{v_m, v_n\}$) and these nodes are in the set of nodes at most r links away from v_i ($v_m \in \{v_x : dist(v_i, v_x) \leq r\} \wedge v_m \in \{v_x : dist(v_i, v_x) \leq r\}$).

As an example consider the normal definition of betweenness, $\sum_{v_j \in V} \sum_{v_k \in V, v_k \neq v_j} \frac{\#(P(v_j, v_k, v_i))}{\#(P(v_j, v_k))}$, for the node v_i. If we were to calculate radius six betweenness (betweenness calculated using only the graph formed by the nodes up to six links away from the focus node) we would calculate $\sum_{v_j \in \{v_x : dist(v_i, v_x) \leq 6\}} \sum_{v_k \in \{v_x : dist(v_i, v_x) \leq 6\}, v_k \neq v_j} \frac{\#(P(v_j, v_k, v_i))}{\#(P(v_j, v_k))}$. As a graphical example, when calculating a monadic metric at radius of two for the orange nodes in the graph below we would use only the colour nodes in the graph and ignore all the grey ones.

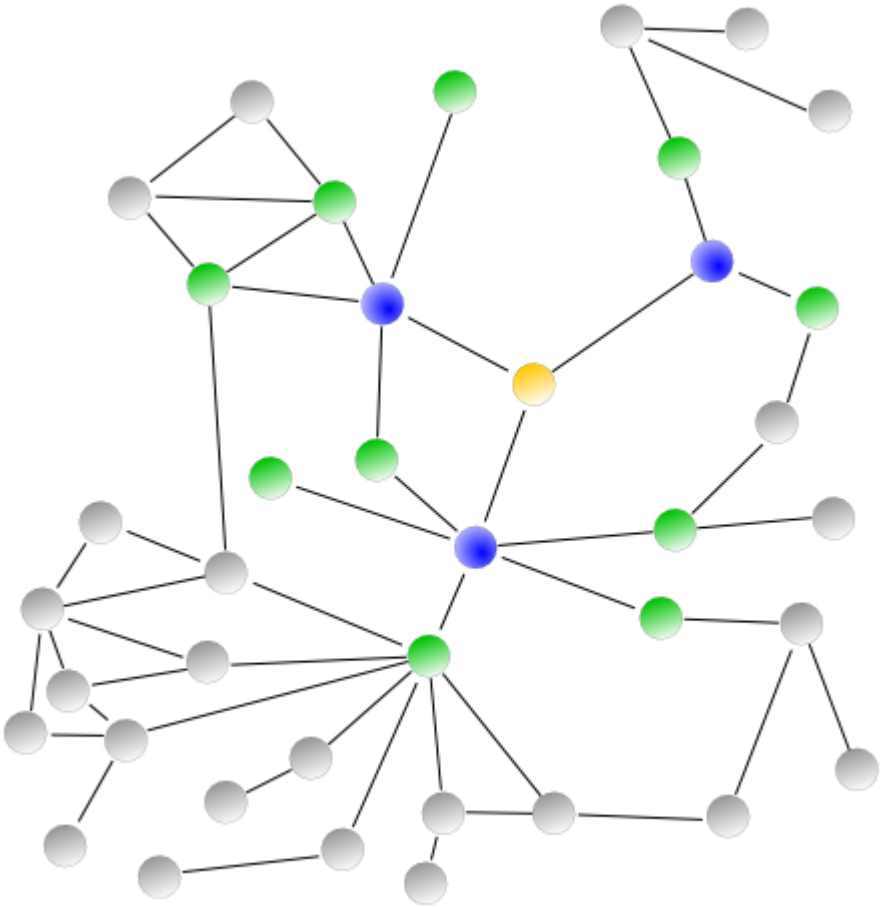

Figure 29. A local egocentric subgraph of radius two

6.2.2. Dyadic common neighbour-based local metric definitions

Given a graph $G_n = \langle V_n, E_n \rangle$ at time step n and a dyadic common neighbour-based metric λ applied to two nodes v_i and v_j that are a member of the set of nodes of the graph, we traditionally calculate the metric by defining common neighbours using the set of common neighbours, $\Gamma(v_i) \cap \Gamma(v_j)$, defined as $\{v_x : e_{v_i, v_x} \in E\} \cap \{v_x : e_{v_j, v_x} \in E\}$ or equivalently $\{v_x : e_{v_a, v_x} \in E \wedge dist(v_a, v_i) \leq 1\} \cap \{v_x : e_{v_b, v_x} \in E \wedge dist(v_b, v_j) \leq 1\}$. In other words we calculate the local metric for the nodes v_i and v_j using only common neighbours one link away from each of the focus nodes. If we wished to use a local metric at a radius of r instead, the metric would be calculated using the common neighbours set defined as $\{v_x : e_{v_a, v_x} \in E \wedge dist(v_a, v_i) \leq r\} \cap \{v_x : e_{v_b, v_x} \in E \wedge dist(v_b, v_j) \leq r\}$. Simply put we are looking at nodes not only immediately adjacent to the focus nodes, but also those

that are common to the focus nodes at distances greater than one. In other words we are defining $\Gamma_r(v_i)$ as all the neighbours of v_i up to r links away. Thus the traditionally used common neighbour definition, $\Gamma(v_i) \cap \Gamma(v_j)$,is equivalent to $\Gamma_1(v_i) \cap \Gamma_1(v_j)$ in this new system. So instead of calculating $\lambda_{\Gamma_1(v_i) \cap \Gamma_1(v_j)}(v_i, v_j)$ (i.e. calculating λ using common neighbours one link away) we now calculate $\lambda_{\Gamma_r(v_i) \cap \Gamma_r(v_j)}(v_i, v_j)$ (i.e. calculating λ using common neighbours r links away).

As an example consider the normal definition of Jaccard's coefficient, $\frac{\#(\{\Gamma(v_i) \cap \Gamma(v_j)\})}{\#(\{\Gamma(v_i) \cup \Gamma(v_j)\})}$. If we were to calculate to local metric version of Jaccard's coefficient for a radius of six we would calculate $\frac{\#(\{\Gamma_6(v_i) \cap \Gamma_6(v_j)\})}{\#(\{\Gamma_6(v_i) \cup \Gamma_6(v_j)\})}$. A graphical example is shown below. The following three graphs illustrate the number of common neighbours (shown in blue) of the focus nodes (shown in orange) at radius one, radius two and radius three.

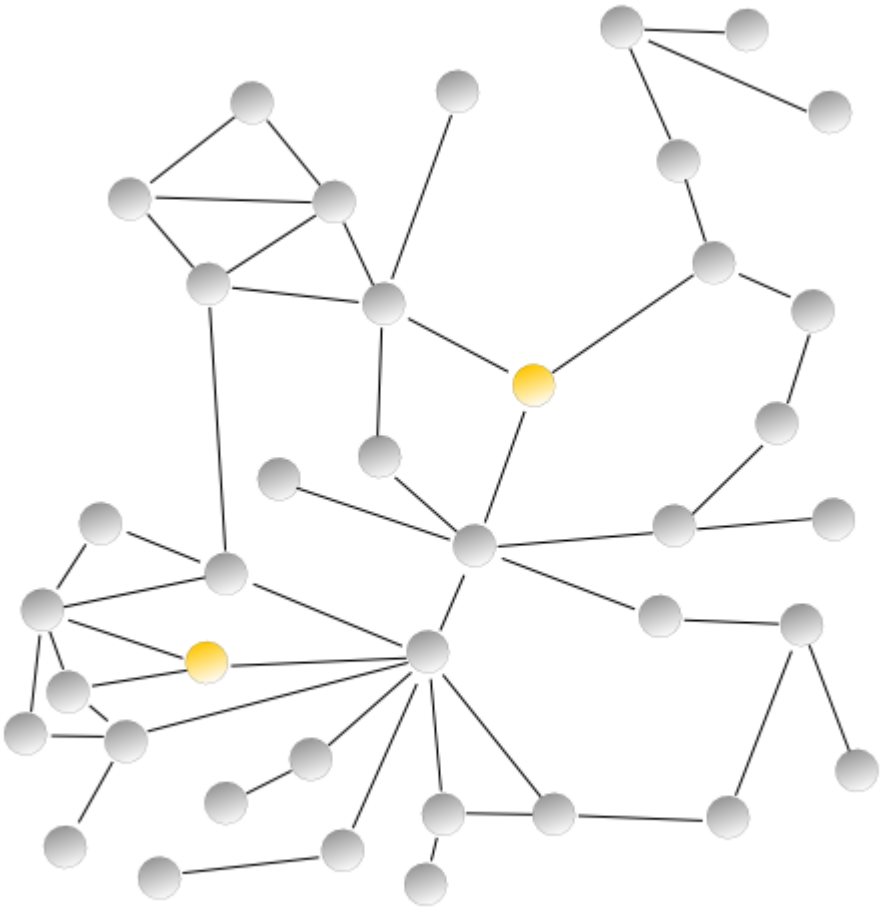

Figure 30. Radius one common neighbours

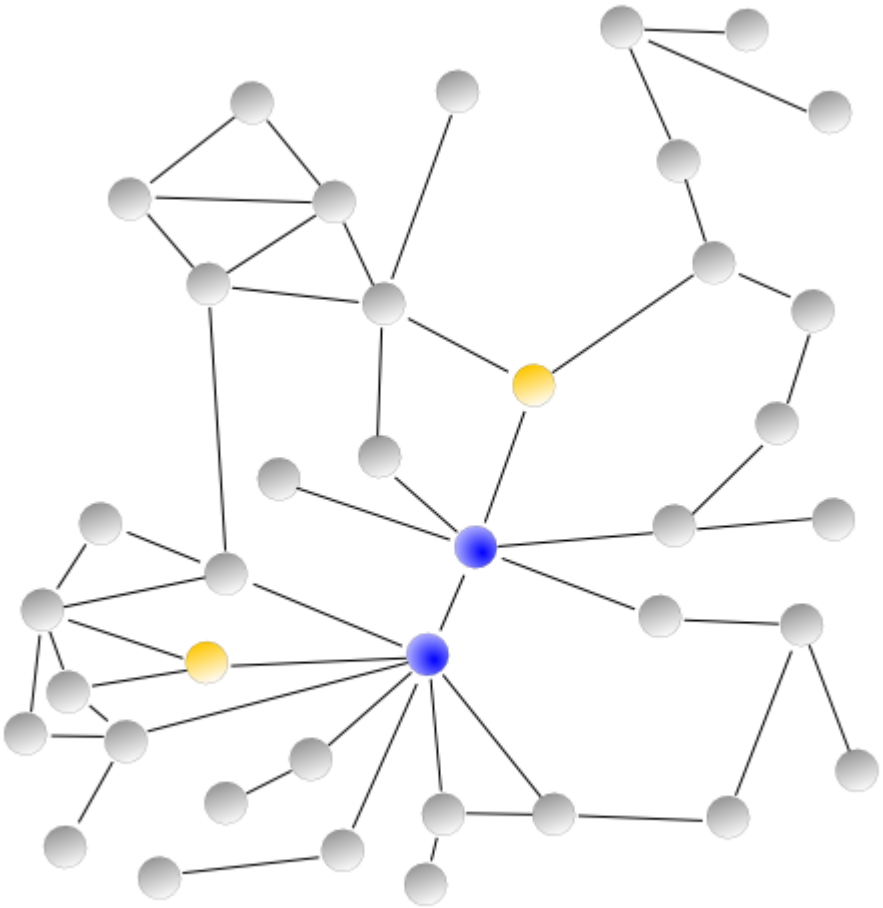

Figure 31. Radius two common neighbours

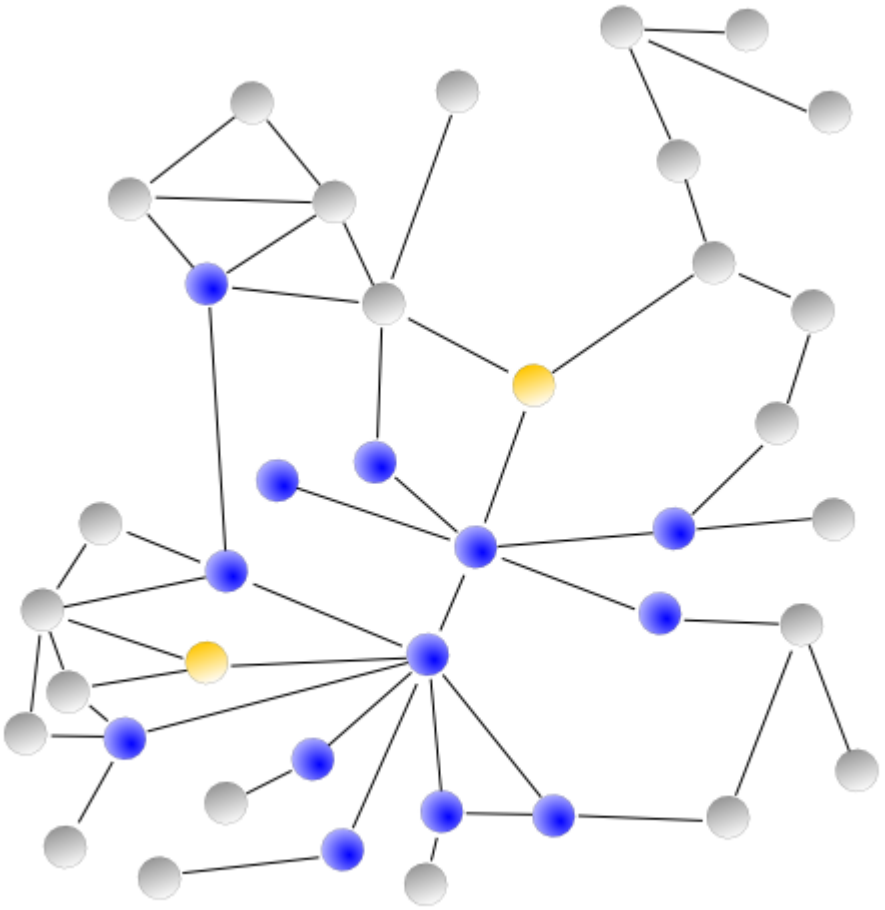

Figure 32. Radius three common neighbours

6.3. Computational complexity of local metrics

We now discuss how local metrics might compare to global metrics in terms of speed (computational complexity). Notice how the number of nodes in each neighbourhood of an egocentric graph increases exponentially as we consider larger radii. In general, if the mean degree of a node is d, the number of nodes at radius r will be $d(d-1)^{(r-1)}$ and therefore neighbourhood t (all the nodes up to t links away from the focus node) will contain $d\sum_{i=0}^{t-1}(d-1)^{i}$ nodes (excluding the focus node itself). This formula is for a tree structure – chosen to maximise the number of nodes at each radius and therefore consider the worst-case scenario for algorithm running time. An example of such a tree for a radius of two with an average degree of four is shown below.

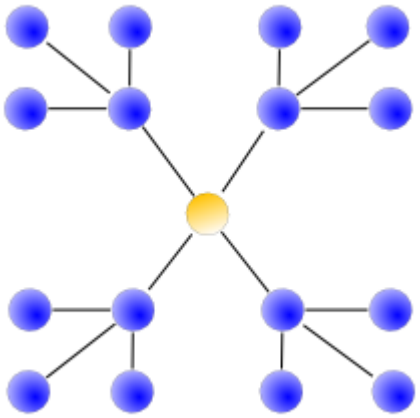

Figure 33. A tree centred on the orange node of radius two and average degree four

Neighbourhood zero (the nodes at a radius of zero) will always contain one node by definition. As an example, if people in a network usually have ten friends then the number of friends and friends of friends an individual has (including himself) will be ($10(9^0+9^1)$). We know Brande's algorithm runs in O(nm) time, where n is the number of nodes in a graph and m is the number of edges. The number of nodes in a neighbourhood is $d\sum_{i=0}^{t-1}(d-1)^i$. The number of links is also $d\sum_{i=0}^{t-1}(d-1)^i$, since in a tree there is one link for every node added. So nm is $(d\sum_{i=0}^{t-1}(d-1)^i)(d\sum_{i=0}^{t-1}(d-1)^i)=d^2(\sum_{i=0}^{t-1}(d-1)^i)^2$. d will remain roughly constant for a given graph, as it changes only gradually over time, and t is always constant as it is chosen before the calculation of a local metric. This means that the calculation of local betweenness will have a constant order for any given node (i.e. O(c)). Thus the order of the calculation of local betweenness for every node in the graph will be O(nc). It is therefore difficult theoretically to compare the speed of local metric calculation with the speed of traditional metric calculation. The larger a value for t that is chosen, the larger c will be in O(nc) and the slower the algorithm will be. Thus it may be true that local metric calculation is faster than traditional metric calculation for small values of t, but slower for large values. Also, the more connected a graph is, the closer local metric calculation becomes to global metric calculation, since a subgraph will contain almost as many nodes as the whole graph.

6.4. Expected usefulness of local metrics

It is expected that local metrics will be faster to compute than global metrics under certain conditions. However, we have to consider how much information useful for link prediction is being sacrificed for speed when discarding many of the nodes in the graph that may hold valuable paths or other structural information. Both the speed and information content of local and global metrics are compared in the experiment presented in this chapter. But let us first consider the theoretical implications of computing metrics in a local neighbourhood. Take for example the closeness centrality of a node. The local metric will probably have a far higher value than the global one because the local metric is calculated from a subgraph in which the focus node is highly central – it is after all a subgraph of the node's friends and their friends. At a radius of zero the node will be perfectly central. As the radius is successively increased the centrality of the node will begin to drop as more shortest paths are found between other nodes that do not include the focus node. Thus calculating the metric in the graph globally will tend to "drown out" the node's centrality amongst the multitude of connections that are available. As we use a larger radius for local calculation more and more connections become part of the graph that may bypass the focus node. Thus we can consider the radius we choose for metric calculation to be "tuning" the level of centrality nodes have. Using a local metric might highlight nodes that are highly central in their social neighbourhood – nodes that might have been ignored when using traditional metrics. Secondly, people tend to form new relationships with those socially close to them. In other words people make new friends with those who are friends with their friends (radius 2), or perhaps even with friends of friends of friends (radius 3). Thus for the purposes of link prediction especially (though hopefully not exclusively) it is likely that local metrics will be very informative.

Critics might point out that using local metrics for link prediction ignores that people form new relationships with those far outside of their local neighbourhood who may have similar interests. While this is true, since people with similar interests tend to form relationships they are likely to belong to their own interest subgraph. Local metrics might therefore not predict a link forming between a man who has just taken up a new sport and another sportsman whom he emails, but after the initial email they should be able to predict links between the same man and any other people who play the same sport, as they are now part of a local neighbourhood. This is, after all, the underlying assumption of all the common neighbour-based metrics, which have been found to be useful for prediction. Another fundamental advantage of local metrics is that they can be computed in parallel. In other words, if we wish to analyse a graph to predict forming links we can assign each node to a separate thread and allow the thread to calculate the given local metric. This is easier to accomplish with local metrics than with global ones as local metrics can be quickly calculated at small radii and

need less memory to store small local subgraphs.

6.5. Hypothesis statement

The null hypothesis of this experiment is that local metrics are no more useful for link prediction than global metrics and take an equal amount of time to compute. More mathematically, the null hypothesis is that the kappa value of a regression performed using local metrics will be equal to the kappa of a regression performed using global metrics. We would hope that using local metrics in addition to global metrics might yield additional information and hence lead to increased prediction accuracy.

6.6. Methodology

This section describes the experiment undertaken to find if calculating local metrics is faster than calculating global metrics, and whether they are useful for link prediction. It follows the general methodology described in the research methodology chapter. The two types of local metrics that are included are common neighbour-based dyadic metrics and monadic metrics. Two aspects of local and global metrics are evaluated, speed and usefulness. Sample metric values were taken for 9939 instances per class for two classes over 100 time steps. The y-variable chosen equalled one if the two nodes under consideration were forming a new link and equalled zero if the two nodes were unconnected. The monadic distance metric used was betweenness, which was calculated for both nodes in the dyad. The dyadic common neighbour-based metrics used were common neighbours, the Adamic\Adar number and Jaccard's coefficient. All metrics used were calculated in four different ways: using the entire graph, using nodes within a radius of one, using nodes within a radius of two and using nodes within a radius of three. The time taken to compute the metrics was recorded. The column headings for the data set calculated are:
BR1F, BR2F, BR3F, BRF, BR1T, BR2T, BR3T, BRT, CNR1, CNR2, CNR3, AAR1, AAR2, AAR3, JCR1, JCR2, JCR3, BR1FTE, BR2FTE, BR3FTE, BRFTE, BR1TTE, BR2TTE, BR3TTE, BRTTE, CNR1TE, CNR2TE, CNR3TE, AAR1TE, AAR2TE, AAR3TE, JCR1TE, JCR2TE, JCR3TE.

F stands for From (the first node in a dyad), *T* stands for To (the second node in a dyad), *R1* means the metric was calculated within a radius of two, *R2* means the metric was calculated within a radius of two, *R3* means the metric was calculated within a radius of three, *B* stands for betweenness, *CN* for common neighbours, *AA* stands for Adamic\Adar and *JC* stands for Jaccard's coefficient. The time elapsed to calculate each metric individually is recorded in the final columns, named as the metric abbreviation suffixed with the letters *TE*. Aggregate statistics were calculated for the computation

speeds of each metric type. The accuracy and speed of each type of metric for link prediction was recorded and is compared in the next section.

6.7. Results

This results section presents the statistics of the metrics calculated, their predictive accuracy, and the time taken to compute the metrics at various radii. These results are presented in the tables and graphs below. Each row in the table below gives the statistics associated with a certain metric. Whereas the metrics in the previous chapter were grouped by the base type of temporal variants, the metrics here are grouped by the base type of the radial variants. The thick black horizontal lines delineate the different metric groupings. The first column in the table gives the name of the metric. The second and third columns give the average of the values for the metric for the unconnected- and forming dyads respectively. The fourth and fifth columns give the standard deviation of the values for the metric for the unconnected- and forming dyads respectively. The mean difference column shows the value of the unconnected mean minus the forming mean. Thus, it is negative when metrics associated with a forming dyad are larger than those associated with an unconnected dyad. The test statistic column gives the value of the Normal distribution test statistic, calculated using the mean and standard deviation values as described in the research methodology chapter. The subsequent column gives the significance level associated with the test statistic. The final column shows the kappa statistic for the metric, as given by Weka, using logistic regression. It is not related to the standard hypothesis testing statistics given in the previous column, but allows the reader to compare the general usefulness of each temporal variation of the statistic.

Table 14. Metric statistics, grouped by category

Metric	Unconnected mean	Forming mean	Unconnected standard deviation	Forming standard deviation	Mean Difference	Test statistic	Significance level	Kappa
BR1F	0.3340	0.6377	0.4672	0.4758	-0.3037	-45.4	0.01%	30.53%
BR2F	0.1854	0.4224	0.2901	0.3608	-0.2370	-51.03	0.01%	31.60%
BR3F	0.0921	0.1868	0.1671	0.1974	-0.0947	-36.49	0.01%	26.68%
BRF	0.0002	0.0013	0.0005	0.0054	-0.0011	-21.1	0.01%	23.58%
BR1T	0.3311	0.5983	0.4668	0.4862	-0.2672	-39.52	0.01%	27.80%
BR2T	0.1842	0.3846	0.2893	0.3592	-0.2004	-43.31	0.01%	26.55%
BR3T	0.0917	0.1791	0.1685	0.2030	-0.0874	-33.04	0.01%	24.36%
BRT	0.0002	0.0012	0.0007	0.0050	-0.0010	-20.24	0.01%	22.96%
CNR1	0.2160	0.7674	4.6342	5.0327	-0.5514	-8.03	0.01%	15.04%
CNR2	4.2813	20.2140	28.2728	59.4047	-15.9327	-24.14	0.01%	21.36%
CNR3	60.5142	191.382	201.5010	355.1156	-130.8680	-31.95	0.01%	24.21%
AAR1	0.2959	1.0101	6.6808	7.2930	-0.7142	-7.2	0.01%	15.04%
AAR2	5.5188	27.5717	39.3169	83.1699	-22.0529	-23.9	0.01%	20.33%

Metric	Unconnected mean	Forming mean	Unconnected standard deviation	Forming standard deviation	Mean Difference	Test statistic	Significance level	Kappa
AAR3	85.0529	271.402	289.5494	513.1495	-186.3493	-31.53	0.01%	23.77%
JCR1	0.0020	0.0148	0.0442	0.1683	-0.0128	-7.34	0.01%	14.74%
JCR2	0.0097	0.0448	0.0767	0.2767	-0.0351	-12.18	0.01%	21.98%
JCR3	0.0468	0.1403	0.1702	0.3295	-0.0935	-25.13	0.01%	25.49%

The table below lists each metric individually in separate rows and describes their usefulness (contribution to accuracy) in a logistic regression. The first column shows the kappa statistic, which is a measure of how much more accurate a prediction is, compared to a random prediction. If the value is negative it means that using the given metric gives a prediction model which is even worse than guessing randomly. The metrics are ranked in descending order of their kappa value. The total accuracy can be described in different ways. The last column shows the overall accuracy of the regression. The third and fourth columns are important and show the true positive rate for each class.

Table 15. Metric predictive accuracy, ranked by kappa

Metric	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
BR1F	30.53%	66.7%	63.8%	65.2750%
BR2F	31.60%	72.7%	58.9%	65.8451%
BR3F	26.68%	76.3%	50.3%	63.4306%
BRF	23.58%	87.1%	36.4%	61.9718%
BR1T	27.80%	66.6%	61.2%	63.9168%
BR2T	26.55%	70.9%	55.6%	63.3300%
BR3T	24.36%	75.4%	48.9%	62.2736%
BRT	22.96%	85.5%	37.3%	61.6533%
CNR1	15.04%	96.1%	18.8%	57.8303%
CNR2	21.36%	92.6%	28.7%	60.9155%
CNR3	24.21%	87.8%	36.3%	62.2904%
AAR1	15.04%	96.1%	18.8%	57.8303%
AAR2	20.33%	92.9%	27.3%	60.4125%
AAR3	23.77%	88.0%	35.6%	62.0724%
JCR1	14.74%	96.1%	18.5%	57.6794%
JCR2	21.98%	89.7%	32.1%	61.2005%
JCR3	25.49%	86.2%	39.1%	62.9108%

The table below uses the same columns as the one above, but instead of showing the accuracy of each metric individually it shows the accuracy of regressions performed with different sets of metrics. The metric subset column describes the type of metrics used in italics and then lists all the metrics in the set.

Table 16. Metrics set predictive accuracy

Metric subset	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
Traditional metrics: BRF, BRT, CNR1, AAR1, JCR1	39.83%	79.8%	59.9%	69.9698%
Radius 1 metrics: BR1F, BR1T, CNR1, AAR1, JCR1	31.50%	67.1%	64.4%	65.6712%
Radius 1 betweenness and radius 3 common neighbour-based: BR1F, BR1T, CNR3, AAR3, JCR3	33.04%	73.8%	59.2%	66.5661%
Traditional and radius 2 metrics: BRF, BRT, CNR1, AAR1, JCR1, BR2F, BR2T, CNR2, AAR2, JCR2	41.24%	73.9%	67.3%	70.6405%

The table below shows the mean and standard deviation of the time taken in milliseconds to compute a metric for one node (or pair of nodes).

Table 17. Metric computation time

Metric	Mean	Standard deviation
BR1FTE	5.14	63.22
BR2FTE	123.68	925.37
BR3FTE	3541.26	18454.26

Metric	Mean	Standard deviation
BRFTE	1254975 (134.9 per node)	37.7 per node
BR1TTE	4.27	51.43
BR2TTE	112.59	770.25
BR3TTE	3232.82	16840.54
BRTTE	1254975 (134.9 per node)	37.7 per node
CNR1TE	0.16	1.26
CNR2TE	0.86	2.86
CNR3TE	4.78	8.7
AAR1TE	0.14	1.18
AAR2TE	0.91	2.95
AAR3TE	5.44	9.43
JCR1TE	0.13	1.14
JCR2TE	0.87	2.9
JCR3TE	4.64	7.95

The following two tables, 18 and 19, and two graphs, Figure 34 and Figure 35, were calculated from a new experiment run on the same data set. Metrics were computed up to a radius of six for common neighbour-based metrics only. The same number of instances were used. This new experiment was run because it was noted in the first experiment that the accuracy of common neighbour-based metrics kept increasing as the radius used increased. This table allows us to see whether the kappa of a metrics keeps increasing indefinitely at larger radii, and at what rate. This is explained in the conclusions section. The table below shows the predictive accuracy of common neighbour-based metrics up to a radius of six.

Table 18. Common neighbours metric predictive accuracy, ranked by kappa

Metric	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
CNR1	14.77%	95.8%	18.8%	57.6962%
CNR2	21.36%	91.6%	29.7%	60.8987%
CNR3	23.69%	86.5%	37.1%	62.0221%
CNR4	24.77%	79.9%	44.8%	62.5084%
CNR5	26.90%	75.3%	51.5%	63.5312%
CNR6	27.45%	71.9%	55.6%	63.7827%
AAR1	14.77%	95.8%	18.8%	57.6962%
AAR2	21.26%	92.4%	28.8%	60.8652%
AAR3	23.65%	87.1%	36.5%	62.0054%
AAR4	24.76%	80.3%	44.4%	62.5084%
AAR5	26.80%	75.2%	51.5%	63.4809%
AAR6	27.56%	71.7%	55.8%	63.8330%
JCR1	0.00%	0.00%	100.0%	49.5305%

Metric	Kappa	Unconnected TP rate	Forming TP rate	Overall accuracy
JCR2	20.96%	89.3%	31.6%	60.6975%
JCR3	24.52%	85.2%	39.2%	62.4245%
JCR4	24.47%	81.6%	42.8%	62.3742%
JCR5	24.12%	82.2%	41.9%	62.2066%
JCR6	23.23%	83.0%	40.1%	61.7706%

The table below shows the mean and standard deviation of the time taken in milliseconds to compute a common neighbour-based metric for one node (or pair of nodes) up to a radius of six.

Table 19. Metric computation time up to radius six

Metric	Mean	Standard deviation
CNR1TE	0.19	2.03
CNR2TE	1.08	4.71
CNR3TE	5.79	13.2
CNR4TE	22.2	31.52
CNR5TE	59.01	62.47
CNR6TE	120.17	107.71
AAR1TE	0.2	2.45
AAR2TE	1.18	5.07
AAR3TE	6.61	14.11
AAR4TE	26.34	37.18
AAR5TE	67.42	71.38
AAR6TE	133.32	121.21
JCR1TE	0.2	1.94
JCR2TE	1.14	5.87
JCR3TE	6.1	14.39
JCR4TE	22.42	31.59
JCR5TE	58.57	62.1
JCR6TE	120.01	108.07

The graph below shows the total accuracy of each of the three common neighbour-based metrics at each of the six radii. Some lines overlap.

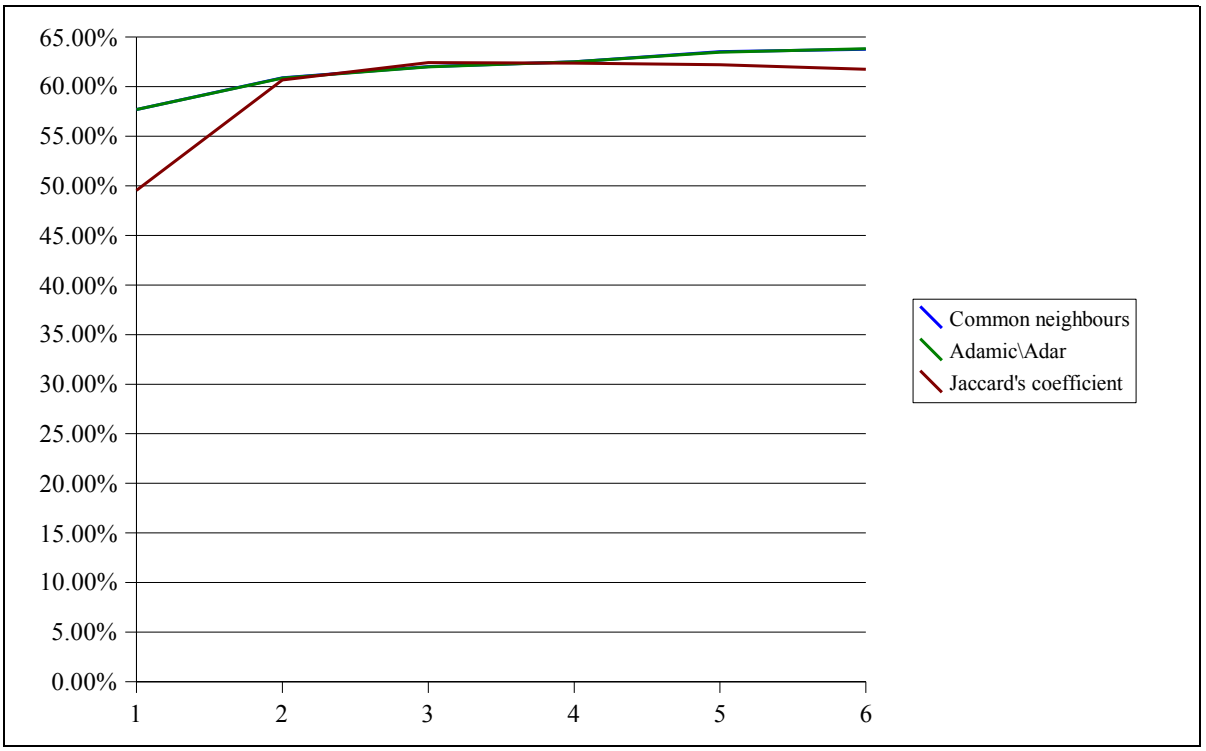

Figure 34. Total predictive accuracy per radius

The graph below shows the time taken in milliseconds to compute each of the three common neighbour-based metrics at each of the six radii. Some lines overlap.

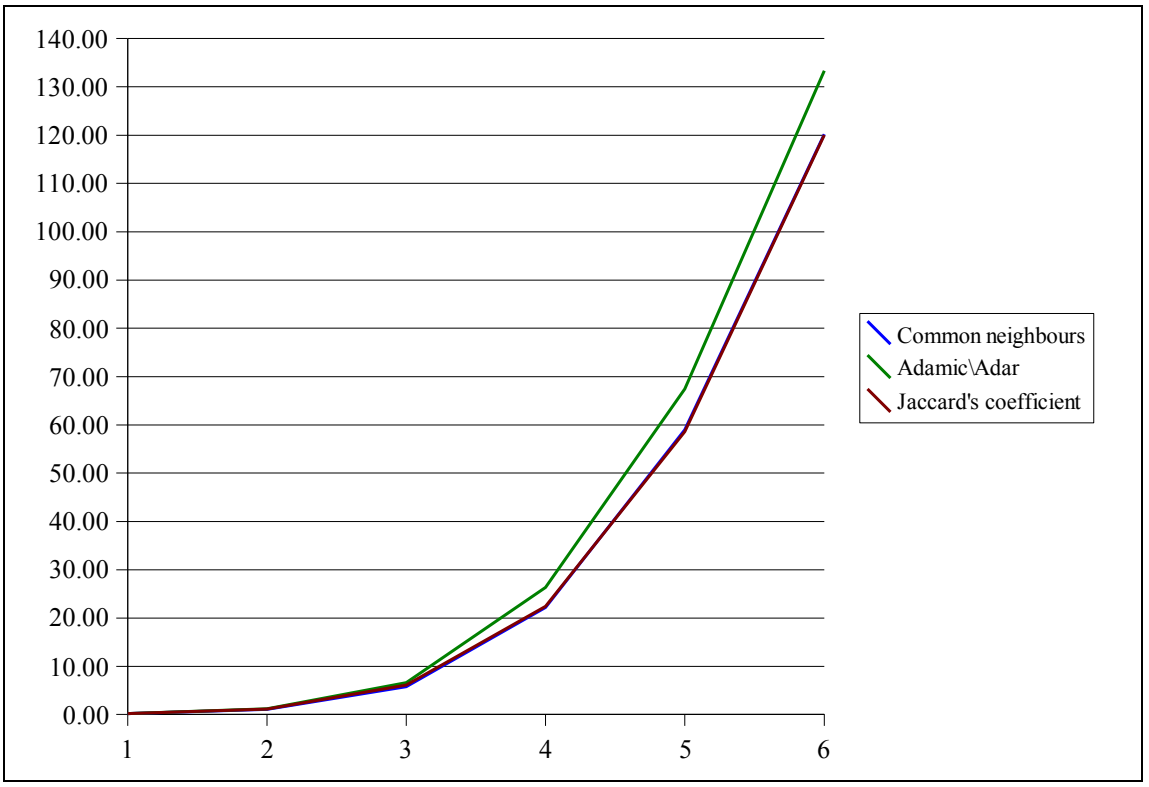

Figure 35. Computational time in milliseconds per radius

6.8. Conclusions

Table 14 shows us that, unlike in the previous experiments, for all the metrics at all radii, the forming mean is greater than the unconnected mean. Furthermore these differences are all very highly statistically significant. Table 15 presents a detailed overview of the accuracy of these metrics in a logistic regression. A glance at the kappa and accuracy columns shows us that accuracy increases as the radius decreases for betweenness, but that accuracy increases as the radius increases for common neighbour metrics. This relationship is not as simple as it first appears. If we look at the TP columns for betweenness we see that as radius decreases the unconnected TP rate drops while the forming TP rate rises. And if we look at the common neighbour-based metrics TP columns we see that as radius increases the unconnected TP rate drops while the forming TP rate rises. However, the overall accuracy still increases in both cases, meaning that the unconnected TP rate does not drop as much as the forming TP rate rises. In both cases the overall accuracy rises only about 4%.

Table 16 shows how the metrics perform when used in groups for prediction. The first row shows the traditional metrics that would be used for prediction, common neighbour metrics calculated at a radius of one and betweennness calculated for the entire graph. The overall accuracy is relatively high, at 70%, with both TP rates above 59%. The second row shows the accuracy when the betweenness metrics used in the first row are replaced by betweenness metrics calculated at a radius of one. The accuracy drops, but down only 4%. It is surprising that using the betweenness metrics at a lower radius in the set would decrease accuracy when individually they have a higher accuracy at lower radii. If we use betweenness calculated at a radius of one and common neighbour metrics at a radius of three, the accuracy increases a percent from 66% to 67%. The highest possible accuracy, 71%, was obtained using both the traditional set of metrics and the metrics calculated at a radius of two.

These accuracies are not important of themselves, but rather when viewed in light of the time taken to compute the metrics used. This is because one of aims of this chapter was to perform link prediction successfully using metrics calculated locally and quickly. Table 17 lists the mean time taken to calculate a metric for a given node or pair of nodes at each radius. We can see that betweenness quickly scales up from five milliseconds at a radius of one to around three and a half seconds at a radius of three. This is expected, based on the discussion of the computational complexity of the calculation of betweenness given at the beginning of this chapter. It takes roughly 21 minutes to calculate the global betweenness for a node in an average time step in the graph (1254975 milliseconds). However, when calculating betweenness using the entire graph we can calculate the betweenness value for every node at once. Therefore the average time per node is 135 milliseconds. This is a purely theoretical number since even if we want to calculate the betweenness for only one

node it still takes the full 21 minutes using Brandes' algorithm. The common neighbour-based metric computation times also increase exponentially with radius, but remain remarkably small – under six milliseconds. There is little variance in the elapsed times between the different types of common neighbour-based metrics as the majority of calculation time is spent finding and counting common neighbours, which is common to all the metrics.

Since betweenness has its highest kappa value at a radius of one, we do not need to investigate it at large radii. However, the common neighbour-based metrics keep on increasing and have their highest value at radius three. Thus the experiment was run again for only the common neighbour-based metrics up to a radius of six. Table 18 and Table 19 show the results of this second experiment. We can see that the accuracy of the metrics keeps on increasing until a radius of six, though at a decreasing rate. The graph below the tables illustrates how the increase in accuracy appears to level off at about a radius of four. This indicates that taking common neighbours at a higher radius is not of much use. This is especially true as the time taken to compute the metrics keeps increasing exponentially, reaching a tenth of a second at a radius of six. The combined meaning of the accuracy and computational time of the metrics is now discussed. Firstly, it has been shown that common neighbour-based metrics become more useful for prediction at radii higher than one. Computing these metrics at a radius higher than one also effectively solves the "distance three task" discussed by Liben-Nowell and others. These metrics remain quick to calculate, even up to a radius of six. Secondly, it has been shown that distance-based metrics such as betweenness can be calculated at far smaller radii than the entire graph and still remain highly useful. There is a drop in accuracy of only 3% when they are used in combination with other metrics, and they are in fact more useful individually when computed at a radius of one. However, this small drop in accuracy is compensated by a massive decrease in computational time – from twenty one minutes to between five milliseconds and three seconds (radius dependent) for a single node. Thus the local metric approach has helped us solve the distance three task, increase the accuracy of common neighbour-based metrics and decreased the computation time of distance-based metrics such as betweenness and closeness.

To summarise, it has been found that:

- The "distance three task" can be solved by using the expanded definition common neighbour-based metrics that accommodate higher radii.
- Common neighbour-based metrics become better link predictors when calculated at radii greater than one, levelling off around a radius of four.
- Distance-based metrics, such as betweenness and closeness, can be calculated extremely quickly using a local definition and small radii while still retaining nearly all of their usefulness.

Chapter 7. A combined approach

This final experimental chapter describes the effectiveness of combining the techniques discussed in the three previous experiments. This means that an experiment was conducted that attempted to classify unconnected-, hidden- and forming links using both local and temporal techniques simultaneously. Furthermore, this experiment used the email data set obtained from Netcash in addition to the Pussokram data set to see how the techniques fared on the type of data set that would more commonly be used in reality.

7.1. Hypothesis statement

The null hypothesis of this experiment is that using temporal metrics and local metrics in addition to traditional metrics will not increase the accuracy of link detection and link prediction. More mathematically, the null hypothesis is that the kappa value of a regression performed using temporal and local metrics in addition to traditional metrics will be equal to the kappa value of a regression performed using traditional metrics. As we have studied the mean values of various metrics in previous chapters they are not included in this chapter.

7.2. Methodology

The methodology of this experiment follows the general methodology described in the research methodology chapter and uses parts of all the previous experiments. From the first experiment we use the idea of a three class y-variable. In other words the y-variable can be one of three different classes: unconnected, hidden, or forming. From the second experiment we take the idea of recency and the average of a mean over the ten previous time steps. We use only ten time steps as this was recommended as a good compromise between speed and accuracy in the second experiment's conclusions. From the third experiment we use local metrics calculated at a radius of four. Once again, this radius was used as per the recommendations of the third experiment. Only the most useful metrics from each chapter were used. The metrics chosen were distance, the Katz measure, monadic recency, common neighbours, preferential attachment and degree. Additionally common neighbours, preferential attachment and degree were calculated at a radius of four. Finally the average of all these metrics was taken over the ten previous time steps. Thus the column headings calculated are:
Dist, Katz, RecF, RecT, CN, PA, DegreeF, DegreeT, CNR4, PAR4, DegreeFR4, DegreeTR4, DistA10, KatzA10, RecFA10, RecTA10, CNA10, PAA10, DegreeFA10, DegreeTA10, CNR4A10, PAR4A10, DegreeFR4A10, DegreeTR4A10.

The suffix *F* stands for From (the first node in a dyad), *T* stands for To (the second node in a dyad), *R4* stands for radius four and indicates the metric was calculated using a local radius of four and *A10* stands for average for the past ten time steps. The experiment was run on the Pussokram data set for time steps 50 to 150 as usual. Additionally the experiment was run on the Netcash email data set, also for time steps 50 to 150.

7.3. Results

This results section presents the predictive accuracy of the metrics calculated individually and in sets. The first subsection presents the results of the experiment using the Pussokram data set and second subsection presents the results of the experiment using the Netcash data set. These results are presented in two tables in each section. The first table shows detailed results of the accuracy attained by logistic regression using each metric individually. The second table shows shows the accuracy attained using sets of metrics.

7.3.1. Pussokram results

This subsection presents the classification results of the Pussokram data set. It is presented in the same way as the Netcash results shown in the next section. The table below lists each metric individually in separate rows and describes their usefulness (contribution to accuracy) in a logistic regression. The first column shows the kappa statistic, which is a measure of how much more accurate a prediction is, compared to a random prediction. The metrics are ranked in descending order of their kappa value.

Table 20. Individual metric predictive accuracy, ranked by kappa

Metric	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
RecF	28.67%	35.7%	46%	75.6%	52.448%
DegreeTA10	26.07%	76.4%	28.7%	46.9%	50.7825%
DegreeTR4A10	25.17%	68%	45%	37.3%	50.1341%
DegreeT	17.17%	82.5%	30.1%	21.7%	44.858%
DegreeTR4	15.11%	71%	48.9%	10.3%	43.4161%
PA	15.1%	87.2%	19.2%	23.6%	43.4943%
DegreeFA10	13.42%	34.1%	36.1%	56.5%	42.2647%
DegreeF	12.74%	63%	19%	43.2%	41.8958%
Katz	11.58%	79.3%	6.2%	37.4%	41.1804%
CNR4	11.38%	79.9%	6.2%	36.6%	41.0575%
RecFA10	9.89%	3%	52.6%	64.2%	39.8279%

Metric	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
PAR4	9.63%	85.3%	8.4%	25.4%	39.8949%
RecT	9.26%	34.9%	14.2%	69.2%	39.5372%
DegreeFR4	8.69%	56.1%	14.2%	46.9%	39.2131%
PAA10	7.98%	77.8%	19.6%	18.3%	38.6877%
DegreeFR4A10	5.91%	0%	70.4%	41.5%	37.1004%
KatzA10	3.66%	67.8%	12.7%	26.5%	35.8149%
CNR4A10	3.27%	69.2%	11.2%	25.9%	35.569%
PAR4A10	2.60%	74.4%	12.3%	18.2%	35.1218%
DistA10	2.43%	17.2%	87.7%	0%	34.6971%
CN	1.66%	99.9%	3.5%	0%	34.6412%
Dist	-1.28%	19.7%	33.7%	44.2%	32.495%
CNA10	-1.73%	86.9%	9.5%	0%	32.2714%
RecTA10	-1.97%	32.6%	13.6%	49.6%	32.0143%

The table below uses the same columns as the one above, but instead of showing the accuracy of each metric individually it shows the accuracy of regressions performed with sets of metrics. The metric subset column describes the type of metrics used in italics and then lists all the metrics in the set. The last two rows of the table show the results of the predictions performed only using the unconnected- and forming classes, without the hidden link class.

Table 21. Metrics set predictive accuracy

Metric subset	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
Traditional metrics: Dist, Katz, CN, PA, DegreeF, DegreeT	27.58%	74.5%	42.9%	37.7%	51.755%
All metrics	40.64%	69.9%	42.8%	68.4%	60.4516%
All metrics except Katz	40.62%	69.1%	43.3%	68.7%	60.4404%

Metric subset	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
Weka's logistic subset classifier genetic search attribute selection search: Dist, Katz, RecF, RecT, CN, DegreeF, DegreeT, PAR4, DegreeFR4, DegreeTR4, DistA10, KatzA10, RecFA10, RecTA10, CNA10, DegreeTA10, CNR4A10, PAR4A10, DegreeFR4A10, DegreeTR4A10	39.98%	69.1%	42.5%	68.2%	60.0156%
Traditional metrics: Dist, Katz, CN, PA, DegreeF, DegreeT	37.7%	80.3%	-	57.4%	68.8799%
All metrics	60.53%	80%	-	80.5%	80.2649%

7.3.2. Netcash results

This subsection presents the classification results of the Netcash data set. It is presented in the same way as the Pussokram results shown in the previous section. The table below lists each metric individually in separate rows and describes their usefulness (contribution to accuracy) in a logistic regression. The first column shows the kappa statistic, which is a measure of how much more accurate a prediction is, compared to a random prediction. The metrics are ranked in descending order of their kappa value.

Table 22. Individual metric predictive accuracy, ranked by kappa

Metric	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
DegreeT	69%	97.9%	83.3%	57.3%	79.3056%
DegreeTA10	50.46%	94.5%	83.1%	24%	66.875%
RecTA10	45.23%	68.9%	89.8%	32.2%	63.4722%
DegreeTR4	44.87%	97.9%	84.2%	8.6%	63.125%
RecT	44.73%	71.9%	89%	29.2%	63.125%
DegreeTR4A10	43.59%	92.6%	81.7%	13.8%	62.2917%
KatzA10	22.53%	90.3%	33.5%	21.6%	48.125%
CNA10	22.33%	90.3%	32.9%	21.8%	47.9861%
PAA10	21.49%	90.3%	29.2%	23.8%	47.4306%
PAR4A10	21.4%	90.3%	31%	21.8%	47.3611%
CNR4A10	21.3%	90.3%	31%	21.6%	47.2917%
DegreeF	20.6%3	98.1%	30.4%	13.1%	46.8056%
DegreeFR4A10	20.48%	92.8%	31.9%	16.6%	46.7361%
DegreeFA10	19.03%	93%	32.3%	13.1%	45.7639%
RecF	17.94%	57.7%	15.2%	63%	45.3472%
PA	17.16%	100%	30.2%	4.5%	44.4444%
Dist	16.96%	100%	33.8%	0.6%	44.3056%
Katz	16.86%	100%	34.2%	0%	44.2361%
CN	16.66%	100%	33.8%	0%	44.0972%
DistA10	16.15%	100%	32.1%	0.6%	43.75%
PAR4	16.13%	100%	31.7%	0.01%	43.75%
CNR4	16.03%	100%	31.7%	0.8%	43.6806%
DegreeFR4	15.09%	98.1%	32.5%	0%	43.0556%
RecFA10	13.35%	53.5%	16.5%	56.9%	42.2917%

The table below uses the same columns as the one above, but instead of showing the accuracy of each metric individually it shows the accuracy of regressions performed with sets of metrics. The metric subset column describes the type of metrics used in italics and then lists all the metrics in the set. The last two rows of the table show the results of the predictions performed only using the unconnected- and forming classes, without the hidden link class.

Table 23. Metrics set predictive accuracy

Metric subset	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
Traditional metrics: Dist, Katz, CN, PA, DegreeF, DegreeT	73.36%	97.7%	85%	64.5%	82.2222%
All metrics	76.88%	94.1%	87.5%	72.5%	84.5833%

Metric subset	Kappa	Unconnected TP rate	Hidden TP rate	Forming TP rate	Overall accuracy
All metrics except Katz	77.09%	94.1%	87.9%	72.5%	84.7222%
Weka's logistic subset classifier genetic search (and best search) attribute selection search: Katz, RecT, CN, DegreeF, DegreeT, DegreeTR4, DistA10, KatzA10, RecFA10, RecTA10, PAA10, DegreeFA10, DegreeTA10, CNR4A10, DegreeTR4A10	78.34%	93.2%	87.9%	75.8%	85.5556%
Traditional metrics: Dist, Katz, CN, PA, DegreeF, DegreeT	75.98%	97.2%	-	79.4%	87.9167%
All metrics	84.21%	97%	-	87.6%	92.0833%

7.4. Conclusions

This section draws conclusions from the results presented in the previous section. A general overview of the results shows us that the kappa values for the Netcash data were consistently higher than those for the Pussokram data. As discussed in chapter three, the ratio of nodes to bidirected links of Netcash was far larger than the ratio of Pussokram. Since Netcash had relatively few bidirected links (between 131 and 227) it might be that the same hidden links were used over and over again in each timestep. This would allow Weka to learn their associated metrics easily, leading to increased classification accuracy. Therefore the experiment was rerun using only five instances of each class per timestep (instead of the average of 47) to see if raising the ratio of available hidden links to forming links would decrease classification accuracy. The kappa of this experiment (which is not

presented here) using all metrics was virtually identical (within a percent) to the kappa of the original experiment. This suggests that the ratio of bidirected links to nodes may not be the cause of Netcash's increased prediction accuracy. Therefore it may be that the prediction is easier for a network that is obtained from email records than one obtained from other sources (it was mentioned in chapter three that Pussokram is the set of graph sequences obtained from the messages exchanged by registered users on a social networking website and Netcash is the set obtained from a business server's email records). Alternatively it may be that prediction was simpler in this data set in particular, and that the same results may not be reproducible in other networks. In other words some data sets may be more random than others, i.e. more complex for the purposes of prediction. We could thus classify the Pussokram network as complex and the Netcash network as simple.

We now examine tables 20 and 22, the individual metric regression results for both data sets. The overall accuracy of the individual metric regressions ranges from 32% to 52% for Pussokram and from 42% to 79% for Netcash. No Pussokram kappa value is above 40%, although there are several above 40% for Netcash. Metrics that are useful in both data sets are: DegreeT, DegreeTA10, DegreeTR4, and DegreeFA10. Thus the popularity of the target node in a forming link is a reliable indicator. Metrics that are important in one data set but not in the other include: RecT, Katz, PA and RecF. This indicates that different metrics may be more applicable to different networks. A prediction system needs to test various different metrics to discover what type work best for a given network. The temporal and local variations of the traditional metrics have kappas of varying size in both data sets. This suggests that local and temporal metrics are useful, though the exact type of metric variation required may differ from network to network.

We now consider the combined metric predictive accuracy tables, 21 and 23. The overall accuracy for Pussokram is raised from 52% to 60% when temporal and local metrics are used in addition to traditional metrics. This increase is hardly affected if we do not include the Katz metric, which is computationally complex and preferable to avoid if possible. Additionally, the kappa value increases from 28% to above the level of significance, 40%. The Netcash accuracy is raised only from 82% to 85% with the inclusion of the new metrics. However, using a suitable subset of the metrics increases accuracy to 86%. These accuracy increases in both data sets indicate that local and temporal metrics are useful additions to the prediction problem. Furthermore, the highest accuracy of the unconnected- and forming classes are above 68% for both data sets and the accuracy of the Netcash hidden link class is 88%. Thus it seems that hidden links do not confound the prediction of forming links in complex networks (such as Pussokram). In addition, it seems that hidden links can be accurately predicted and separated from unconnected- and forming links in simple networks (such as Netcash). This stands in contrast to the findings of chapter four (on link prediction versus link detection), where

it seemed that hidden links could not be accurately distinguished from forming links.

The last two rows of tables 21 and 23 show the classification of only the unconnected- and forming link classes. This is the link prediction problem proper, without including the link detection problem. The kappa value for the Pussokram network rises from 38% to 61% when including local and temporal metrics in addition to traditional metrics and the kappa value rises from 76% to 84% for the Netcash network. This shows that local and temporal metrics add accuracy to the link prediction problem solution in both simple and complex networks, but add greater value to complex networks. In other words, local and temporal metrics are of great use in networks that traditional link prediction cannot cope with, where new links appear to be random. Furthermore, these accuracies can be obtained from metrics that are fast to compute. Both common neighbour-based dyadic local metrics and small radius shortest path-based monadic local metrics are computationally simple. Temporal metrics take longer to compute, but not so much longer as to be impractical, such as a metrics like globally computed betweenness. However the Pussokram kappa of 61% using both temporal and local metrics in this experiment did not exceed the kappa of 64% using only temporal metrics in chapter four's experiment. Thus it appears that local metrics do not increase prediction accuracy above the accuracy obtained using temporal metrics and traditional metrics alone. However, local metrics are still very useful in that they are faster to compute than both temporal metrics and many traditional metrics.

To summarise, it has been found that:

- Different types of local and temporal metric variations are useful in different networks. A network should be tested with several types of metrics to find which are most suited to prediction for that network in particular. However, variations of FromDegree and ToDegree seem to be consistently useful.
- Local and temporal metrics are helpful additions to the solution of the link prediction problem, but come to the fore in “complex” networks, where traditional metrics cannot give accurate predictions in a seemingly random network.
- High link prediction accuracies (kappas of 61% and 84%) can be attained using combinations of traditional, local and temporal metrics in both complex and simple networks. Furthermore, these accuracies can be obtained from metrics that are fast to compute.
- Local metrics do not increase prediction accuracy above the accuracy obtained using temporal metrics and traditional metrics alone. However, local metrics are still very useful in that they are faster to compute than both temporal metrics and many traditional metrics.
- Hidden links can be accurately predicted and separated from unconnected- and forming links in “simple” networks (such as Netcash). This stands in contrast to the findings of chapter

four (on link prediction versus link detection), where it seemed that hidden links could not be accurately distinguished from forming links.

Chapter 8. Conclusion

This chapter concludes this dissertation. It summarises the work described in all the previous chapters and the significant conclusions from each.

8.1. Summary of the work undertaken

This dissertation presented an investigation into various aspects of the link prediction problem in social networks. It described experiments conducted to distinguish between link prediction and link detection and to test the usefulness of temporal metrics and local metrics in link prediction. The original aim was to improve the accuracy of link prediction in sequences of large networks using metrics that were fast to compute and would be usable in reality.

8.2. Experimental findings

The experiment conducted to discover whether there was a difference between link prediction and link detection found that there is a difference in structure between hidden and forming links. Hidden links have twice as many common neighbours, half as large a preferential attachment and more disparate nodal degrees than forming links. Distance is not a distinguishing factor. Although all metrics differences are very highly significantly different no individual metric alone is useful for regression in accurately classifying the two classes. Furthermore, even when using all the metrics in combination it was not possible to successfully distinguish hidden links from forming ones. However, the final experiment showed that it was possible to distinguish hidden and forming links when using local and temporal metrics on relatively "simple" networks.

The experiments conducted to discover whether local and temporal metrics could aid link prediction found that these new metric definitions were indeed useful. In particular it was found that static metrics that are not useful for prediction for a given data set, may become more useful when converted to a temporal variant of the metric. Metrics that are not useful individually may become useful when used in combination with other metrics in a set. A metric's moving average is more useful than the static metric alone (except in the case of the recency metric). The recency metric is a useful new contribution to link prediction, but its temporal variants are useless and can be ignored. Using temporal metrics enhances link prediction significantly. This enhancement cannot be bettered by including local metrics – however local metrics are still quicker to compute than temporal metrics and many traditional metrics, and can be used in their place. Dyadic common neighbour-based local

metrics can also be used to solve the "distance three task". Additionally, distance-based metrics, such as betweenness and closeness, can be calculated extremely quickly using a local definition at small radii while still retaining nearly all of their usefulness.

The final experiment found that different types of local and temporal metric variations are useful in different networks. Thus a network should be tested with several types of metrics to find which are most suited to prediction for that network in particular. However, variations of FromDegree and ToDegree seem to be consistently useful. Finally it seems that local and temporal metrics are helpful additions to the solution of the link prediction problem, but come to the fore in "complex" networks, where traditional metrics cannot give accurate predictions in a seemingly random network.

8.3. Future work

The experiments conducted in this work were all investigations into new and original graph analysis techniques. There is thus a lot of scope for other researchers to verify and extend these ideas. Some possibilities for future work include:

- Inventing and testing new types of temporal and local metrics. Only a few ideas were presented in this work and an imaginative analyst should be able to design several more.
- Verifying the results of this work on other data sets. It would be interesting to see how well local and temporal metrics perform on different types of networks.
- Investigating to what extent using different link types in metric definitions affect the accuracy of predictions. For instance, degree can be defined as in-degree, out-degree, bidirected-degree or inout-degree. Only bidirected links were used in this research.
- Including link strength (e.g. number of emails exchanged between two nodes) in metrics calculations to investigate its effect on the accuracy of predictions. This also includes the possibility of links decreasing in strength over time (i.e. friendships fading away).
- Combining local and temporal metrics with other data mining systems more sophisticated than logistic regression, in order to further increase prediction accuracy. This includes systems that use content analysis.

The work presented in this dissertation is a valuable contribution to the problem of link prediction, leading to significantly increased prediction accuracies and presenting two original new approaches to analysing graphs. It is hoped that other researchers and intelligence analysts will use these techniques in real world endeavours and extend the ideas presented in this discussion in new and interesting ways.

References

[1] Adamic, LA & Adar, E 2003, 'Friends and Neighbors on the Web', Social Networks, vol. 25, no. 3, pp. 211-230.

[2] Adamic, LA, Lukose, RM, & Huberman, BA 2002, 'Local search in unstructured networks', in *Handbook of Graphs and Networks: From the Genome to the Internet*, Bornholdt, S & Schuster, HG (eds), Wiley-VCH, Berlin, viewed 12 June 2006, http://arxiv.org/PS_cache/cond-mat/pdf/0204/0204181.pdf.

[3] Aiello, W, Chung, F & Linyuan L 2002, 'Random evolution in massive graphs', in *Handbook of Massive Data Sets*, Abello, J & Pardalos, PM & Resende, MGC (eds), Kluwer Academic, Dordrecht.

[4] Alberich, R, Miro-Julia, A, & Rossello, F 2006, 'Marvel Universe looks almost like a real social network', *arXiv.org*, viewed 12 June 2006, http://arxiv.org/PS_cache/cond-mat/pdf/0202/0202174.pdf.

[5] April, K, Potgieter, A & Cooke, R 2005, 'Using Bayesian Agents to Enable Distributed Network Knowledge: A Critique', *Proceedings 4th International Critical Management Studies Conference*, 4-6 July 2005.

[6] April, K, Potgieter, A, Cooke, R & Lockett, M 2006, 'Adaptive Bayesian agents: Enabling distributed social networks', *South African Journal of Business Management,* vol. 37, no. 1, pp. 41-55.

[7] Beinhocker, ED 1997, 'Strategy at the edge of chaos', *The McKinsey Quarterly*, vol. 1997, no. 1, viewed 12 June 2006, http://www.rose-hulman.edu/~christ/mgrecon/beinhocker.pdf.

[8] Benjamini, I & Lovasz, L 2002, 'Global information from local observation', paper presented at *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, 16-19 November 2002.

[9] Brandes, U 2001, 'A Faster Algorithm for Betweenness Centrality', *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163-167.

[10] Campbell, CS, Maglio, PP, Cozzi, AC & Dom B 2003, 'Expertise Identification using Email Communications', *Proceedings of the twelfth international conference on Information and knowledge*

management, 3-8 November 2003, New Orleans.

[11] Carletta, J 1996, 'Assessing agreement on classification tasks: the kappa statistic', *Computational Linguistics*, vol. 22, no. 2, pp 249-254, viewed 12 June 2006, http://acl.ldc.upenn.edu/J/J96/J96-2004.pdf.

[12] Carley, K 2003, 'Dynamic Network Analysis', forthcoming in in the Summary of the NRC workshop on Social Network Modeling and Analysis, Breiger, R & Carley, KM (eds), National Research Council, viewed 12 June 2006, http://stiet.si.umich.edu/researchseminar/Winter%202003/DNA.pdf.

[13] Carpenter, T & Karakostas, G & Shallcross, D 2002, 'Practical Issues and Algorithms for Analyzing Terrorist Networks', Telcordia Technologies, viewed 12 June 2006, http://www.cas.mcmaster.ca/~gk/papers/wmc2002.pdf.

[14] Carre, B 1979, *Graphs and Networks*, Oxford University Press, Oxford.

[15] Chen, F & Farahat, A & Brants, T, 'Multiple Similarity Measures and Source-Pair Information in Story Link Detection', *Proceedings of the Human Language Technology Conference, North American Chapter of the Association for Computational Linguistics Annual Meeting*, 2-7 May 2004, Boston, USA, viewed 12 June 2006, http://acl.ldc.upenn.edu/hlt-naacl2004/main/pdf/18_Paper.pdf.

[16] Coffman, T, Greenblatt, S & Marcus, S 2004, 'Graph-Based Technologies for Intelligence Analysis', *Communications of the ACM*, vol. 47, no. 3, pp 45-47.

[17] Crucitti, P, Latora, V, & Marchiori, M and Rapisarda, A 2002, 'Efficiency of Scale-Free Networks: Error and Attack Tolerance', *Physica A*, vol. 320, pp. 622-642, viewed 12 June 2006, http://arxiv.org/PS_cache/cond-mat/pdf/0205/0205601.pdf.

[18] Deo, N 1974, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Upper-Saddle River, New Jersey.

[19] Desikan, P & Srivastava, J 2004, 'Mining temporally evolving graphs', paper presented at *Proceedings of the sixth WEBKDD workshop in conjunction with the 10th ACM SIGKDD conference*, August 22 2004, Seattle, viewed 12 June 2006, http://maya.cs.depaul.edu/webkdd04/final/desikan.pdf.

[20] Farrell, S, Campbell, C & Myagmar, S 2005, 'Relescope: An Experiment in Accelerating Relationships', paper presented at *Conference on Human Factors in Computing Systems*, 2-7 April 2005, Portland.

[21] Fellman, PV & Wright, R 2004, 'Modelling Terrorist Networks - Complex Systems at the Mid-Range', viewed 12 June 2006, http://www.psych.lse.ac.uk/complexity/Conference/FellmanWright.pdf.

[22] Fleiss, JL 1981, *Statistical methods for rates and proportions*, second edition, John Wiley & Sons, New York.

[23] Gell-Mann, M 1995, *The Quark and the Jaguar: Adventures in the Simple and Complex*, W. H. Freeman, New York.

[24] Gentle, JE 1998, *Numerical linear algebra for applications in statistics*, viewed 12 June 2006, http://www.science.gmu.edu/jgentle/myfiles/linbook.pdf.

[25] Getoor, L & Diehl, C 2005,'Link Mining: A Survey', *ACM SIGKDD Explorations Newsletter*, vol. 7, issue 2, pp 3-12.

[26] Golbeck, J 2005 'Semantic Web Interaction through Trust Network Recommender Systems', *End User Semantic Web Interaction Workshop at the 4th International Semantic Web Conference, November 2005.*

[27] Golbeck, J & Hendler, J 2004, 'Reputation Network Analysis for Email Filtering', *Proceedings of the First Conference on Email and Anti-Spam*, 30-31 July 2004.

[28] Goldenberg, A, Kubica, J Komarek, P, Moore, A & Schneider, J 2003, 'A Comparison of Statistical and Machine Learning Algorithms on the Task of Link Completion', *Proceedings of the KDD Workshop on Link Analysis for Detecting Complex Behavior*, August 2003, viewed 12 June 2006, http://www.autonlab.org/autonweb/14624/version/2/part/5/data/linkcomplete2003.pdf?branch=main&language=en.

[29] Gould, RJ 1988, *Graph Theory*, The Benjamin/Cummings Publishing Company, San Francisco.

[30] Hanneman, R 2001, *Introduction to Social Network Methods*, viewed 12 June 2006,

http://faculty.ucr.edu/~hanneman/nettext/networks.zip.

[31] Henziger, M 2000,'Link Analysis in Web Information Retrieval', *IEEE Data Engineering Bulletin*, vol. 23, no. 3, pp 3 – 8, viewed 12 June 2006, http://www.eicstes.org/EICSTES_PDF/PAPERS/Link%20analysis%20in%20web%20information%20retrievel%20(Henzinger).pdf.

[32] Holme, P 2003,'Network dynamics of ongoing social relationships', *Europhys. Lett*., no. 64, pp. 427-433, viewed 12 June 2006, http://arxiv.org/PS_cache/cond-mat/pdf/0308/0308544.pdf.

[33] Holme, P, Edling, C & Liljeros, F 2004, 'Structure and Time Evolution of an Internet Dating Community', *Social Networks*, no. 26, pp. 155-174, viewed 12 June 2006, http://arxiv.org/pdf/cond-mat/0210514.

[34] Hosmer, DW & Lemeshow, S, *Applied logistic regression*, John Wiley & Sons, New York, 1989.

[35] Huang, Z, Li, X & Chen, H 2005, 'Link Prediction Approach to Collaborative Filtering', *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, 7-11 June 2005.

[36] Jenssen, JI & Koenig, HF 2002, 'The Effect of Social Networks on Resource Access and Business Start-ups', *European Planning Studies*, vol. 10, no. 8.

[37] Kempe, D & McSherry, F 2004, 'A Decentralized Algorithm for Spectral Analysis', *Proceedings of the Thirty-sixth annual ACM symposium on Theory of computing*, 13-15 June 2004.

[38] Kleinberg, J 2000, 'The Small-World Phenomenon: An Algorithmic Perspective', *Proceedings of the 32nd ACM Symposium on Theory of Computing*.

[39] Komarek, P 2004, 'Logistic Regression for Data Mining and High-Dimensional Classification', thesis at the Robotics Institute, Carnegie Mellon University.

[40] Leskovec, J, Kleinberg, J & Faloutsos, C 2005, 'Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations', *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 21-24 August 2005.

[41] Liben-Nowell, D 2005, An Algorithmic Approach to Social Networks, PhD thesis at MIT

Computer Science and Artificial Intelligence Laboratory.

[42] Liben-Nowell, D & Kleinberg, J 2003, 'The link prediction problem for social networks', *Proceedings of the twelfth international conference on information and knowledge management*, 3-8 November 2003, pp. 556-559.

[43] Lim, M, Negnevitsky, M & Hartnett, J 2005, 'Artificial Intelligence Applications for Analysis of E-mail Communication Activities', *Proceedings of the International Conference On Artificial Intelligence In Science And Technology*, pp. 109-113.

[44] Yeomans, M 2005, 'Taming the wild web', *TIME Europe magazine*, 14 August, viewed 12 June 2006, http://www.time.com/time/europe/magazine/article/0,13005,901050822-1093679,00.html.

[45] McCullagh, P & Nelder, JA 1983, *Generalized linear models*, Chapman and Hall, London.

[46] McMahon, SM, Miller, KH & Drake, J 2001, 'Networking Tips for Social Scientists and Ecologists', *Science*, vol. 293, pp. 1604-1605 viewed 12 June 2006, http://online.sfsu.edu/~webhead/scipersp.pdf.

[47] Mitchell, M, Hraber, PT & Crutchfield, JP 1993, 'Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations', *Complex Systems*, vol. 7, no. 2, pp. 89-130, viewed 12 June 2006, http://www.santafe.edu/research/publications/workingpapers/93-03-014.pdf.

[48] Murphy, K 1998, 'A Brief Introduction to Graphical Models and Bayesian Networks', viewed 12 June 2006, http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html.

[49] Newman, MEJ 2002, 'Assortative Mixing in Networks', *Physical Review*, vol.89, no. 20, viewed 12 June 2006, http://arxiv.org/PS_cache/cond-mat/pdf/0205/0205405.pdf.

[50] Ng, AY, Zheng, AX & Jordan, MI 2001, 'Link Analysis, Eigenvectors and Stability', *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, August 2001, pp. 903-910.

[51] Nilsson, N, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann, 1998.

[52] Pattison, P & Robins, G 2001, 'Neighborhood-based models for social networks', *Sociological*

Methodology, vol. 32, no. 1, pp. 301-337, viewed 12 June 2006, http://www.psych.unimelb.edu.au/staff/gr/neighborhood.pdf.

[53] Pioch, NJ, Barlos, F, Fournelle, C & Stephenson, T 2005, 'A Link and Group Analysis Toolkit (LGAT) for Intelligence Analysis', viewed 12 June 2006, https://analysis.mitre.org/proceedings/Final_Papers_Files/348_Camera_Ready_Paper.pdf.

[54] Popescul, A & Ungar, LH 2003, 'Structural Logistic Regression for Link Analysis', *Proceedings of KDD Workshop on Multi-Relational Data Mining*, 2003, viewed 12 June 2006, http://www.cis.upenn.edu/~popescul/Publications/popescul03mrdm.pdf.

[55] Popescul, A and Ungar, LH 2004, 'Cluster-based Concept Invention for Statistical Relational Learning', *Proceedings of Conference Knowledge Discovery and Data Mining (KDD-2004)*, 22-25 August 2004, viewed 12 June 2006, http://www.cis.upenn.edu/~popescul/Publications/popescul04clusterbased.pdf.

[56] Prystowsky, J and Gill, L 2005, 'Calculating Web Page Authority Using the PageRank Algorithm', viewed 12 June 2006, http://online.redwoods.cc.ca.us/instruct/darnold/laproj/fall2005/levicob/LinAlgPaperFinal2-Screen.pdf.

[57] Raab, J and Milward, HB 2003, 'Dark Networks as Problems', *Journal of Public Administration Research and Theory*, vol. 13, no. 4. pp 413-439.

[58] Rattigan, MJ and Jensen, D 2005, 'The Case for Anomalous Link Detection', *Proceedings of the 4th international workshop on multi-relational mining*, August 21 2005, pp.69-74.

[59] Read, R 1972, *Graph Theory and Computing*, Academic Press, New York.

[60] Resnick, M 1997, *Turtles, Termites, and Traffic Jams*, Massachusetts Institute of Technology Press, Cambridge.

[61] Ross, S, Westerfield, R, Jordan, B & Firer, C 2001, *Fundamentals of Corporate Finance*, McGraw-Hill, St. Louis.

[62] Roumeliotis, SI & Mataric, MJ 2000, '"Small-World" Networks of Mobile Robots', *Proceedings*

of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, 30 July – 3 August 2000.

[63] Taskar, B, Wong, M-F, Abbeel, P & Koller, D 2004, 'Link prediction in relational data', *Proceedings of Neural Information Processing Systems*, 13-18 December 2004.

[64] Taskar, B, Abbeel, P, Wong, M-F and Koller, D 2003, 'Label and Link Prediction in Relational Data', *Proceedings of the IJCAI Workshop on Learning Statistical Models from Relational Data*, 11 August 2003.

[65] Travers, J & Milgram, S 1969, 'An experimental study of the small world problem', Sociometry, vol. 32, no. 4, pp. 425-443.

[66] Trudeau, RJ 1993, *Introduction to Graph Theory*, Dover Publications, New York.

[67] Underhill, L & Bradfield, D 1996, *IntroSTAT*, Juta and Company, Cape Town.

[68] van den Honert, R 1997, *Intermediate statistical methods for business and economics*, UCT Press, Cape Town.

[69] Wasserman, S & Faust, K 1994, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge.

[70] Witten, I & Frank, E 2005, *Data Mining: Practical Machine Learning Tools and Techniques*, second edition, Morgan Kauffman.

[71] Zanette, DH 2002, 'Dynamics of rumor propagation on small-world networks', *Physical review E*, vol. 65, no. 4, viewed 12 June 2006, http://prola.aps.org/pdf/PRE/v65/i4/e041908.

[72] Zhou, D & Scholkopf, B 2004, 'A regularization framework for learning from graph data', *Proceedings of Workshop on Statistical Relational Learning at International Conference on Machine Learning*, Banff, viewed 12 June 2006, http://www.cs.umd.edu/projects/srl2004/Papers/zhou.pdf.

[73] Zhu, J 2003, 'Mining Web Site Link Structures for Adaptive Web Site Navigation and Search', PhD thesis, University of Ulster, viewed 12 June 2006, http://kmi.open.ac.uk/people/jianhan/thesis.pdf.