Generalizing link prediction:
Collaboration at the University of Antwerp as a case study

Raf Guns
University of Antwerp, IBW, Venusstraat 35, B-2000 Antwerpen, Belgium, raf.guns@ua.ac.be

The link prediction (LP) approach tries to predict links in an unknown network on the basis of a known network. It is argued that LP evaluation can be treated analogous to Information Retrieval evaluation. This characterization entails three generalizations of LP: both appearing and disappearing links can be predicted, LP is not necessarily time-based, and LP is complementary to anomalous link and gap discovery.
Multi-input LP tries to increase precision and recall by having more than one known network as input. These concepts are applied to an informetric case study of collaboration at the University of Antwerp. Performance of different prediction methods is discussed. Furthermore, we establish a small but positive influence for multi-input LP.

Introduction

Social network analysis (SNA) is concerned with describing and explaining social structure by means of network theory. During the last decades, many measures and techniques originally devised for SNA have been successfully applied in other research fields. More recently, they have also been introduced in information science (e.g., Björneborn, 2006; Kretschmer, 2004; Otte & Rousseau, 2002). Indeed, in the field of informetrics the interactions between documentary and/or social entities form an important study object (Wilson, 1999). They can be represented abstractly as networks of citations, collaborations, downloads etc.

The links in social and informetric networks do not appear randomly (see Newman (2003) for an overview of the differences between random and ‘real world’ networks). In the present paper, we explore factors that can be influential on link formation and evolution using an approach known as *link prediction* (LP). The problem that LP tries to tackle is this: given a snapshot of an evolving social network, how can one predict which new links will appear in some future snapshot of the same network? This and related questions have been studied by, among others, (Huang, 2006; Huang et al., 2005; Popescul & Ungar, 2003; Liben-Nowell & Kleinberg, 2003; Liben-Nowell & Kleinberg, 2007). As in previous studies, we will only study the prediction of links between existing vertices, – links to new vertices are outside the scope of this paper.

The potential of LP can be seen both on a theoretical and a practical level. On a theoretical level, LP may help to test and validate the myriad of network (evolution) models, – given enough data, such models can be tested with LP. Practically, one can imagine many possible applications: LP can be used to recommend related items in digital libraries, to suggest candidates for collaboration or relevant references in research. Further on in the paper, we outline an approach (referred to as ‘multi-input LP’) that could help university policy makers determine some of the factors that contribute to policy goals such as collaboration or internationalization.

Most research around LP consists of two broad steps.
(i) Some predicting method is applied to a training network which results in a prediction of possible new links. This method can be simple (e.g., implementing a proximity measure) or elaborate. Intuitively, it makes sense to assume that those vertices that are already close in some sense (e.g. they share many friends) will likely form a link at some later point.
(ii) The predictive power of the method is evaluated by comparing the prediction to an actual later snapshot (the test network). This is represented graphically in Figure 1.

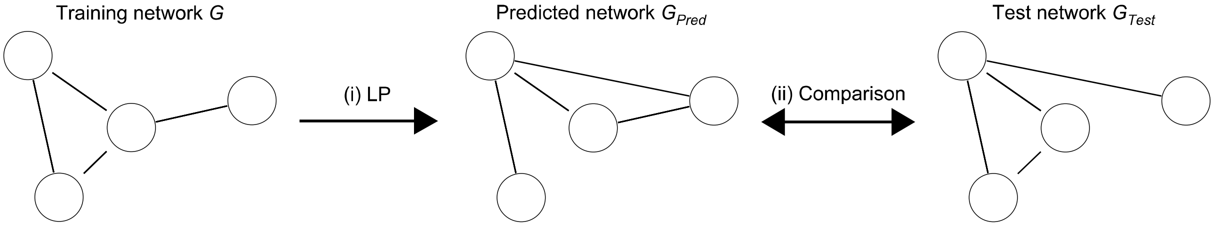

Figure 1. An LP method (i) predicts (the links in) a network G_{Pred} on the basis of the training network G and (ii) can be evaluated by comparing with G_{Test}

The training network $G = (V, E)$ consists of a set of nodes or vertices V and a set of links or edges E. Link prediction can then be formally characterized as a function LP: $V \times V \rightarrow \mathbf{R}$, that maps a pair of vertices to a real-valued likelihood score $w \in [0,1]$. The score w expresses the likelihood that a link between these two vertices exists in the predicted network. Note that the link prediction function only indirectly specifies a new, 'predicted' network, e.g. if combined with a threshold value for the likelihood score. The predicted network then consists of all links whose associated likelihood score exceeds the threshold value. Moreover, since the function only assigns a score to vertex pairs from the training network, (links to) new vertices cannot be predicted.

LP as originally described is just one possible case in a larger 'family' of approaches. The next section discusses three ways in which LP can be generalized. These open the gate for so-called 'multi-input LP', which is based on more than one network. We explore the potential of both single-input and multi-input LP on a collaboration case study. The last section contains the conclusions.

Three generalizations of link prediction

How can an LP method be evaluated? We now assume that a threshold has been applied, such that E_{Pred} is the set of predicted links whose w exceeds the threshold and $G_{Pred} = (V, E_{Pred})$. Likewise, G_{Test} denotes the test network, and E_{Test} denotes the set of links in G_{Test}. If the links are unweighted, comparison between the predicted network and the test network is fairly straightforward. In fact, it can borrow a lot from standard Information Retrieval (IR) measures: E_{Test} is similar to 'what is relevant' and E_{Pred} is similar to 'what is found'. Just like in IR, there are four sets to consider (Van Rijsbergen, 1979), shown in Table 1 as a contingency table. ($\neg A$ refers to the complement of A.)

Table 1. Four possible relations between the link set of the predicted network and the link set of the test network

$E_{Pred} \cap E_{Test}$	$\neg E_{Pred} \cap E_{Test}$
$E_{Pred} \cap \neg E_{Test}$	$\neg E_{Pred} \cap \neg E_{Test}$

This means we can determine (the trade-off between) precision and recall for each predicting method. The overlap between E_{Pred} and E_{Test} can be determined with a similarity measure like Dice's coefficient, which is close to 1 if both recall and precision are close to 1; if either recall or precision are low, it rapidly approaches 0. Note that we assume here that G_{Pred} is an unweighted network. If G_{Pred} were weighted, more sophisticated comparison methods would be necessary.

In view of the analogy to IR, LP can be considered as a function that tries to maximize the number of elements in the intersection of E_{Pred} and E_{Test} (upper left quadrant of Table 1). We now introduce three generalizations that are based on this abstract characterization.

Anomalous link and gap discovery

LP provides us with a likelihood score per link for an (in principle) unknown network on the basis of a known network. The likelihood score can be employed to determine the most likely links but also to detect outliers in two complementary ways: anomalous link discovery and anomalous gap discovery.

The links in the test network with the lowest likelihood scores form unexpected connections between vertices that are, in some way, 'far apart'. On the one hand, such a link may be 'unstable' and likely to disappear in the future. On the other hand, this may also signify a special situation.

Rattigan & Jensen (2005) have explored the latter possibility and named their approach *anomalous link discovery* (ALD). The basic idea is that such unexpected outliers are often the most interesting links in the network. It is striking that their application to co-authorship in the DBLP database automatically discovered an error where two authors sharing the same name were combined into one. ALD could, for instance, also be used in webometrics to discover 'transversal' hyperlinks that cross knowledge domain boundaries (Björneborn, 2006).

Vertex pairs that have high likelihood scores but do not form links in the test network constitute 'anomalous gaps'. Again, this may be due to chance – a link is likely to (re-)appear in the future – or they may indicate some sort of boundary between (groups of) vertices. For example, if authors A and B have cited each other's work many times, we may derive a strong likelihood that citation links between A and B will also occur in the test network. If such links do not occur, then this may indicate that A's (or B's) research subject has changed.
We are not aware of any studies of anomalous gaps, presumably because anomalous gap discovery (AGD) is more problematic than ALD: in social networks, the absence of a link is the default – i.e., social networks are sparse!

LP, ALD and AGD each focus on a different cell from Table 1, as illustrated in Table 2. The fourth quadrant is generally uninteresting, apart from the fact that its size can be used in a similarity measure. Therefore, it is not associated with a particular approach.

Table 2. Relation between link prediction and anomalous link and gap discovery

$E_{Pred} \cap E_{Test}$ link prediction	$\neg E_{Pred} \cap E_{Test}$ anomalous link discovery
$E_{Pred} \cap \neg E_{Test}$ anomalous gap discovery	$\neg E_{Pred} \cap \neg E_{Test}$

Appearing and disappearing links

LP is typically applied in a dynamic context: it presupposes that changes may occur in a network over time. Nevertheless, most research only tries to predict new links and does not pay attention to disappearing links. While the former part – which currently nonexistent links are likely to appear? – is useful in itself for applications such as recommendation systems (Huang et al., 2005), we argue that the latter part – which current links are likely to disappear? – is equally important when, for instance, studying the evolution of a social network (Burt, 2000). Put more generally, we are concerned with the following problem: given a snapshot of existing links, which vertex pairs are most likely to have links, be they existing or new, in a future snapshot? Note that this question envelopes both the appearance and the disappearance of links.

We only consider those vertices that are present in both training and test network, and that are connected to a predetermined minimum number of other vertices. Following Liben-Nowell & Kleinberg (2003, 2007), we introduce two parameters, $\kappa_{training}$ and κ_{test}, and only look at those vertices with a minimum degree of $\kappa_{training}$ in the training network and a minimum degree of κ_{test} in the test network. For the case study in this article we take $\kappa_{training} = \kappa_{test} = 1$, thereby only ignoring all isolates.

LP on another basis than time

The canonical case of LP is time-based: G and G_{Test} represent the same network at different points in time. One can, however, also imagine cases where one wants to predict unknown relations between entities on the basis of another, known relation. For instance, one could hypothesize that friendship bonds in a social network are mainly determined through homophily. On the basis of training networks involving race, gender, social class etc. one could then try to predict the existing friendship network without access to an older snapshot of the same network. An example in collaboration is studied further on.

There are no mathematical constraints on the relation between the training network on the one hand and the predicted and test network on the other hand, other than that they all have the same vertex set V. It should, however, be stressed that it only makes sense to search for a prediction function between networks where a correlational or causative relation can reasonably be assumed.

Multi-input LP

Once we acknowledge that the training network in LP is not necessarily an older snapshot of the test network, a new question arises: can we combine two or more networks for training? This approach is one of several future directions mentioned by Liben-Nowell & Kleinberg (2007); they propose fine-tuning the predictions by taking institutional affiliation and geographic location into account. We will refer to this variant as *multi-input LP*, while the classical variant will be referred to as *single-input LP*.

Multi-input LP takes as its input a set $S = \{G_1, G_2,\ldots, G_t\}$ where $G_i = (V, E_i)$ $(1 \leq i \leq t)$. Assume now that we want to determine the likelihood score w of a link between vertices u and v. This can be as simple as applying LP to u and v for each network in S and taking the average score:

$$w = \frac{1}{t}\sum_{i=1}^{t} w_i \qquad (1)$$

A slightly more refined variant assigns a weight γ to each network in S, such that more important networks have greater impact:

$$w = \sum_{i=1}^{t} \gamma_i w_i \ \text{ with } \sum_{i=1}^{t} \gamma_i = 1 \qquad (2)$$

Of course, other ways of determining a value for w in multi-input LP are possible, but more research is necessary to establish their merit.

Case study: collaboration at the University of Antwerp

As a practical case study of both single-input and multi-input LP, we have studied the collaboration network of researchers at the University of Antwerp. The University of Antwerp (UA) is the third largest university in Flanders, the northern part of Belgium. It was founded in 2003 as a merger of the three smaller universities that were situated in Antwerp (UIA, RUCA and UFSIA).

Prior to 2003 the three universities did already cooperate on a number of domains, including library automation. One example is the *Academic Bibliography* (AB), in which all academic publications by UA staff are recorded from 1991 until the present. In informetric studies, a citation database like Web of Science or Scopus is a more common data source, but the use of a local database has some benefits:

- The AB has a much larger coverage of publications authored at the UA than citation databases do. This is especially beneficial for areas of research that are traditionally less well-represented in such databases, like the Humanities. Indeed, all publications in ISI-covered journals are automatically inserted, in addition to all academic publications a researcher himself chooses to submit. Since these data can be used in e.g. promotions, it is in the researchers' own interest to submit all relevant publications. Experience has taught that most keep their AB record well up-to-date.
- Each author is uniquely identified by a URI. Thus, we avoid problems of homonymy (different authors with the same name) and synonymy (one author with several names; a female author may for example have published under her maiden name and her husband's name).
- The most important factor, however, is the availability of 'local' information, i.e. information regarding each researcher's department(s) and location. This information will be exploited for multi-input LP.

In our case, the test network is the collaboration network formed between January 2004 and December 2006, where vertices are authors affiliated with the UA and weighted links represent the number of co-authored papers during this period. Since we wanted to use the preceding time period (2001–2003) as one of the training networks, we only consider the 1102 authors that have at least one co-author in 2001–2003 and at least one co-author in 2004–2006 ($\kappa_{training} = \kappa_{test} = 1$). Details are provided in Table 3.

Table 3. Vertices and links in time periods 2001–2003 and 2004–2006

	Number of non-isolate vertices	Number of links
2001–2003	1366	3930
2004–2006	1548	4825
Under study	1102	6921

For the current investigation, we use the methods that are listed in Table 4. Most of these have been tested previously in LP. Since most methods are neighbor-based, we introduce the following notation: Γ(x) denotes the set of neighbors of vertex x. The 'Equal' method just makes a copy of the training network. If the training network is weighted, links with a higher weight get a higher likelihood score. This method has not been used before, since most studies only try to predict new links in the test networks and, by definition, the Equal method does not include any new links. The following methods should be normalized in order to get scores between 0 and 1: Equal, Adamic/Adar, Common Neighbors, and Preferential Attachment. Although the proximity measure defined by Katz (1953) seems promising, we have not included it because of its computational complexity.

Table 4. LP methods, where Γ(x) is the set of neighbors of vertex x

Name	Value of w for vertices u and v
Equal	weight of link between u and v
Adamic/Adar (Adamic & Adar, 2003)	$\sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log(\lvert \Gamma(z) \rvert)}$
Common Neighbors	$\lvert \Gamma(u) \cap \Gamma(v) \rvert$
Cosine	$\frac{\lvert \Gamma(u) \cap \Gamma(v) \rvert}{\sqrt{\lvert \Gamma(u) \cdot \Gamma(v) \rvert}}$
Graph Distance	1 / length of geodesic between u and v
Jaccard	$\frac{\lvert \Gamma(u) \cap \Gamma(v) \rvert}{\lvert \Gamma(u) \cup \Gamma(v) \rvert}$
Overlap	$\frac{\lvert \Gamma(u) \cap \Gamma(v) \rvert}{\max(\Gamma(u), \Gamma(v))}$
Preferential Attachment	$\lvert \Gamma(u) \rvert \cdot \lvert \Gamma(v) \rvert$

Single-input LP in the case study

We use time-based LP as an example of single-input LP: the training network is based on data from 2001–2003 and the test network is based on data from 2004–2006. All vertex pairs are ranked in decreasing order of their corresponding likelihood score w. $E_{Pred[i]}$ denotes the i first vertex pairs. Then, recall and precision of the first i (i = 1, 2, …, n) predictions are calculated as follows. For recall,

$$R_i = \frac{\lvert E_{Pred[i]} \cap E_{Test} \rvert}{\lvert E_{Test} \rvert} \qquad (4)$$

and for precision,

$$P_i = \frac{\lvert E_{Pred[i]} \cap E_{Test} \rvert}{\lvert E_{Pred[i]} \rvert} = \frac{\lvert E_{Pred[i]} \cap E_{Test} \rvert}{i} \qquad (5)$$

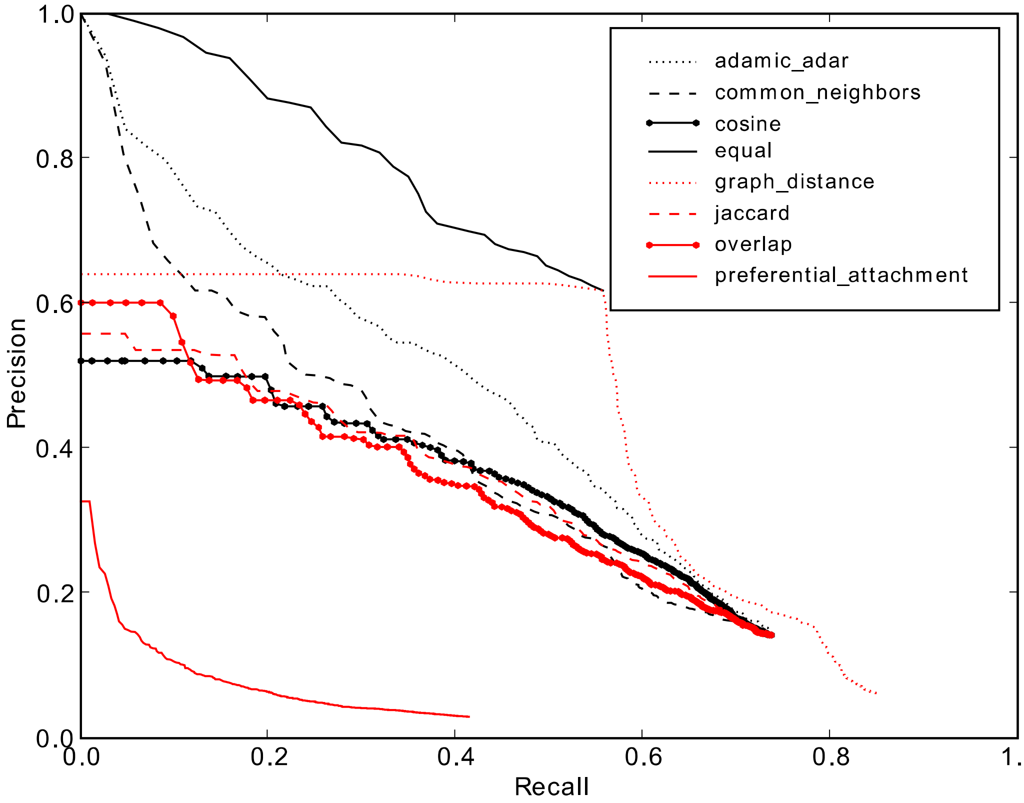

Figure 2. Recall–precision curves of different methods applied to the 2001–2003 snapshot

We thus construct the recall–precision curves shown in Figure 2. Recall–precision curves of LP have previously been employed by (Kashima & Abe, 2006; Popescul & Ungar, 2003). We note that two or more predictions may have the same likelihood score. Thus, ties are possible and ranking in those cases is arbitrary, but experimentation has shown that this has hardly any effect on the resulting curve.

We find that no single method achieves 100% recall: if two vertices belong to different components in the training network, none of these methods can predict an edge between them. Note that the study by Liben-Nowell & Kleinberg (2003, 2007) deliberately restricted predictions to the largest component, which automatically increases the potential recall.

The Equal method is included since we try to predict both recurring and new links. Its maximum recall is limited since it does not predict any new links, but in terms of precision, it outperforms any of the other methods.

The Adamic/Adar method seems to be the best neighbor-based predictor because it prefers ‘rare’ (i.e. having low degree) neighbors over common ones. Preferential Attachment is a very poor predictor, although its predictions are still significantly better than random predictions. The other neighbor-based methods (Common Neighbors, Cosine, Jaccard and Overlap) are all clearly less precise although, surprisingly, the very simple Common Neighbors method still fares better than the other, more sophisticated ones.

Contrary to Huang et al. (2005), we find relatively good performance of the Graph Distance method. This is, however, largely due to the fact that it also ‘predicts’ those links already present in the training network (geodesic length = 1). Geodesics of length 2 or greater are a much worse predictor, as can be seen by the rapid decline in precision around 55% recall. Note that this method’s precision is worse for connected vertices than the Equal method. Most likely, this is because this method does not take link weight into account.

It is possible that some combination of these methods (for instance, Equal combined with Adamic/Adar) would be able to achieve even better recall and precision. Due to space limitations we do not pursue this line of inquiry here.

Multi-input LP in the case study

Even the best single-input prediction methods make many faulty predictions and do not include many correct ones. We now turn to the question if multi-input LP can be used to increase precision and/or recall. We explore this question using a training set *S* containing three networks, which we will now describe.

The first network is the *collaboration network* of authors at the UA between 2001 and 2003 studied before. Reuse of the training network of the preceding section allows direct comparison with single-input LP.

The second network is the *department network* where vertices represent authors that are linked if they belong to the same department. At the UA, each person is affiliated to one or more departments that each have a unique identifier. For instance, the identifier “APSW” refers to the faculty of Social Sciences, “APSWP” refers to the department of Political Sciences at said faculty. We have tried to assign authors to the department with the highest degree of granularity, e.g., “APSWP” is preferred to “APSW”. Subsequently, all persons associated with the same identifier are linked in this network.

The third and last network is the *physical location network* where vertices represent authors that are linked if they work in physical proximity. The UA is spread over four campuses, each with several buildings. People working close together are more likely to collaborate, even if they do not belong to the same department, and vice versa, people from the same department working at different locations are less likely to collaborate. Where possible, an author is assigned to a physical location (campus + building + floor) and subsequently, all authors associated with the same location are linked.

Unfortunately, we could not access all required data about each author. Therefore, the department network contains only 98% of all authors and the location network only 57%. Details are provided in Table 5. These two networks are also atypical in that they are based on a hierarchy of vertices (Goussevskaia et al., 2007): all vertices with a common characteristic are linked. In other words, these training networks have much higher density than the test network. They would make poor training networks for single-input LP; especially precision would be very low. They do, however, have the potential of increasing both precision and recall in combination with a ‘better’ network like the one from 2001–2003.

Table 5. Data availability of department and physical location of authors

	Available for … authors	Number of groups
Department	1085	38
Physical location	632	132

For this reason, we determine the likelihood score *w* of a link between *u* and *v* as a weighted average:

$$w = \frac{4}{5} w_{collaboration} + \frac{1}{10} w_{department} + \frac{1}{10} w_{location} \qquad (6)$$

The more important collaboration network has a weight of 0.8, while the auxiliary location and department networks each have a weight of 0.1. Experimenting with tweaking the weights in formula (6) yielded the following results:

The location and department networks complement each other. Leaving either out decreases both precision and recall.

The weights as assigned in formula (6) are near optimal. If the proportion of the collaboration network becomes lower, precision is drastically lowered as well.

The resulting recall–precision curves are shown in Figure 3. In comparison to Figure 2, maximum recall is increased for most methods, but the precision of these ‘extra’ predictions is very low. Observe for instance the sharp decline in precision of Equal after the predictions that were also present in the preceding section. In terms of precision, most methods seem hardly affected in comparison with single-input prediction. Where the effect is observable, however, it is an improvement.

Figure 4 shows that, for most methods, multi-input LP forms a very slight improvement over single-input LP in terms of average precision. Precision of the results with the highest likelihood score can however be improved significantly, especially for the Cosine, Graph Distance, Jaccard and Overlap methods.

Conclusions

In this article, we have explored the analogy between LP and IR, and between LP evaluation and IR evaluation. This has lead to a number of generalizations and illustrates that recall–precision curves are an adequate tool for LP evaluation. By trying to predict the entire network (both appearing and disappearing links), it can be seen that the very basic Equal method easily outperforms the other methods in terms of precision.

As a second generalization, we have discussed the natural relations between LP, ALD and AGD. Finally, we have argued that the training network in LP does not have to be an older snapshot, which opens the door to multi-input LP. In our case study, multi-input LP offers only limited improvements over single-input LP, although it seems to benefit some methods more than other, especially for the highest ranked predictions. Our case study is limited in that the two additional networks are rather dense and imprecise. In further research, we will explore how more typical networks can be combined for multi-input LP and how ALD and AGD complement the LP approach in practice.

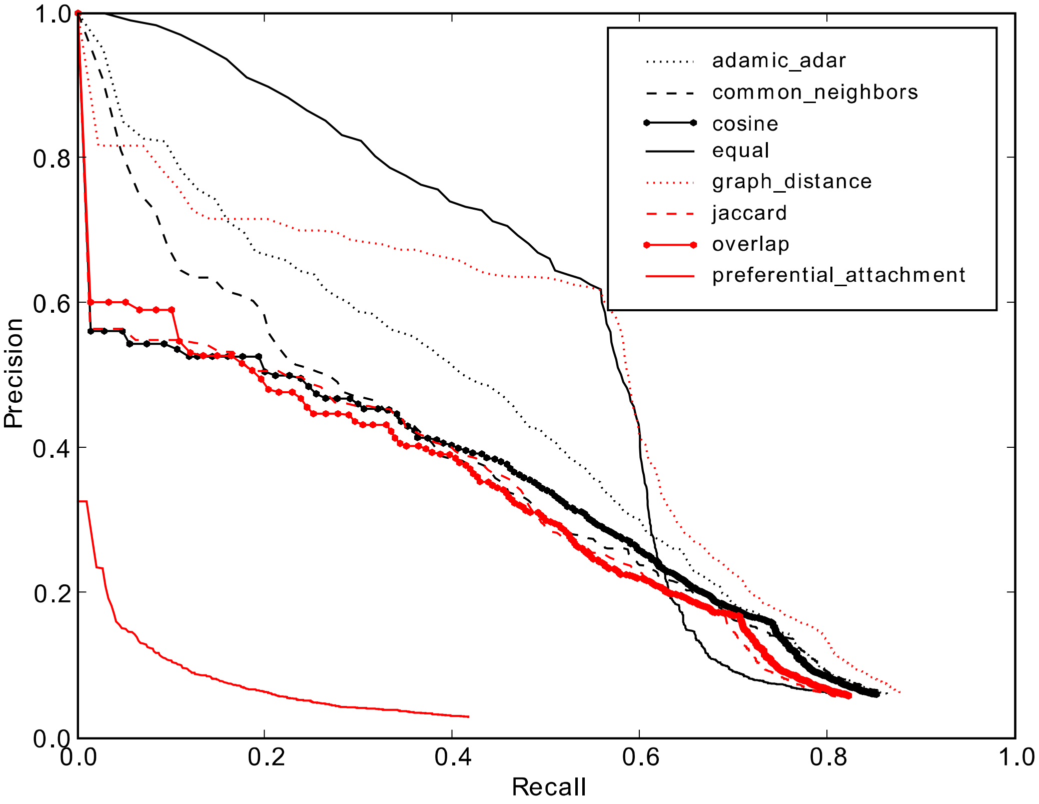

Figure 3. Recall-precision curves for multi-input LP

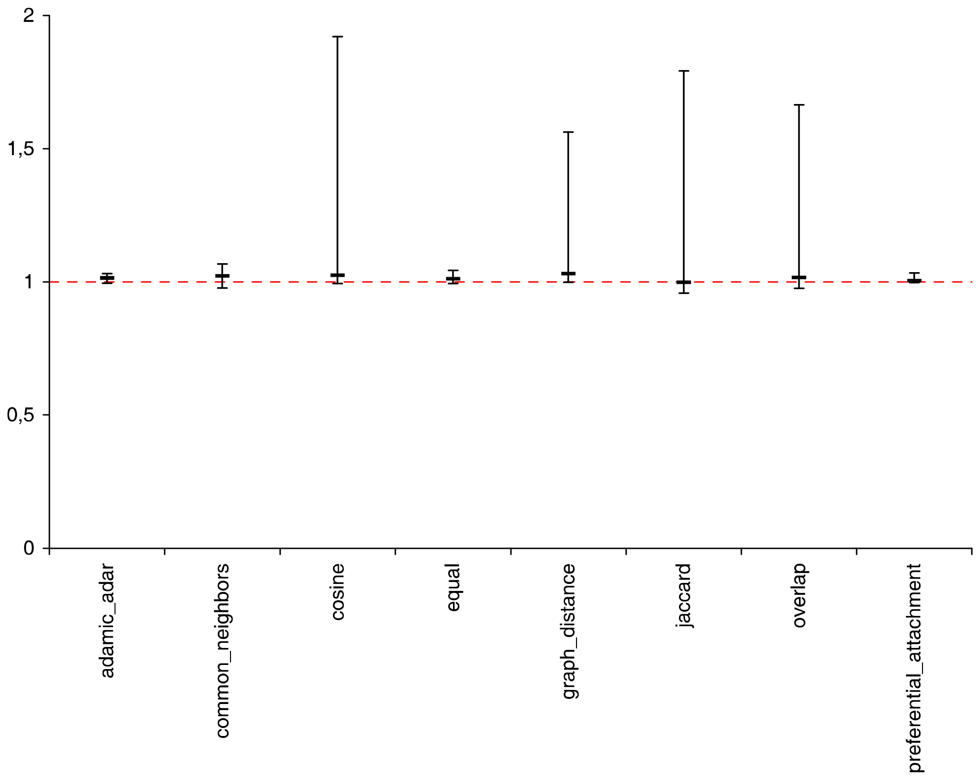

Figure 4. Average precision ratio of multi-input LP relative to single-input LP (dashed red line), where error bars indicate best and worst ratio

Acknowledgment

I am grateful to Prof. Ronald Rousseau, whose insightful comments helped to improve the paper significantly. I also thank the anonymous reviewers for some valuable suggestions.

References

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

Björneborn, L. (2006). 'Mini small worlds' of shortest link paths crossing domain boundaries in an academic Web space. *Scientometrics*, *68*(3), 395–414.

Burt, R. S. (2000). Decay functions. *Social Networks*, *22*(1), 1–28.

Goussevskaia, O., Kuhn, M., & Wattenhofer, R. (2007). Layers and hierarchies in real virtual networks. In A. Cuzzocrea (ed.), *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence*, 89–94.

Huang, Z. (2006). Link prediction based on graph topology: The predictive value of the generalized clustering coefficient. In *Proc. of KDD '06 Workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD2006)*.

Huang, Z., Li, X., & Chen, H. (2005). Link prediction approach to collaborative filtering. In *Proc. of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05)*, 141–142. NY: ACM Press.

Kashima, H. & Abe, N. (2006). A parameterized probabilistic model of network evolution for supervised link prediction. In *Proc. of the 2006 IEEE International Conference on Data Mining (ICDM 2006)*, 340–349.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, *18*(1), 39–43.

Kretschmer, H. (2004). Author productivity and geodesic distance in co-authorship networks, and visibility on the Web. *Scientometrics*, *60*(3), 409–420.

Liben-Nowell, D. & Kleinberg, J. (2003). The link-prediction problem for social networks. In *Proc. of the 12th International Conference on Information and Knowledge Management (CIKM)*, 556–559.

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

Newman, M. E. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256.

Otte, E. & Rousseau, R. (2002). Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, *28*(6), 441–453.

Popescul, A. & Ungar, L. H. (2003). Structural logistic regression for link analysis. In S. Džeroski et al. (eds.), *Proc. of the 2nd International Workshop on Multi-Relational Data Mining (MRDM-2003)*, 92–106.

Rattigan, M.J. & Jensen, D. (2005). The case for anomalous link discovery. *ACM SIGKDD Explorations Newsletter*, *7*(2), 41–47.

Van Rijsbergen, C.J. (1979). *Information Retrieval*. London: Butterworths.

Wilson, C.S. (1999). Informetrics. *Annual Review of Information Science and Technology*, *34*, 107–247.