Contents

1 Introduction 1

2 Literature Review 2

2.1 Similarity Measures 3

2.2 Learning based methods 4

2.2.1 Feature based Classification 4

2.2.2 Probabilistic Models 5

2.2.3 Kernel based Models 5

2.2.4 Matrix Factorization 6

2.3 Application: Weighted Networks 6

2.4 Current trends 7

3 Summary & Conclusion 7

1 Introduction

Social networks are structures modelling humans as actors or nodes and the ties established between them become edges in a global, dynamic graph [2]. The focus of social network analysis is understanding and predicting the relationships between human entities, rather than studying them individually. In the last decade, social network analysis has flourished due to the prevalence of online social networks e.g. Twitter, Facebook, LinkedIn, which became a trove of data available to study explicitly modelling the edges as friendships or follow-actions [3].

The *link prediction problem* in social network analysis refers to predicting the probability of a future link given that there is no current one. Its applications are numerous, from recommender system design to biological problems. This problem comes in two flavours [4]: *network completion* and *future link prediction.* Network completion applies link prediction in incomplete graph snapshots to infer the missing links, while the future link prediction uses link prediction methods to understand network evolution. One of the first papers to formally define this problem is by Liben-Nowell & Kleinberg [1]: given a network $G(V, E)$ where $e = (u, v) \in E$ at a particular timestamp t, we can define the $G[t, t']$ as the subgraph with all edges with timestamp between t ant t'. Then, given $G[t_0, t'_0]$ the goal is to predict the edges in $G[t_1, t'_1]$.

Link prediction applications span multiple disciplines such as biology, business, economics, academical interest focusing also on particular aspects specific to the application. An application with high economic impact is recommender systems such as finding new friends [5, 6], as well as collaborators and business partners [7, 8, 9]. They can also be used for automatic hyperlink creation [10, 11], for detecting potential criminal connections [12, 13]. In Bioinformatics this is also an active research interest [14], being used primarily to predict protein-protein interaction [15]. Although this list is far from comprehensive, we would like to underline the potential of link prediction in collaborative filtering (bipartite networks of type user-item). If the network topology is known, as well as various actor attributes, one could infer the propensity of an actor to purchase for instance a certain brand (principle: if my friends like it, I should try). However, only recently have there been researched methods dealing with such a large and dynamic graph

[16].

There have already been published numerous extensive literature reviews for link prediction, starting with the pioneering work from Liben-Nowel & Kleinberg [1] and most recently the comprehensive review by Wang & al. [17]. While overlap is inevitable, we are reviewing a narrower subfield (static social networks) and discuss in more detail newer published works (after 2015) to establish new areas of research. Static networks have been chosen because research focused on these has proven most fruitful (they aggregate more information than temporal-based ones). The main criteria for including a paper in this review is related to how much attention has the academic community paid to it[1]. For the last few years (starting with 2015), general trends were identified and key ideas critically reviewed.

The structure of the remainder of this literature survey follows a similar outline as [17]. In section 2 we introduce the general framework for link prediction and present the main techniques applied for link prediction classified in two main categories: unsupervised similarity measures in subsection 2.1, learning based methods in subsection 2.2. An application in a particular type of network, we discuss link prediction in weighted networks in subsection 2.3. We continue to discuss current trends and how they may solve the link prediction challenges in subsection 2.4 and conclude in section 3.

2 Literature Review

We start by introducing the general framework for evaluating link prediction techniques [1], to be able to critically evaluate some studies discussed in following sections. Let $t0 < t0' < t1 < t1'$ then, given access to $G[t0, t0']$ (also called training interval) the link prediction algorithm must output the links not $inG[t0, t0']$ but present in $G[t1, t1']$ (also called the test interval). The general training/testing framework for similarity measures and learning based models is detailed in figure 1. While the training phase is relatively straightforward (either compute a ranked list of high scored pairs of nodes or some form of loss minimization), the evaluation of the results is problematic primarily due to class imbalance (there are much more nonexistent links than existent). Most papers dealt with this by employing ROCs (Receiver Operating Curves) and measuring AUC (Area under curve), whose benefits compared to ”top K precision” are comprehensively explored by Yang et al. [18]. We argue that, since class imbalance is one of the defining problems for link prediction [18], the evaluation methodology is an important issue in all research on this subject.

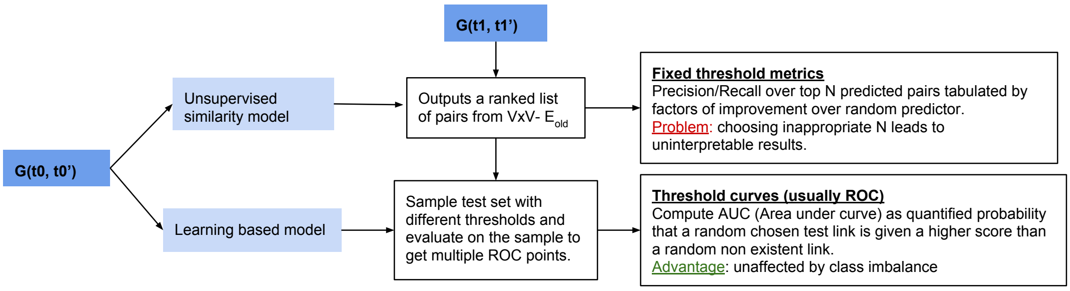
G(t1, t1')
Unsupervised similarity model
Outputs a ranked list of pairs from VxV- E_{old}
Fixed threshold metrics
Precision/Recall over top N predicted pairs tabulated by factors of improvement over random predictor.
Problem: choosing inappropriate N leads to uninterpretable results.
G(t0, t0')
Learning based model
Sample test set with different thresholds and evaluate on the sample to get multiple ROC points.
Threshold curves (usually ROC)
Compute AUC (Area under curve) as quantified probability that a random chosen test link is given a higher score than a random non existent link.
Advantage: unaffected by class imbalance

Figure 1: Link prediction evaluation (original diagram by D. Cremarenco)

[1] e.g. did their results affect future research? did a significant number of other papers consider its results?

2.1 Similarity Measures

A similarity measure between nodes computes the likelihood of their association [17] and can be derived from two sources: actor attributes (e.g. text-similarity between profiles, mailing lists [19], location [20], affiliation [21] etc.) and topological measures (based on the network graph). The actor attributes are likely to have high prediction power [22] (e.g. overlap in research interests for co-authorship networks was top prediction feature in [22]). Methods incorporating node attributes are also less likely to face the cold-start problem: where new actors are introduced in the graph, but no links are known [16]. However, such attributes may be difficult/impossible to obtain (e.g. Zheleva & al. [23] use family links explicitly included in special network datasets (Catster) unlike any other dataset).

Most research effort has concentrated on topological measures (extracting information from the graph structure only), which come in three flavours: **neighbourhood methods** (potential links at geodesic distance = 2), **short path methods** (potential links found at shortest path(geodesic) distance ≤ fixed constant) and **global methods** (considering all the potential links in the graph). Unlike other reviews [17, 24], we considered short path-based methods as a third category, primarily because their complexity is vastly lower than the global-based ones [17]. We expose some of the most influential similarity measures in these categories in Table 1, Table 2, Table 3 with the following standard notations $\Gamma(x)$ is the set of neighbours of node x, $|\Gamma(x)|$ is the size of this set and A/D is the adjacency matrix, degree matrix respectively.

Name	Score(x, y)	Observations
Common Neighbours (CN)[25]	$\|\Gamma(x) \cap \Gamma(y)\|$	Very popular due to its simplicity and good results [1, 26].
Jaccard's Coefficient (JC)[1]	$\frac{\|\Gamma(x)\cap\Gamma(y)\|}{\|\Gamma(x)\cup\Gamma(y)\|}$	Normalized CN, often used as baseline, no study indicated good performance.
Adamic/Adar (AA)[19]	$\sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{log(\Gamma(z))}$	Initially used for webpage similarity, with very good results as a topological measure [1, 27, 26].
Preferential Attachment (PA)[25]	$\|\Gamma(x)\| \cdot \|\Gamma(y)\|$	Useful in disassortative networks [28], but otherwise weak.
Resource Allocation (RA)[26]	$\sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\Gamma(z)}$	Similar to AA, but decreases contribution of high degree nodes more than AA, with better performance in high-degree networks.

Table 1: Overview of local node-based similarity measures

Name	Score(x, y)	Observations
Shortest Path (SP)	length of shortest path	Most basic similarity measure [1], very low performance.
Local Path Index (LPI)[29]	$LP = A^2 + \alpha A^3$	Uses local paths of l ≤ 3, with adjustment factor α applied to paths of l=3. Improvement over CN and lower complexity than Katz.
FriendLink(FL)[30]	$\sum_{i=2}^{l} \frac{1}{i-1} \frac{\|paths^{i}_{x,y}\|}{\prod_{j=2}^{i}(n-j)}$	Uses all paths with max l length with attenuation factor to provide faster and more accurate prediction.
PropFlow (PP) [27]	$PF(a, z)\frac{w_{zy}}{\sum_{k\in\Gamma_x} w_{zk}}$	Probability that a restricted random walk of length max l using link weights as transition probabilities starts at x and ends at y. We compute PF(a, z) recursively as the score of the previous node on the random walk to current node z.

Table 2: Overview of short-path based similarity measures

Name	Score(x, y)	Observations
Katz$_\beta$ [31]	$\sum_{l=1}^{\infty} \beta^{\|} paths_{xy}^{l}\|$	Counts all paths between the nodes, exponentially decreasing contributions of longer ones (β^l). Computationally expensive (faster method in O(n+m) in [32]), but high accuracy [1].
Hitting Time (HT)/Commute time (CT)[1]	$H_{x,y}$ / $H_{y,x}$	$H_{x,y}$ is the expected number of steps required for a random walk from x to reach y. Low performance due to potential effect of a far node z with high stationary probability [1]
SimRank[29]	$\gamma \frac{\sum_{a\in\Gamma(x)}\sum_{b\in\Gamma(y)} SR(a,b)}{\|\Gamma(x)\|\cdot\|\Gamma(y)}$	Idea: similar nodes are connected if they are neighbours with similar nodes. Very high complexity, thus limited on large networks.
Rooted PageRank[1]	$RPR = (1-\beta)(1-\beta D^{-1}A)^{-}1$	Represents stationary probability of y in a random walk that returns to x with probability β and moves on with $1-\beta$. Improvement of hitting time, but still lower than Katz.

Table 3: Overview of global similarity measures

2.2 Learning based methods

We start examining the learning based link prediction techniques by outlining the comparative advantages of supervised methods. Lichtenwalter & al [27] note that supervised models can model interdependency relations between simpler similarity measures, can reduce variance by employing ensemble methods and, most importantly, can address the class imbalance problem that unsupervised methods are oblivious to. After the model is trained, link prediction becomes a binary classification problem, with an extreme class imbalance problem (studied in [27, 18]). Undersampling is usually employed, but if the class distribution does not match the real one, the results are usually uninterpretable, no conclusion can be drawn from results based on manicured datasets.

2.2.1 Feature based Classification

The most intuitive supervised learning model is described first by [22], where multiple similarity measures are used as features in a binary classification model. They used both actor attributes and topological measures (9 in total) to predict links in a co-authorship network e.g. keyword match count and shortest distance. The radias basisc function (RBF) kernel SVM is found to have the best performance, but the results are unaccountable since they have undersampled the test set to balance.

An interesting approach is taken by [21] where feature candidates are generated by constructing join queries on the database model of the data and on average 40 features are selected by statistical model selection criteria using all tables. However, they too undersample test data to balance.

The final analysed feature based supervised model is HPLP+, developed in [27], which built on the idea developed in [22]. They use 16 topological based features with bagged random forests and show that the AUC is by far superior to any considered unsupervised similarity measure (AA, Katz, PropFlow etc.) on a phone call network and a physics academic collaboration network. They study class imbalance in depth and obtain these results by undersampling without modifying the class distribution.

As a final note, in each of these studies, the classification model had little or no impact on the results, save for using some form of ensemble methods: bagging or random forests. The feature set seems to have the highest value, especially if node attributes are used ([20] shows how location features improve the accuracy significantly even if the node pairs considered are 15% of the total number).

2.2.2 Probabilistic Models

In probabilistic models, the algorithm computes the probability of nodes being connected and either uses it as a link prediction method directly [33] or as a feature in a supervised model[34]. We refrain from comparing these methods, except for saying that they are suitable only for small datasets because they do not evaluate on the same datasets and the approaches taken are radically opposed.

Clauset & al. [33] propose a model that identifies the hierarchical structure of the network where each node has a probability p_r associated. Then the probability of two nodes being connected is equal to the probability of their least common ancestor in the dendrogram. They create multiple probabilistic dendrograms of the network and establish the missing link as the one with high average probability of connection. This method appears to have very good results in small networks but is expensive in larger ones.

Kashima & al [35] propose a parametrized probabilistic evolutionary model of the network by modelling probabilistic flips of the value of the edge label based on copy-paste inspired mechanism of edges. A random edge is copied from l to m based on the probability w_{lm}. Both the W probabilities and edge labels are then trained using Expectation Maximization (EM). Despite limitations in large networks, the model has good results in small ones and by testing with Spearman's rank coefficient, the predicted links are very different than what typical methods output.

Lastly, Wang & al. [34] compute the co-occurrence probability of two nodes by first defining the local neighbourhood set of the node and then learning a maximum entropy Markov Random Field model that estimates the joint probability of the nodes in the set. This probability is then used alongside topological and semantic features to train a Logistic Regression classifier. Unfortunately, they only contribute part of their negative instances (10:1), so the results are unreliable.

2.2.3 Kernel based Models

Kernel based methods are used in link prediction to provide a way of specifying similarity between nodes by taking into consideration all the paths between them [36]. A kernel method takes as input the training set's adjacency matrix (A) or Laplacian matrix (L) and applies a transformation (say F, similar to a function on matrices) that outputs another matrix W ($w_{i,j}$ is meaningful regarding the similarity of nodes i and j). Kunegis et al. [37] generalize the problem of finding F by describing a way to learn it by mathematically minimizing the Frobenius norm (F is a spectral graph transformation, A the training set adjacency matrix, B is for the testing set.):

$$min_f ||F(A) - B||_f$$

By using the eigenvalue decomposition of $A = U \Lambda U^T$, further reduced to a chosen rank K, they reduce the above formulation to a one dimensional regression problem with runtime depending on K. Traditionally [38], the graph kernel F is not learned, but chosen from a set of standard kernels, extensively reviewed in [38]. We summarize some of these standard forms in table 4, inspired by [24]. We note that, unlike the method proposed by [37], these methods are computationally expensive (e.g. computing L^+ the pseudo-inverse of L). An even bigger disadvantage is that kernel methods cannot incorporate any non-topological information for link prediction.

Another flavour of kernel based methods is the pairwise kernel approach, described in [39]. Pairwise kernels model the similarity between pairs of nodes, rather than nodes (as above). In this case, the kernel matrix is used to find similar pairs of nodes to already linked pairs of nodes (similar nodes has similar links). Typical pairwise kernels include Kronecker product kernel, Cartesian kernel ([39]), which are not precomputed kernels (as above) but need to be learnt in a semisupervised matter. Due to the high time/space complexity of these methods, additional effort has been dedicated to finding a faster solution. In [40], they use Kronecker sum similarity instead Kronecker product similarity and an efficient algorithm based on conjugate gradient problem to diminish the complexity of the algorithm with better results than standard pairwise kernel methods.

Name	Score(x, y)
Path kernel	$F_P(A) = \sum_{i=1}^{d} \alpha_i A^i$
Exponential kernel	$F_{EXP}(A) = \sum_{i=1}^{\infty} \frac{\alpha_i}{i!} A^i$
Von Neumann kernel	$F_{NEU}(A) = (I - \alpha A)^{-1} = \sum_{i=0}^{\infty} \alpha_i A^i$
Commute time kernel	$F_{COMM}(L) = L^+$
Regularized commute time kernel	$L_{COMMR}(L) = (I + \alpha L)^{-1}$
Heat diffusion kernel	$F_{HEAT}(L) = exp(-\alpha L)$

Table 4: Overview of standard kernel functions

2.2.4 Matrix Factorization

We conclude reviewing well-known link prediction methods by considering the matrix factorization technique described by Menon et al. [41]. Given a matrix G as a partial adjacency matrix (we only know some 0s and 1s), they use matrix factorization to complete all the unknown labels for the remaining pairs. G is factorized into $G \approx L(U \Lambda U^T)$ where U is the latent feature matrix for each node i (u_i is the latent feature vector with k features) and L is the link function. Explicit node or even edge attributes (e.g. number of published papers, number of published papers together as weight etc.) can also be included in the learning model as constants. The loss function derived using a bilinear regression model (accounting for both latent and explicit features) is minimized using stochastic gradient descent, thus making the model suitable for large graphs. Class imbalance is addressed by learning directly over AUC on the training set and the results show a net improvement in 6 datasets over typical unsupervised measures as well as a simple supervised classification model. Best results are obtained with square loss, identity link function and incorporating unsupervised scores as explicit features.

2.3 Application: Weighted Networks

Link prediction in weighted networks is controversial (do weights actually improve prediction?), prominent papers presenting contradicting results [28, 4, 42]. A weighted social network is based on the same underlying graph, but edges are now weighted by usually the number of interactions between the end nodes (the actors). In [28], they redefine common unsupervised similarity measures into their weighted version: Adamic/Adar weighted (AAw), Common Neighbours

(CNw) and Preferential Attachment (PAw) and they test them on Question Answering Bulletin Boards (QABB), a larger and more dynamic network (over 50000 users). Their results show that using weights improves link prediction, but their result evaluation is unreliable (they measure accuracy over newly appeared links in test network thus ignoring class imbalance problems). Conflicting results are presented in [4] where they apply the same weighted measures (additional Resource Allocation[26]) on smaller networks (a transportation network, a biological one and a co-authorship one). They infer, based on the Weak Tie Theory [43], that weak ties have a more significant role in link prediction. They attempt to prove this by parameterizing the edge weight in similarity score calculation, and showing how weak links may be even more important than strong ones. In a supervised learning context [42], edge weights appear to be slightly beneficial in supervised approaches to link prediction, but again, they derive the results using a balanced test set, thus may be biased. Sett et al. [44] perform an extensive comparison, define new edge weighting models and show that each dataset may benefit from a differently tuned model. Their analysis also shows that unweighted models are likely to outperform weighted ones when detecting inter-community links (similar to Weak Tie Theory) and vice-versa when operating within a community.

2.4 Current trends

The current research interest in link prediction reflects the exponential increase in available data and explores **graph streams** as methods to deal with both extremely large and fast evolving graphs. In [45, 16] it is argued that static methods are unable to solve these problems, since they assume the entire graph fits into memory and operate offline on the snapshot. In [45], one of the first works to consider streams for link prediction, graph streams are defined as an infinite sequence of edge update operations (new edge or deletion of an old one). As it is common for graph stream solutions [46], Zhao et al. [45] define particular graph sketches to approximate graph metrics, in this case to approximate unsupervised similarity methods: Jaccard, Adamic/Adar and Common Neighbours. They show that the approximations are not far-off from the exact values in terms of accuracy of top N elements (they also compare with Katz and PropFlow), but the runtime decreased from $>$ 2 hours to less than 1 second with small space cost. In [16], they develop the framework SLIDE, which also maintains a low rank sketching matrix containing a summary of the seen data, but they do not try to adapt any classical link prediction method. In SLIDE, they recognize, however, the predictive power of node attributes (also solving the cold-start problem) and find a way to store and update them in the sketching matrix, which is then used to infer missing links on the fly. They show comparable or even better results in terms of AUC on datasets such as Epinions, DBLP with a similar decrease in runtime and memory cost as in [45].

3 Summary & Conclusion

Link prediction is a relatively old problem (influential work started to be published over 20 years ago), but the assumptions it works on have changed considerably. Liben-Nowell et al. [1] performed in 2003 (when the first version was published) experiments on a static snapshot of the network with less than 2000 nodes. In 2016, Zhao et al. [45] used a graph stream approach to predict links in a network with more than 1.8 million nodes. In recent years, to assume a static network or even a network modelled in time frames is hardly reasonable anymore. Moreover, developing a solution that does not work well on highly dynamic online social networks is not sensible anymore. The size of the networks available for analysis is growing (e.g. Twitter has

336 million users [47]), thus the class imbalance problem worsens and space/time inefficient solutions become irrelevant. This change in assumptions reflects directly in current research trends for link prediction. Graph streams based methods for link prediction are attempting to solve the problem with one pass through the graph and in smaller time and space complexities by order of magnitudes [16].

Unlike literature reviews in other fields, we could not empirically compare the works against a state-of-the-art model or even between each other (comparisons are also noticeably absent from other recent reviews [17]). This is primarily (but not only) due to the lack of benchmark datasets and evaluation methods for link prediction, problem underlined also in [17]. Almost each research paper uses a different dataset possibly curated to show how the proposed method is better and most likely not publicly available. Some of the common datasets include DBLP (co-authorship network), ArXiv, Twitter, Slashdot, Epinions, Wikipedia, biological networks (protein protein interaction sets), while other works simply construct their new datasets [28, 23, 20]. While the problem of different evaluation methods has diminished (most papers use AUC evaluation in recent years [18]), the lack of benchmark datasets makes it impossible to compare approaches systematically to identify state-of-the-art in a given network type. While we agree that various methods are likely to work only a particular type of networks (e.g. bipartite networks), researchers could work on maintaining a set of benchmark datasets which could include: a large co-authorship dataset with node/edge info, one without additional info, a standard biological network, one standard online social network (snapshot of it) etc. Then each new proposed method should be evaluated using AUC on these benchmark datasets and practical guidelines about choosing a suitable method for a type of dataset could be easily derived.

We conclude by offering practical guidelines(though limited) that might help narrow down choices for link prediction. If the network offers any type of additional node information (non-topological data), always incorporate it into the model [22, 41]. Location features for actors are especially likely to offer a significant advantage in link prediction [20]. It is always better to choose a learning based model, rather than an unsupervised similarity model [27]. If the network is relatively small and can be treated as a snapshot, all the discussed methods are viable but pay attention to promising probabilistic approaches [33, 35]. If the network is large (over 10000 nodes), you can probably safely discard probabilistic models, as well as expensive kernel based models, but you could look into matrix factorization[41] or learning the kernel [37]. If the network is especially large, consider the SLIDE framework and treat the graph as a stream of new data points [16]. Lastly, compute the clustering coefficient of the network and determine whether the links are likely to be within a community (high coefficient) or between communities (low coefficient) and consider an edge weighting model to match these findings.

References

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

[2] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[3] Shazia Tabassum, Fabiola SF Pereira, Sofia Fernandes, and João Gama. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1256, 2018.

[4] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[5] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Friendship prediction and homophily in social media. *ACM Transactions on the Web (TWEB)*, 6(2):9, 2012.

[6] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206. ACM, 2010.

[7] Junichiro Mori, Yuya Kajikawa, Hisashi Kashima, and Ichiro Sakata. Machine learning approach for finding business partners and building reciprocal relationships. *Expert Systems with Applications*, 39(12):10402–10407, 2012.

[8] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1285–1293. ACM, 2012.

[9] Qing Guan, Haizhong An, Xiangyun Gao, Shupei Huang, and Huajiao Li. Estimating potential trade links in the international crude oil trade: A link prediction approach. *Energy*, 102:406–415, 2016.

[10] Sisay Fissaha Adafre and Maarten de Rijke. Discovering missing links in wikipedia. In *Proceedings of the 3rd international workshop on Link discovery*, pages 90–97. ACM, 2005.

[11] Jianhan Zhu, Jun Hong, and John G Hughes. Using markov models for web site link prediction. In *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, pages 169–170. ACM, 2002.

[12] Giulia Berlusconi, Francesco Calderoni, Nicola Parolini, Marco Verani, and Carlo Piccardi. Link prediction in criminal networks: A tool for criminal intelligence analysis. *PloS one*, 11(4):e0154244, 2016.

[13] Valdis E Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002.

[14] Wadhah Almansoori, Shang Gao, Tamer N Jarada, Abdallah M Elsheikh, Ayman N Murshed, Jamal Jida, Reda Alhajj, and Jon Rokne. Link prediction and classification in social networks and its application in healthcare and systems biology. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 1(1-2):27–36, 2012.

[15] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, Eric P Xing, and Tommi Jaakkola. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, volume 15, 2006.

[16] Jundong Li, Kewei Cheng, Liang Wu, and Huan Liu. Streaming link prediction on dynamic attributed networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 369–377. ACM, 2018.

[17] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

[18] Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. Evaluating link prediction methods. *Knowledge and Information Systems*, 45(3):751–782, 2015.

[19] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[20] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054. ACM, 2011.

[21] Alexandrin Popescul and Lyle H Ungar. Statistical relational learning for link prediction. In *IJCAI workshop on learning statistical models from relational data*, volume 2003. Citeseer, 2003.

[22] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[23] Elena Zheleva, Lise Getoor, Jennifer Golbeck, and Ugur Kuter. Using friendship ties and family circles for link prediction. In *Advances in Social Network Mining and Analysis*, pages 97–113. Springer, 2010.

[24] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[25] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[26] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The*

European Physical Journal B, 71(4):623–630, 2009.

[27] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.

[28] Tsuyoshi Murata and Sakiko Moriyasu. Link prediction of social networks based on weighted proximity measures. In *Proceedings of the IEEE/WIC/ACM international conference on web intelligence*, pages 85–88. IEEE Computer Society, 2007.

[29] Linyuan Lü, Ci-Hang Jin, and Tao Zhou. Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, 80(4):046122, 2009.

[30] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. Fast and accurate link prediction in social networking systems. *Journal of Systems and Software*, 85(9):2119–2132, 2012.

[31] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[32] Kurt C Foster, Stephen Q Muth, John J Potterat, and Richard B Rothenberg. A faster katz status score algorithm. *Computational & Mathematical Organization Theory*, 7(4):275–285, 2001.

[33] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98, 2008.

[34] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *icdm*, pages 322–331. IEEE, 2007.

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

[36] Francois Fouss, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of five graph kernels on a collaborative recommendation task. In *IEEE International Conference on Data Mining (ICDM)*, 2006.

[37] Jérôme Kunegis and Andreas Lommatzsch. Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM, 2009.

[38] François Fouss, Kevin Francoisse, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. *Neural networks*, 31:53–72, 2012.

[39] Hisashi Kashima, Satoshi Oyama, Yoshihiro Yamanishi, and Koji Tsuda. On pairwise kernels: An efficient alternative and generalization analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 1030–1037. Springer, 2009.

[40] Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 1100–1111. SIAM, 2009.

[41] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2011.

[42] Hially Rodrigues De Sá and Ricardo BC Prudêncio. Supervised link prediction in weighted networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2281–2288. IEEE, 2011.

[43] Peter Csermely. Weak links: Stabilizers of complex systems from proteins to social networks. *Weak Links: Stabilizers of Complex Systems from Proteins to Social Networks, by Peter Csermely. 2006 XX, 408 p. 37 illus. 3-540-31151-3. Berlin: Springer, 2006.*, page 37, 2006.

[44] Niladri Sett, Sanasam Ranbir Singh, and Sukumar Nandi. Influence of edge weight on node proximity based link prediction methods: An empirical analysis. *Neurocomputing*, 172:71–83, 2016.

[45] Peixiang Zhao, Charu Aggarwal, and Gewen He. Link prediction in graph streams. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 553–564. IEEE, 2016.

[46] Sudipto Guha and Andrew McGregor. Graph synopses, sketches, and streams: A survey. *Proceedings of the VLDB Endowment*, 5(12):2030–2031, 2012.

[47] Craig Smith. 400 interesting twitter stats and facts (2018) — by the numbers - dmr, 2018.