Highly Accurate Link Prediction in Networks Using Stacked Generalization

Amir Ghasemian
Department of Computer Science,
University of Colorado
Boulder, CO
amgh5286@colorado.edu

Aram Galstyan
Information Sciences Institute,
University of Southern California
Marina del Rey, CA
galstyan@isi.edu

Aaron Clauset
Department of Computer Science,
University of Colorado
Boulder, CO
aaron.clauset@colorado.edu

ABSTRACT

Link prediction is an important task in complex networks which has a wide variety of applications in recommendation systems. Although many community detection methods can be used in this end, the recently proved No Free Lunch theorem for community detection implies that each makes some kind of trade-off, and no algorithm can be optimal on all inputs. Different algorithms will thus over- or under-fit on different inputs, finding more, fewer, or just different communities than is optimal. Therefore, algorithms vary widely in how many communities they find in a given input, and these differences induce wide variation in accuracy on link prediction task. On the other hand, stacked generalization approach helps to design a high-level classification model via combining the lower-level models. The goal is to have better predictive performance by learning and fusing the best performance of each link prediction algorithm in different feature configurations. In this paper, we present a novel ensemble approach on link prediction using stacked generalization and show that via this approach we can achieve a higher precision and recall compared to each algorithm and to the best-overall-predictors. We present our results on 11 edge prediction algorithms on a novel and diverse corpus of 406 diverse real-world networks.

KEYWORDS

Link Prediction, Stacked Generalization, Complex Networks, Feature Extraction

ACM Reference Format:
Amir Ghasemian, Aram Galstyan, and Aaron Clauset. 2017. Highly Accurate Link Prediction in Networks Using Stacked Generalization. In *Proceedings of ACM Conference (Conference'18).* ACM, New York, NY, USA, Article X, 5 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Many information systems can be represented through the networks, where the individuals and their relations are denoted by nodes and edges, respectively. Real networks are usually incomplete and have lots of missing edges which can be resulted from variety of reasons, for example, since the existence of the edges in many biological networks like protein-protein or gene-gene interactions should be examined via costly experiments which makes our knowledge of these networks very limited [12], or due to sampling social networks, or because of constraints on number of references in citation networks. Link prediction is one of the most common tasks in networks which has an effective role in recommendation systems, protein-protein interaction prediction, novel interaction between genes, identifying the hidden activities of terrorists and criminals. Also most of the tasks on networks which takes the networks as input are sensitive to missing edges. For example community detection results can be changed via the incomplete networks [3], or many network features like the clustering coefficient are sensitive to missing edges.

Most of the traditional algorithms in link prediction are based on some scoring function [10] to rank the potential links and via computing a proper threshold, propose the top k links as the missing edges or future edges depending on the application. The best algorithms in traditional link prediction coming from probabilistic group of methods in community detection [6]. Although many community detection methods can be used in this end [6], the recently proved No Free Lunch theorem for community detection [16] implies that no method can be optimal on all inputs, and hence every method must make a tradeoff between better performance on some kinds of inputs for worse performance on others. Different algorithms will thus over- or under-fit on different inputs, finding more, fewer, or just different communities than is optimal. Therefore, algorithms vary widely in how many communities they find in a given input, and these differences induce wide variation in accuracy on link prediction task [6].

We can think the traditional approaches as unsupervised or better say semi-supervised link prediction techniques. Designing link prediction approach in a more supervised form can imrove the results due to several reasons more importantly (i) link prediction is a highly imbalanced classification task which supervised techniques can improve the results, (ii) most of the scoring functions in unsupervised approaches are looking to just partial information of the network structure such as the number of common neighbors, larger weights on common neighbors with lower degree knwon as Adamic-Adar, etc.. On the other hand supervised techniques can consider all of these deficiencies because of their inherent characteristics [1, 11]. Therefore, the state of the art algorithms related to the supervised techniques which outperform traditional approaches by extracting more information from the networks using the topological feature configurations on each network.

Ensemble techniques are supervised classification which try to improve the learning task by learning a higher-level model through combining the predictions from base-learners (lower-level ones). Ensemble techniques include bagging [2] for variance reduction,

Conference'18, Feb 2018, Marina Del Rey, CA, USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

boosting [19] to improve the performance of a classifier via decreasing the bias and variance and stacked generalization [20] to improve the prediction task and decrease the variance and bias through combining several classifiers. Most of the previous approaches in link prediction use bagging [1, 11] and a few use boosting [4]. Also recently in Ref. [5] the authors propose a bagging approach to scaling up link prediction by dividing the data to bootstrap samples and solving each part with latent factor models. In another work [9] the authors use Gradient Boosting Decision Tree for feature extraction to derive better feature sets from the initial features.

As mentined earlier the NFL theorem for community detection [16] implies that no method can be optimal on all inputs, and hence every method must make a tradeoff between better performance on some kinds of inputs for worse performance on others. Although this diversity of algorithmic performance make these algorithms vulnerable in link prediction separately, but utilizing them altogether makes them powerful in link prediction.

In this paper we are presenting a stacked generalization technique to improve the link prediction results coming from 11 unsupervised link prediction algorithms all originated from 11 state-of-the-art community detection algorithms by combining them using a classifier. We evaluate the proposed stacked generalization link prediction technique using a novel corpus of 406 real-world networks introduced as CommunityFitNet corpus in [6] from many scientific domains, which provides general insights about their performance in practical settings. The 406 real-world networks are a structurally and size diverse corpus that we believe provides a reasonable estimation of generalization error on other real-world networks. We see that the scoring functions as meta features can be used in link prediction in a supervised setting and adding topological features of networks improves the link prediction significantly. We will present our study for link prediction on the sampled observed networks by training over some other networks.

2 DIVERSITY IN THE RESULTS OF LINK PREDICTION TECHNIQUES

In this section we study the results coming from 11 unsupervised link prediction algorithms, all originated from 11 state-of-the-art community detection algorithms (see Table 1). Traditional link prediction algorithms, using a scoring function, rank the contribution that the added unobserved edge would make to their corresponding partition score functions, to distinguish the true positives (missing links) and true negatives (non-edges). Here we first explain briefly these link prediction techniques we used in our set of algorithms and then we present the results coming from these techniques on the CommunityFitNet corpus, a novel data set containing 406 real-world networks introduced in [6].

2.1 Link Prediction

In our lower-level predictors for a given graph $G = (V, E)$ we define $E' \subset E$ and $G' = (V, E')$. For each method f with a partitioning $C = f(G')$ we define a model-specific score function s_{ij} for $ij \in V \times V \setminus E'$ and evaluate its accuracy at discriminating between the true positives (missing links) $(i, j) \in E \setminus E'$ and true negatives (non-edges) $(i, j) \in V \times V \setminus E$ of G. We keep a fraction of our edges noted with α ($\alpha = 0.9$ in our experiments) i.e. $|E'| = \alpha|E|$ chosen uniformly at random. Link prediction is a kind of binary classification task, meaning its accuracy can be quantified by the area under the ROC curve (AUC). When computing ROC curves, we break ties in the scoring function uniformly at random. Each network produces one AUC value, and we average the AUCs produces for each of our real-world networks into a single value that gives a general measure of the algorithm's performance. Also since the link prediction is highly imbalanced classification problem we report F-measure besides AUC for our evaluations.

Table 1: Abbreviations and Descriptions of 11 Community Detection Methods.

Abbreviation	Ref.	Description
Q	[14]	Modularity, Newman-Girvan
Q-MR	[13]	Modularity, Newman's multiresolution
Q-MP	[21]	Modularity, message passing
B-NR (SBM)	[15]	Bayesian, Newman and Reinert
B-NR (DC-SBM)	[15]	Bayesian, Newman and Reinert
B-HKK (SBM)	[7]	Bayesian, Hayashi, Konishi and Kawamoto
cICL-HKK (SBM)	[7]	Corrected integrated classification likelihood
Infomap	[18]	Map equation
MDL (SBM)	[17]	Minimum description length
MDL (DC-SBM)	[17]	Minimum description length
S-NB	[8]	Spectral with non-backtracking matrix

2.2 Link Prediction Results

The average of accuracy, precision and recall for all these algorithms over the all networks in CommunityFitNet corpus are given in Fig. 1. The simplest approach in combining the outputs, resulted from our set of algorithms, is through the majority vote of the results. The majority vote algorithm classify a query as an edge/non-edge, if it is identified as an edge/non-edge by at least 6 algorithms out of 11. The performance of this majority vote link prediction and the results of each link prediction algorithm are presented in Table 2. Here, we have the mean performance of these algorithms over 406 real world networks. A naive upper bound for recall in link prediction is by inferring a true label if any of the algorithms classify the query pair of nodes as an edge. The result for this naive method is also included in Table 2. The results show that the majority vote improves the results. The recall is large which means by using the majority vote we can identify the true labels of minority class. However, since the precision of this approach is very small it shows we are assigning much more true labels than the reality but still the results are better than any other single algorithm. On the other hand, the recall for our naive approach is too high which says every single minority class can be identified by at least one of our algorithms, but the small precision shows that this naive approach is useless and it identifies many non-edges as links.

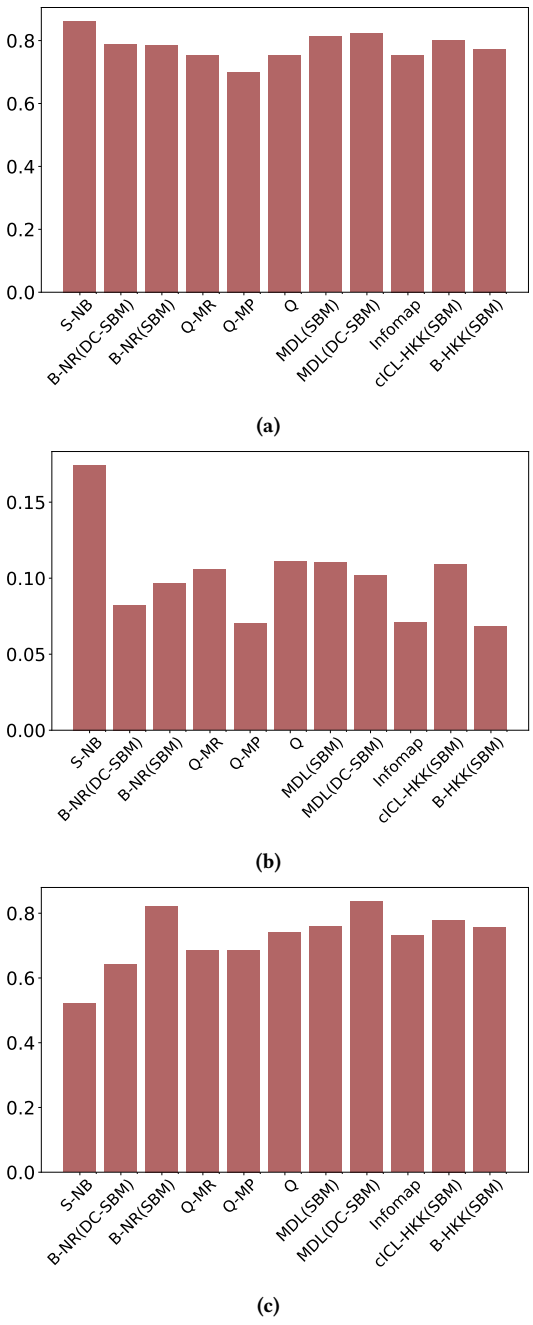

Figure 1: Average of (a) Accuracy (b) Precision (c) Recall for 11 link prediction algorithms in Table 2 over 406 networks in CommunityFitNet corpus.

3 STACKED GENERALIZATION FOR LINK PREDICTION

Due to the diversity of these link prediction techniques we observed that combining these approaches using majority vote improves the results. Using more advanced approaches can improve the results even more. Here, we design a highly accurate link prediction on networks using stacked generalization approach [20]. For our higher-level meta classifier we use a random forest classifier with 50 trees and a maximum depth of 7. We have around 15 million pair of candidates named as the set of unobserved edges in G'. This set contains both the true removed edges and the non-edges. Here for our higher-level classifier we train it using one hundred thousands of unobserved edges chosen randomly from a subset of networks (twenty percent of networks) and train our meta-learner by balancing our training set. The evaluation results on unseen networks (eighty percent of networks) are presented for a test set of size one million in Fig. 2. The results show some interesting trends as (i) the importance of scoring features and their combination (M2 in Fig. 2-b) and specifically its importance in learning the minority class (true edges), (ii) the topological features will improve M2 classifier to better learn the algorithms, and (iii) adding more features in Fig. 2-c shows very promising results.

Table 2: Algorithms mean performance over 406 networks in CommunityFitNet corpus.

Model	AUC	Precision	Recall	F-measure
Q	0.73	0.12	0.74	0.18
Q-MR	0.72	0.11	0.69	0.17
Q-MP	0.67	0.07	0.69	0.12
B-NR (SBM)	0.84	0.10	0.82	0.16
B-NR (DC-SBM)	0.71	0.08	0.64	0.14
cICL-HKK (SBM)	0.82	0.10	0.78	0.16
B-HKK (SBM)	0.79	0.06	0.76	0.10
Infomap	0.76	0.07	0.73	0.12
MDL (SBM)	0.82	0.10	0.76	0.17
MDL (DC-SBM)	0.87	0.1	0.84	0.17
S-NB	0.70	0.16	0.52	0.17
naive upper bound for Recall	-	0.02	1.0	0.05
Majority	0.92	0.18	0.84	0.26

Table 3: Abbreviations and Descriptions of 3 different settings

Classifier	Description
M1	random forest classifier using just the topological features
M2	random forest classifier using just the scores as features
M3	random forest classifier using both the topological features and scores

ACKNOWLEDGMENTS

The authors thank David Wolpert and Brendan Tracey for helpful conversations, and acknowledge the BioFrontiers Computing Core at the University of Colorado Boulder for providing High Performance Computing resources (NIH 1S10OD012300) supported by BioFrontiers IT. Financial support for this research was provided

in part by Grant No. IIS-1452718 (AmirG, AC) from the National Science Foundation.

REFERENCES

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

[2] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[3] Matthew Burgess, Eytan Adar, and Michael Cafarella. 2016. Link-prediction enhanced consensus clustering for complex networks. *PloS one* 11, 5 (2016), e0153384.

[4] Prakash Mandayam Comar, Pang-Ning Tan, and Anil K Jain. 2011. LinkBoost: A novel cost-sensitive boosting framework for community-level network link prediction. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 131–140.

[5] Liang Duan, Shuai Ma, Charu Aggarwal, Tiejun Ma, and Jinpeng Huai. 2017. An Ensemble Approach to Link Prediction. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2402–2416.

[6] Amir Ghasemian, Homa Hosseinmardi, and Aaron Clauset. 2017. Evaluating and Comparing Overfit in Models of Network Community Structure. (2017). submitted.

[7] Kohei Hayashi, Takuya Konishi, and Tatsuro Kawamoto. 2016. A Tractable Fully Bayesian Method for the Stochastic Block Model. *arXiv:1602.02256* (2016).

[8] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. 2013. Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci.* 110, 52 (2013), 20935–20940.

[9] Taisong Li, Jing Wang, Manshu Tu, Yan Zhang, and Yonghong Yan. 2016. Enhancing Link Prediction Using Gradient Boosting Features. In *International Conference on Intelligent Computing*. Springer, 81–92.

[10] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology* 58, 7 (2007), 1019–1031.

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

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

[13] MEJ Newman. 2016. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *arXiv:1606.02319* (2016).

[14] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69, 2 (2004), 026113.

[15] Mark EJ Newman and Gesine Reinert. 2016. Estimating the number of communities in a network. *Phys. Rev. Lett.* 117, 7 (2016), 078301.

[16] Leto Peel, Daniel B Larremore, and Aaron Clauset. 2017. The ground truth about metadata and community detection in networks. *Sci. Adv.* 3, 5 (2017), e1602548.

[17] Tiago P Peixoto. 2013. Parsimonious module inference in large networks. *Phys. Rev. Lett.* 110, 14 (2013), 148701.

[18] Martin Rosvall and Carl T Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA* 105, 4 (2008), 1118–1123.

[19] Robert E Schapire. 1999. A brief introduction to boosting. In *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*. Morgan Kaufmann Publishers Inc., 1401–1406.

[20] David H Wolpert. 1992. Stacked generalization. *Neural networks* 5, 2 (1992), 241–259.

[21] Pan Zhang and Cristopher Moore. 2014. Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. *Proc. Natl. Acad. Sci. USA* 111, 51 (2014), 18144–18149.

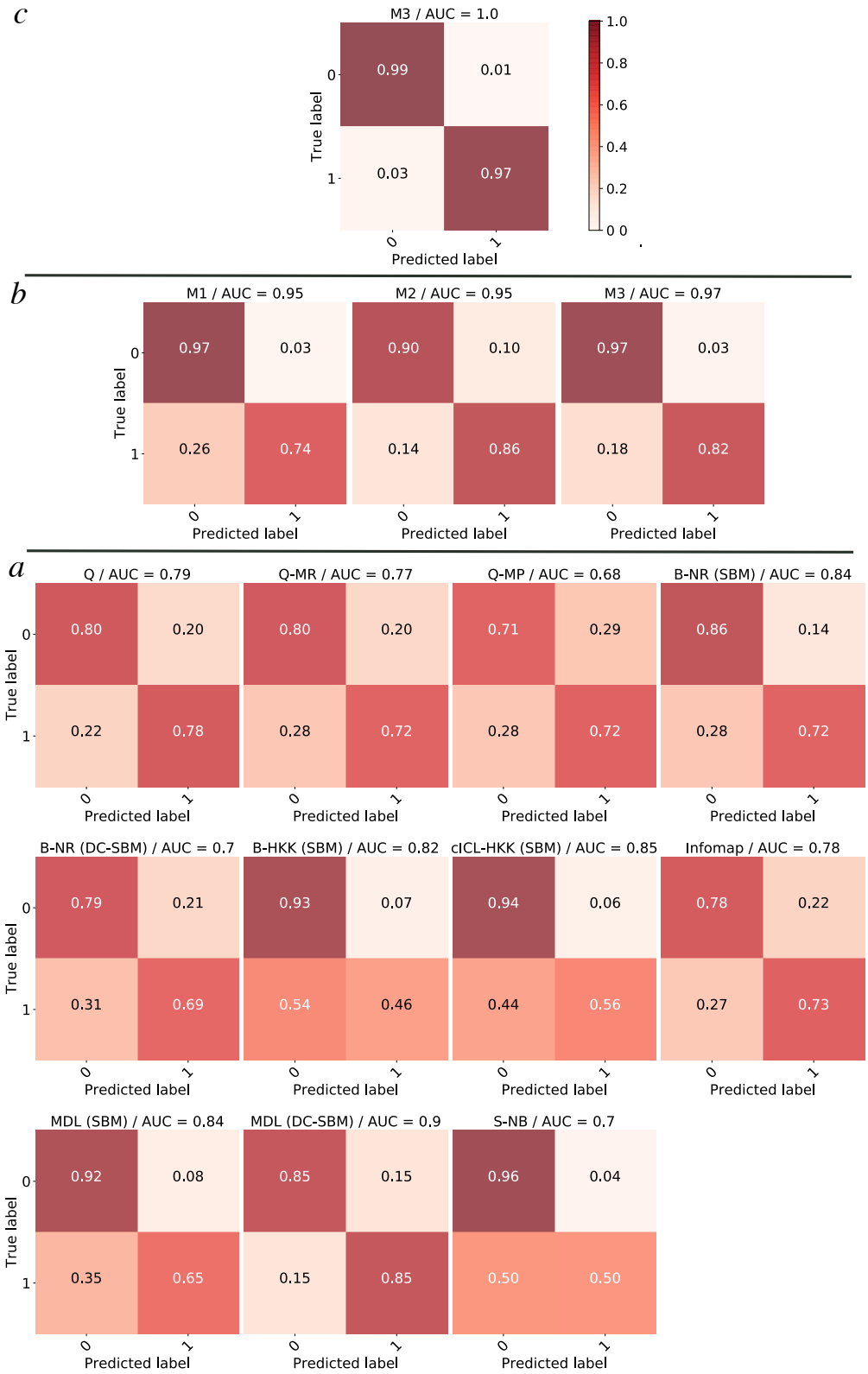

Figure 2: (a) results of lower-level edge predictors, (b) random forest classifier using (1) just the topological features (M1), (2) just the scoring features (M2) (importance of scoring features), and (3) both topological and scoring features (M3) (adding the topological features will improve the results and help in learning the optimal combination of unsupervised link prediction algorithms), (c) adding more features improve the results.