The probability of edge existence due to node degree: a baseline for network-based predictions

This manuscript (permalink) was automatically generated from greenelab/xswap-manuscript@797c29d on August 19, 2019.

Authors

- **Michael Zietz**
  0000-0003-0539-630X · zietzm · ZietzMichael
  Department of Physics & Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America; Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America · Funded by ['Roy and Diana Vagelos Scholars Program in the Molecular Life Sciences', 'the Gordon and Betty Moore Foundation (GBMF4552)']

- **Daniel S. Himmelstein**
  0000-0002-3012-7446 · dhimmel · dhimmel
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America · Funded by Pfizer Worldwide Research, Development, and Medical; the Gordon and Betty Moore Foundation (GBMF4552)

- **Christopher Williams**
  · chrsunwil
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America

- **Michael W. Nagle**
  0000-0002-4677-7582 · naglem · MikeNagle84
  Internal Medicine Research Unit, Pfizer Worldwide Research, Development, and Medical

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene · greenescientist
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America · Funded by ['Pfizer Worldwide Research, Development, and Medical', 'the Gordon and Betty Moore Foundation (GBMF4552)', 'the National Institutes of Health (R01 HG010067)']

Abstract

Networks of biomedical data rarely consist of all true relationships. Instead, networks contain spurious relationships while omitting actual relationships. How a network deviates from the real set of relationships is often biased according to node degree, resulting from processes such as inspection bias and experimental methods. While degree is subject to potentially substantial biases, link prediction methods can be strongly affected by degree. In the present work, we introduce a network permutation framework to quantify the effect of node degree on network-based methods and prediction tasks. We introduce the "edge prior" to quantify the probability that two nodes are connected based only on their degree. After demonstrating that this prior feature shows excellent discrimination and calibration performance for 20 different biomedical networks (16 bipartite, 3 undirected, 1 directed), we conclude that our prior feature represents a suitable baseline for network link prediction tasks, as performance exceeding the baseline is attributable to factors other than degree alone. Additionally, we propose methods to incorporate network permutation and the edge prior into other predictive methods. Our results highlight the importance of degree for link prediction and provide a way to account for its effects when degree bias may be present. We have released a full implementation of our network permutation method and the edge prior as an open-source Python package on GitHub.

Introduction

Networks contain information about relationships between entities ("edges between nodes"). A node's degree is the number of relationships it has in the network. Networks contain many nodes, whose degrees can be aggregated to form the network's degree distribution. Because different nodes can have very different degrees, real networks have highly variable degree distributions (Figure 1).

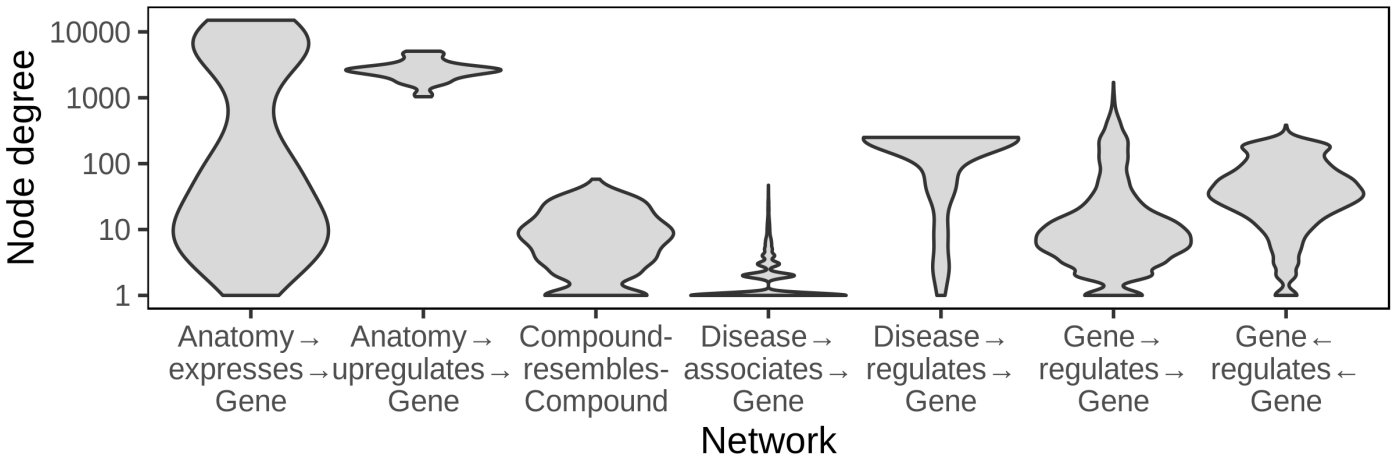


Figure 1: Degree distribution can vary greatly between networks, even between the source and target degree distributions within the same (directed or bipartite) network. Shown are seven degree distributions for six edge type subnetworks of Hetionet [1]. (The final two are the source and target distributions of the gene-regulates-gene directed edge type.)

Degree is an important metric for differentiating between nodes, and it appears in many common edge prediction features [2]. However, reliance on degree can pose problems for edge prediction. Firstly, bias in the data can distort node degree so that degree differences between two nodes may not be meaningful. Secondly, reliance on degree can lead edge prediction methods to make nonspecific or trivial predictions and fail to identify novel or insightful relationships.

Most biomedical data networks are imperfect representations of the true set of relationships. Real networks often mistakenly include edges that do not exist and exclude edges that do exist. How well a network represents the true relationships it attempts to represent depends on a number of factors,

especially the methods used to generate the data in the network [3,4,5]. We define “degree bias” as the type of misrepresentation that occurs when the fraction of incorrectly existent/nonexistent relationships depends on the number of connections that nodes make (their “degrees”). Depending on the type of data being represented, degree biases can arise due to experimental methods, inspection bias, or other factors [3].

Inspection bias indicates that entities are not uniformly studied [6], and it is likely to cause degree bias when networks are constructed using hypothesis-driven findings extracted from the literature, as newly-discovered relationships are not randomly sampled from the set of all true relationships. For example, the number of publications mentioning a gene has little correlation with its degree in a systematically-derived protein interaction network [7]. Therefore, the high correlation between number of publications and degree in low-throughput interaction networks is almost entirely the result of inspection bias. This evidence suggests that many poorly studied genes have similar numbers of interactions as the genes scientists have preferentially examined. The consequence of degree bias is that a difference in degree between two nodes may not reflect a difference in the number of true edges. Reliance on degree may be unfavorable depending on the prediction/analysis task being conducted and the magnitude of possible degree bias in the data.

Another reason why a method’s reliance on degree can be unfavorable is that degree imbalance can lead to prediction nonspecificity. Nonspecific predictions are made on the basis of generic characteristics rather than the specific connectivity information contained in a network. For example, Gillis et al. [8] examined the concept of prediction specificity in the context of gene function prediction and found that many predictions appear to rely primarily on multifunctionality and could be “potentially misleading with respect to causality.” Real networks have a variety of degree distributions (Figure 1), and they commonly exhibit degree imbalance [10,11,9]. Degree imbalance leads high-degree nodes to dominate in the predictions made by degree-associated methods [12], which are effective predictors of connections in some biological networks [13].

Consequently, degree-based predictions are more likely nonspecific, meaning the same set of predictions performs well for different tasks. However, depending on the prediction task, edge predictions between very high degree nodes may be undesired, uninsightful, or nonspecific. Model evaluation is challenging in this context, as nonspecific or trivial predictions need not be incorrect, even if they are not the desired outputs of the predictive model. For example, predicting that the highest degree node in a network shares edges with the remaining nodes to which it is not connected will often lead to many correct predictions, despite the predictions being generic to all nodes in the network. Degree-based features should often be included in the interpretation of predictions to disentangle desired from non-desired effects and to effectively evaluate and compare predictive models.

Degree is important in edge prediction, but it can cause undesired effects. We sought to understand the effect of node degree on edge prediction methods. We introduce a permutation-based framework to find edge existence probabilities due to node degree and to quantify the contribution of degree to edge prediction methods. This method allows edge predictions to be evaluated in the context of degree and its effects on the prediction task. Our results demonstrate that degree-associated methods are very effective for reconstructing a network using a subsampled holdout but ineffective for predicting edges between distinct degree distributions. Using a number of different networks, we provide evidence that degree has a strong effect on the probability of edge existence and that our “edge prior” feature best quantifies this probability.

Methods

Network permutation

Network permutation is a way to produce new networks by randomizing the connections of an existing network. Specialized permutation strategies can be devised that randomize some aspects of networks while retaining other features. Comparing between permuted and unpermuted networks gives insight to the effects of the retained network features. For example, an edge prediction method that has superior reconstruction performance on a network compared to its permutations likely relies on information that is eliminated by permutation. Conversely, identical predictive performance on true and permuted networks indicates that a method relies on information that is preserved during permutation. Network permutation is a flexible framework for analyzing other methods, because it generates networks with identical formats to the original network. We propose using network permutation to isolate degree and determine its effects in different contexts. Degree-preserving network permutation obscures true connections and higher-order connectivity information while retaining node degree, and thereby, the network's degree sequence. Thanks to the flexibility of permutation, our framework can quantify the effect of degree on any network edge prediction method.

XSwap algorithm

Hanhijärvi, et al. presented XSwap [14], an algorithm for the randomization ("permutation") of unweighted networks (Figure 2A). The algorithm picks two existing edges at random, and if the edges constitute a valid swap, exchanges the targets between the edges (Figure 1). This process is repeated many times until the maximum number of steps has been reached. In general, the maximum number of steps should be chosen to be sufficiently large that the fraction of original edges retained in the permuted network is near its asymptotic value for a large number of steps.

To allow greater flexibility, we modified the algorithm by adding two parameters, "`allow_loops`", and "`allow_antiparallel`" that allow a greater variety of network types to be permuted (Figure 2B). Specifically, two chosen edges constitute a valid swap if they preserve degree for all four involved nodes and do not violate the above condition options. The motivation for these generalizations is to make the permutation method applicable both to directed and undirected graphs, as well as to networks with different types of nodes, variously called multipartite, heterogeneous, or multimodal networks.

When permuting bipartite networks, our method ensures that each nodes class membership and with-class degree is preserved. Similarly, heterogeneous networks should be permuted by considering each edge type as a separate network. This way, each node retains its within-edge-type degree for all edge types. We provide documentation for parameter choices depending on the type of network being permuted in the GitHub repository (https://github.com/hetio/xswap). The original algorithm and our proposed modification are given in Figure 2.

A

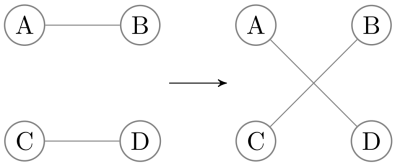
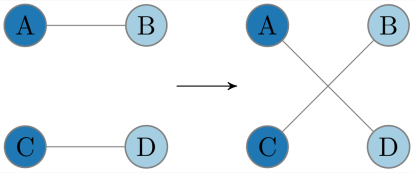
Input: Undirected graph G, distribution ρ, and number of steps T
Output: Edge-swapped graph G_s
  for i = 1,...,T **do**
    Select two edges (i,j), $(k,l) \in E(G_s)$
    if $(i,l) \notin E(G_s)$ and $(k,j) \notin E(G_s)$ **then**
      $E(\widehat{G_s}) \leftarrow \left(E(G_s) \setminus \{(i,j),(k,l)\}\right) \cup \{(i,l),(k,j)\}$
      $G_s \leftarrow \widehat{G_s}$ with probability $\min(\rho(\widehat{G_s})/\rho(G_s), 1)$
    end if
  end for

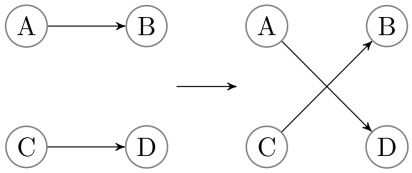
B

Input: Directed, undirected, or bipartite graph G, number of steps T, and booleans `allow_antiparallel` and `allow_loops`
Output: Edge-swapped graph G_T
  Initialize: $G_0 \leftarrow G$
  for i = 1,...,T **do**
    Select two edges (i,j), $(k,l) \in E(G_{i-1})$
    condition_1 $\leftarrow (i,l) \in E(G_{i-1})$ **or** $(k,j) \in E(G_{i-1})$
    condition_2 $\leftarrow$!`allow_antiparallel` **and** $((l,i) \in E(G_{i-1})$ **or** $(j,k) \in E(G_{i-1}))$
    condition_3 $\leftarrow$!`allow_loops` **and** $(i \neq l$ **or** $k \neq j)$
    if *condition_1* **or** *condition_2* **or** *condition_3* **then**
      continue $G_i \leftarrow G_{i-1}$
    else$E(G_i) \leftarrow (E(G_{i-1}) \setminus \{(i,j),(k,l)\}) \cup \{(i,l),(k,j)\}$
    end if
  end for

Figure 2: **A.** XSwap algorithm due to Hanhijärvi, et al. [14]. **B.** Proposed modification to XSwap algorithm

Table 1: Applications of the modified XSwap algorithm to various network types with appropriate parameter choices. For simple networks, each node's degree is preserved. For bipartite networks, each node's number of connections to the other part is preserved, and overall node class memberships are preserved. For directed networks, each nodes' in- and out-degrees are preserved, though parameter choices depend on the network being permuted. Some directed networks can include antiparallel edges or loops while others do not.

Network type	Degree preserved	Figure	`allow_antiparallel`	`allow_loops`
simple	all		False	False
bipartite	in/out		True	True

Network type	Degree preserved	Figure	`allow_antiparallel`	`allow_loops`
directed	in/out		?	?

Edge prior

We introduce the "edge prior" to quantify the probability that two nodes are connected based only on their degree. The edge prior can be estimated using the fraction of permuted networks in which a given edge exists—the maximum likelihood estimate for the binomial distribution success probability. Based only on permuted networks, the edge prior does not contain any information about the true edges in the (unpermuted) network. The edge prior is a numerical feature that can be computed for every pair of nodes that could potentially share an edge, and we compared its ability to predict edges in three tasks, discussed below.

Edge prior analytical approximation

We also considered the possibility that the probability of an edge existing across permuted networks could be written as a closed form equation involving the node pair's degree. A major simplification is the assumption that the probability of an edge existing is independent of all other potential edges. We were unable to find a closed-form solution giving the edge prior without assuming independence in this way, which we believe is incorrect for XSwap. Nonetheless, we discovered a good analytical approximation to the edge prior for networks with many nodes and relatively low edge density.

Let m be the total number of edges in the network, and $d(u_i)$, $d(v_j)$ be the source and target degrees of a node pair, respectively. A good approximation of the edge prior is given by the following:

$$P_{i,j} = \frac{d(u_i)d(v_j)}{\sqrt{(d(u_i)d(v_j))^2 + (m - d(u_i) - d(v_j) + 1)^2}}$$

Further discussion of this approximate edge prior and an derivation are available in the supplement.

Prediction tasks

We performed three prediction tasks to assess the performance of the edge prior. We compared the permutation-based prior with two additional features: a scaled product of source and target degree (scaled to the range [0, 1]) and our approximation of the edge prior. We used 20 biomedical networks from the Hetionet heterogeneous network [1] that had at least 2000 edges for the first two tasks. In the first task, we computed the degree-based prediction features (edge prior, scaled degree product, and prior approximation), and predicted the original edges in the network. We used node pairs that lacked edge in the original network as negative examples and those with an edge as positive examples. To assess the methods' predictive performances, we computed the area under the receiver operating characteristic curve for all three features. In the second task, we sampled 70% of edges from each of the networks, computed features on the sampled network, then attempted to predict held-out edges. For this task, negative examples were node pairs in which an edge did not exist in either original or sampled network, while positive samples were those node pairs without an edge in the sampled network but with an edge in the original network.

The third task evaluated the ability of the edge prior to generalize to new degree distributions. We used two domains where networks were available which shared nodes but had different degree

distributions. Protein-protein interactions (PPI) and transcription factor-target gene (TF-TG) relationships had networks created by literature curation of low-throughput, hypothesis-driven research and by high-throughput, systematic, hypothesis-free experimentation. For the PPI networks, we used the STRING network, which incorporates literature-mining to find relationships [15] and a combination of the high-throughput, proteome-scale interaction networks from Rual et al. [6] and Rolland et al. [7]. We used a transcription factor-target gene (TF-TG) literature-derived network from Han et al. [16] and a high-throughput network from Lachmann et al. [17]. The pairs of networks for PPI and TF-TG data sources are ideal because in one we expect inspection bias and in the other we do not.

As a further basis of comparison, we added a time-resolved co-authorship network, which we partitioned by time to create two separate networks. We created the co-authorship network of bioRxiv preprints using the Rxivist [18,19] database, which was generated by crawling the bioRxiv server. Unlike the other two, the co-authorship network does not have degree bias, as the network faithfully represents all true co-author relationships. We include this network to offer a comparative prediction task in which the degree distributions between training (posted before 2018) and testing (posted during or after 2018) do not differ (Figure 3A). The goal of the third prediction task is to determine feature generalizability for network reconstruction between different degree distributions, especially predicting a network without degree bias using features from a degree-biased network. Further information about the networks used can be found in the supplement.

Degree-grouping

Our method for degree-preserving permutation produces randomized networks that share few of their edges with the original network. The feature values for two node pairs with the same source and target degree are drawn from the same distribution in permuted networks, so nodes with equal degree can be grouped when summarizing features. We used this to augment each node pair's feature values in permuted networks, which allowed these pairs to have more permuted feature values than permuted networks. Degree grouping greatly increased the effective number of permutations for nodes with frequently observed degrees [20]. We used degree grouping throughout our analyses.

Implementation and source code

We implemented the modified XSwap algorithm as a Python package, with the actual edge swap mechanism implemented in C++ for greater speed. In addition to functions that permute networks (represented as edge lists), the package contains utilities for computing the edge prior, converting a network between adjacency matrix and edge list formats, and for assigning unique identifiers to nodes. The Python package is available on the Python Packaging Index under the name "xswap". The full source code for our method of degree-preserving network permutation has also been made freely available (https://github.com/hetio/xswap), as has the code for the analysis, figure generation (https://github.com/greenelab/xswap-analysis), and manuscript (https://github.com/greenelab/xswap-manuscript).

Results

Node degree bias

We found examples of node degree bias in the PPI and TF-TG networks we investigated. Figure 3 shows node degree in separate networks for the same type of data. For the PPI networks, the literature-derived network has a larger mean degree and a longer tail than the systematic network, while in the TF-TG networks this relationship is reversed. Because the TF-TG network contained far

more transcription factors than target genes (144 and 1406, respectively), the distributions of target degrees were far more compact than those of source degrees. Unlike the PPI and TF-TG networks, the co-authorship networks, which were split by date of first co-authorship, did not exhibit a great difference in their degree distributions. All three types of networks (PPI, TF-TG, and co-authorship) exhibit degree imbalance to varying extents. These results indicate that, depending on the methods by which the represented data were generated, networks of the same type of data may have overall degree distributions that differ greatly (Figure 3A), and they may even assign very different degree to the same nodes (Figure 3B).

A prediction task that uses only one type of network is challenging because degree distributions vary greatly within a single domain. Predicting systematic edges using a literature-curated network is particularly challenging, because the degree distribution of systematic edges may be skewed toward higher or lower degree relative to the literature network.

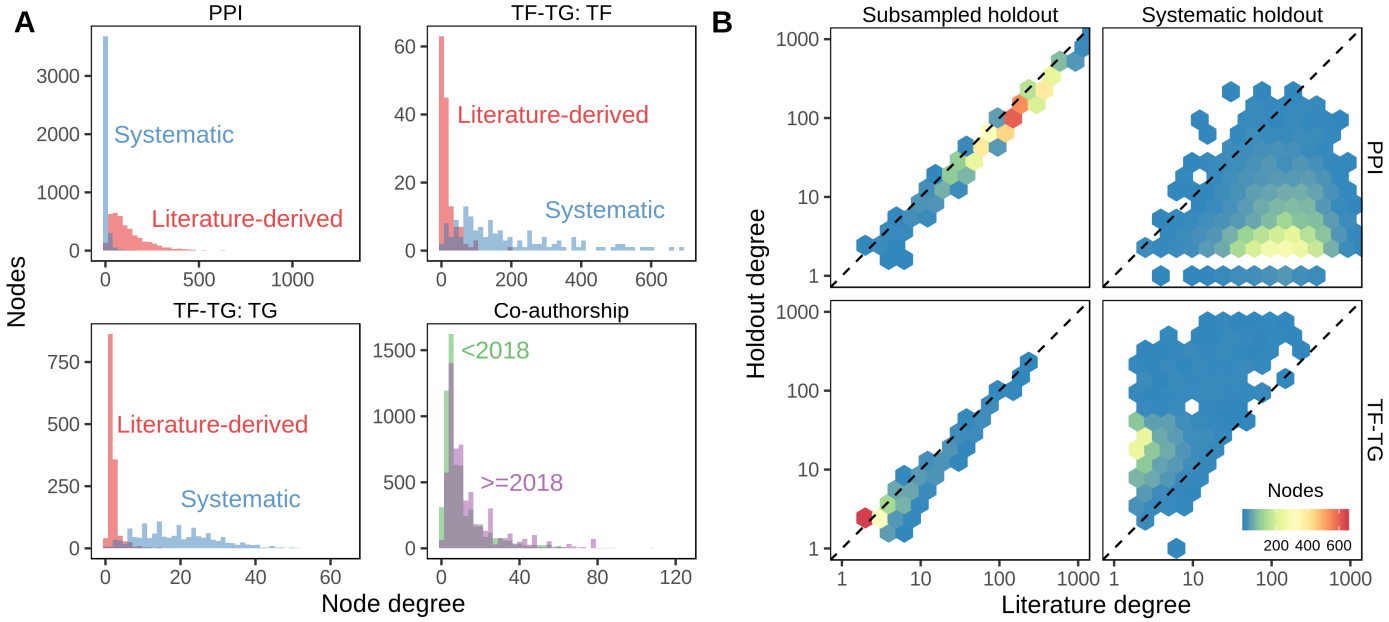


Figure 3: **A.** Degree distributions of networks with and without degree bias can be very different. Data on PPI and TF-TG were split between literature-derived and systematically-derived networks. In both cases, the networks exhibit large differences in degree distribution. Co-authorship relationship networks split by date of first co-authorship roughly share their degree distributions. **B.** Systematically-derived networks are not uniformly sampled sampled from literature-derived networks or vice versa. Uniform random sampling produces linearly-correlated node degree, while non-random sampling produces non-correlated degree. 70% of literature edges were sampled with uniform probability for the "Subsampled holdout" network.

Edge prior

In the first prediction task, we computed three features—the XSwap edge prior, an approximation to the edge prior, and the (scaled) product of source and target node degree—on networks from Hetionet. We then evaluated the extent to which these features could reconstruct the 20 networks. The XSwap-derived edge prior reconstructed many of the networks with a high level of performance, as measured by the AUROC. Of the 20 individual networks we extracted from Hetionet, 17 had an edge prior self-reconstruction AUROC >= 0.95, with the highest reconstruction AUROC at 0.9971 (Compound–downregulates–Gene edge type). Meanwhile, the lowest self-reconstruction performance (AUROC = 0.7697, Disease–localizes–Anatomy edge type) occurred in the network having the fewest node pairs.

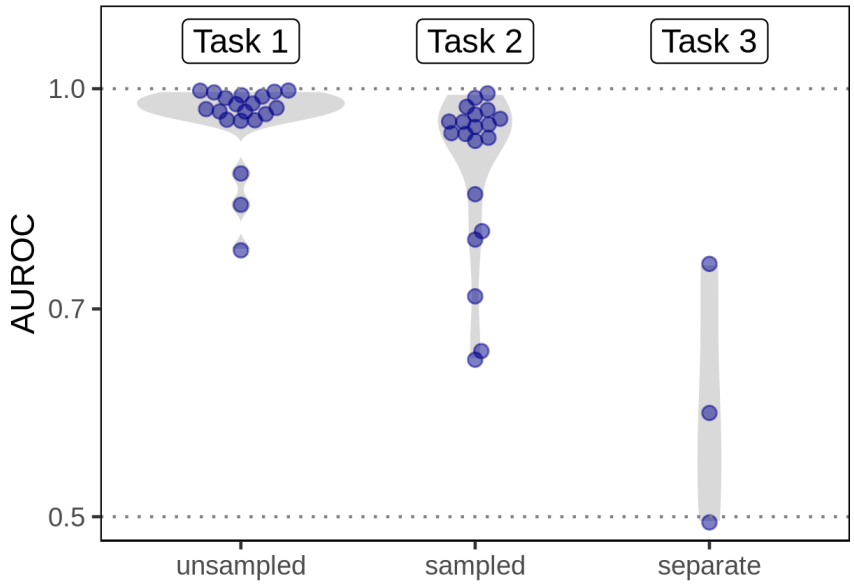


Figure 4: AUROC of network reconstruction by prediction task. The edge prior shows strong performance for network reconstruction when computed on the original (Task 1) and sampled (Task 2) networks. The performance reduction from computing features on sampled networks is real but far smaller compared to a new degree distribution.

The three features that we compared were highly rank correlated (median > 0.99999 across metaedges). The three features also had very similar AUROC reconstruction performance values for the first, second, and third prediction tasks (max difference < 0.027) because AUROC is rank-based. The edge prior was slightly better than the approximations in 12 of 20 networks. However, while the AUROC results were similar, the features were very different in their levels of calibration—the ability of the model to correctly estimate class membership probabilities. We found that the edge prior was very well calibrated for all networks in the first and second tasks, and it provided the best calibration of the three features for each of the prediction tasks (Figure 5A). As the edge prior was not based on the networks' true edges, these results indicated that degree sequence alone was highly informative and that permutation was the only approach that provided a well-calibrated model.

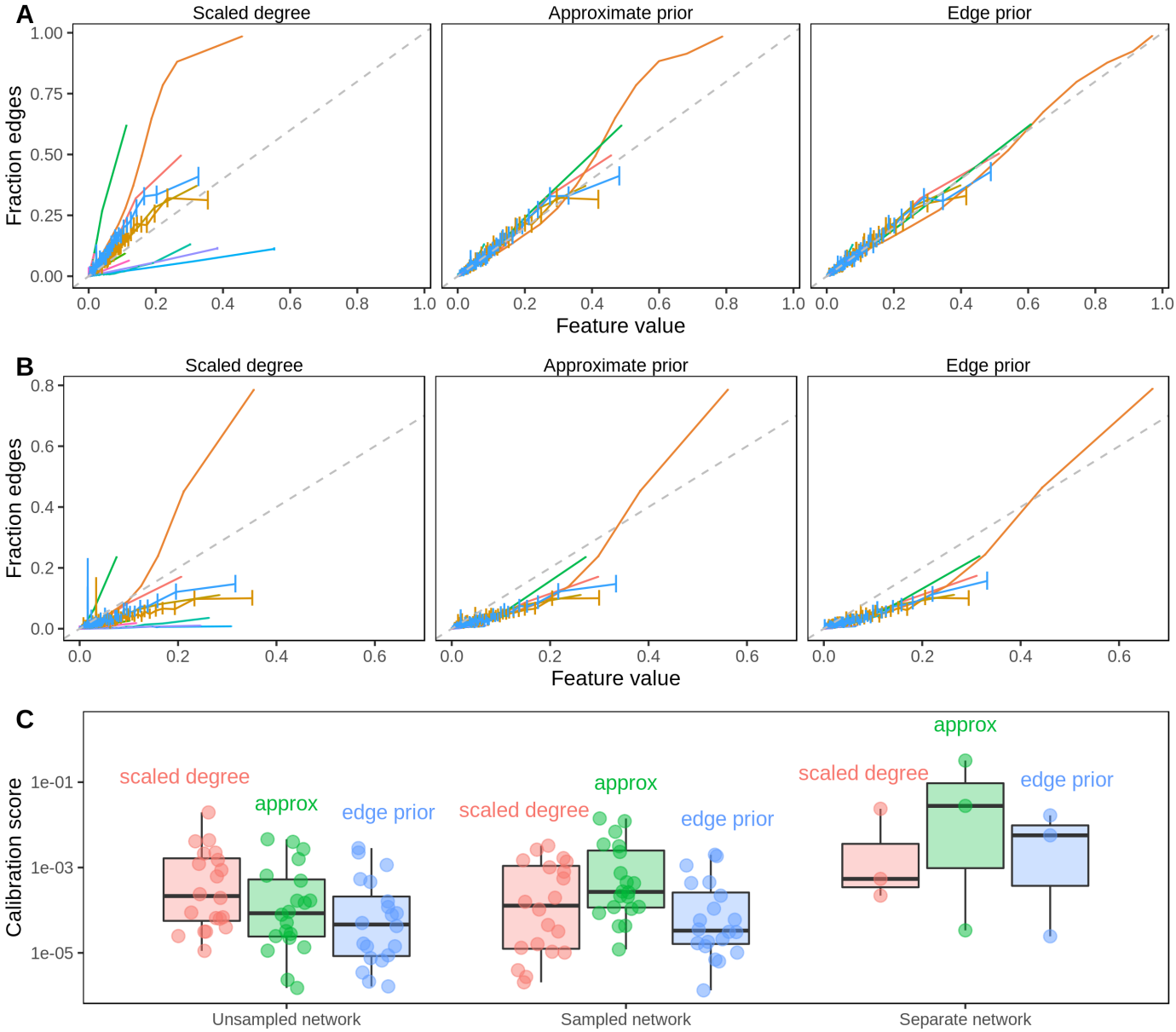


Figure 5: **A.** Calibration curves for full network reconstruction of 20 networks from Hetionet. The permutation-based edge prior's calibration was superior to the other two strategies based on degree. **B.** Calibration curves for sampled network reconstruction. The edge prior shows superior calibration in the 20 Hetionet networks. **C.** Individual Hetionet edge type calibration estimated by the two-component decomposition of the Brier score, in which lower scores indicate better calibration. The edge prior has excellent calibration in unsampled and sampled networks, and each considered method is sensitive to shifts in the degree distribution.

The second prediction task mirrored the first task, but it involved reconstructing networks based on subsampled networks with only 70% of the original edges. Because edges were sampled uniformly without replacement, the subsampled networks share similar degree distributions to the original networks (see Figure 3B). Unlike in the first task, edges that were present in the sampled network were not tested and therefore are not included in the performance metrics. The results of the second prediction task further demonstrate a high level of performance for degree-sequence-based node pair features (Figure 4). The edge prior was able to reconstruct the unsampled network with an AUROC of greater than 0.9 in 14 of 20 networks. As was observed in the first task, node pair features computed in second prediction task were highly rank-correlated, meaning the AUROC values for different features were similar. While performance was slightly lower in the second task than the first, we found that many networks were still well reconstructed. The edge prior was the best calibrated feature for both tasks.

In the third prediction task, we computed the three edge prediction features for paired networks representing data from PPI, TF-TG, and bioRxiv bioinformatics pre-print co-authorship. The goal of the task was to compare predictive performance across different degree distributions for the same type of data. We find that the task of predicting systematically-derived edges using a network with degree bias is significantly more challenging than network reconstruction, and we find consistently lower performance compared to the other tasks (Figure 4). The edge prior was not able to predict the separate PPI network better than by random guessing (AUROC of roughly 0.5). Only slightly better was its performance in predicting the separate TF-TG network, at an AUROC of 0.59. We find superior performance in predicting the co-authorship relationships (AUROC 0.75), which was expected as the network being predicted shared roughly the same degree distribution as the network on which the edge prior was computed. The results of the third prediction task show that a difference in degree distribution between the network on which features are computed and the network to be predicted can make prediction significantly more challenging.

Since the edge prior is based only on degree, it is unsurprising that it exhibits weak performance in predicting a network with a different degree distribution. We have considered the edge prior as a baseline edge predictor, whose performance indicates the utility of degree for a specific prediction task. The edge prior's low performance in the third task indicates that degree is less helpful for edge prediction tasks in which training and testing networks do not share their degree distributions. Moreover, we believe such between-distribution prediction may be a relatively common task, with examples given by the networks in Figure 3.

Assessing feature performance

We conducted a further edge prediction task as an example application of the edge prior and the permutation framework. To begin, we chose the STRING PPI network for the comparison and computed five edge prediction features (Supplemental table 2). The goal of the task was to reconstruct the network on which the features were computed. All five features were correlated with degree (Figure 6), which we quantified for a node pair using the product of source and target degrees. We expected features based on degree to show strong performance for a network reconstruction task without holdout, as found in the first prediction task.

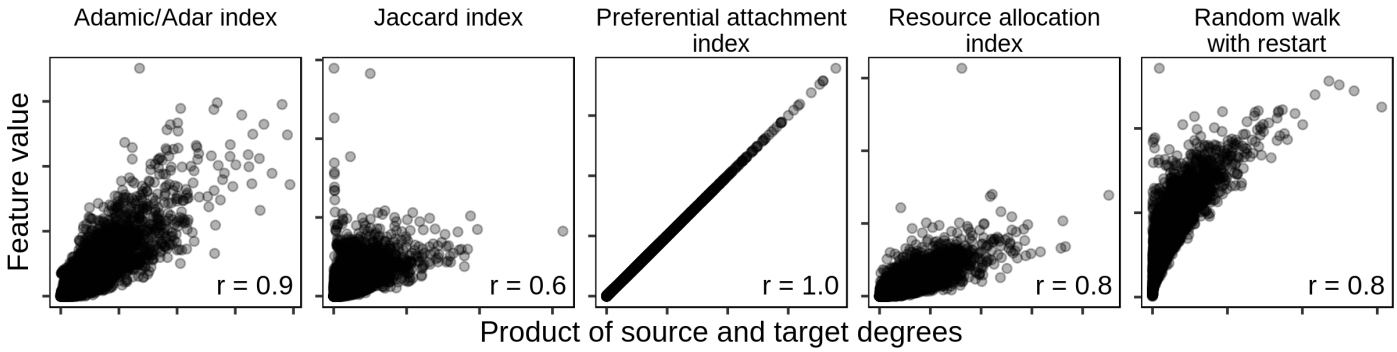


Figure 6: Five common edge-prediction features (Supplemental table 2) are correlated with node degree on the STRING PPI network [15]. All five features show a positive relationship with degree, though the magnitude of this correlation is highly variable. The preferential attachment index is understandably perfectly correlated because it is equal to the product of source and target degree.

We used two permutation-derived null values to evaluate reconstruction and contextualize performance. First, the performance of the edge prior was compared to determine the performance attributable to the degree sequence of the PPI network. The first comparison gave insight into the ability of the PPI network to be reconstructed by degree. Second, the five edge prediction features were computed on 100 permuted networks and used to reconstruct the unpermuted network. Each permuted network corresponded to AUROC values quantifying the performances of features

computed on it. The second comparison gave insight into the performance of each feature if the feature was only picking up on degree.

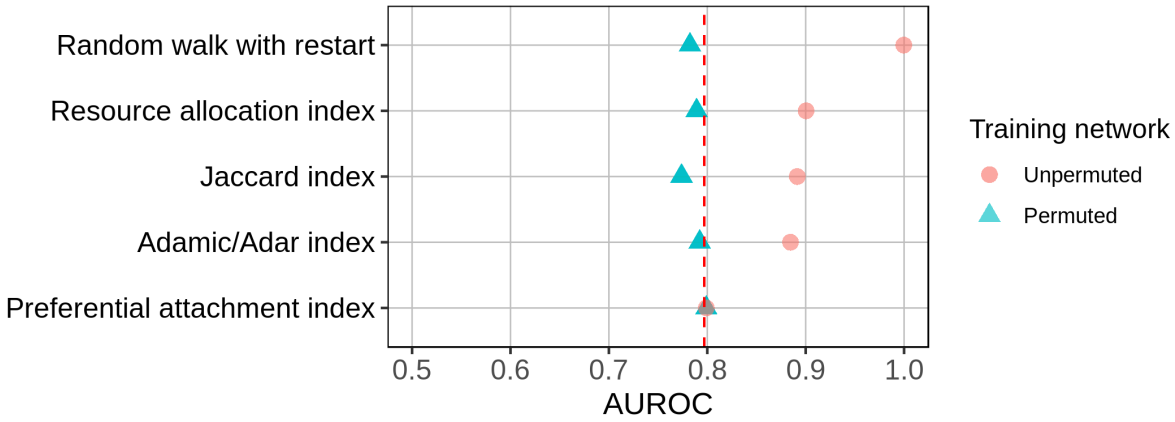


Figure 7: Network reconstruction performances by five edge prediction features. Dotted red line indicates performance of the edge prior. Each feature was computed on the unpermuted and 100 permutations of the STRING PPI network.

The edge prior was able to reconstruct the PPI network with an AUROC of 0.797 (dotted red line in Figure 7). Since the edge prior captures the performance of degree, this result indicated that nonspecific predictions alone showed this level of reconstruction performance. In the second comparison, edge prediction features computed on permuted networks showed performance equal or lower to their performances on the unpermuted networks. This indicated that four out of five edge prediction features picked up on more than node degree for the prediction task. The preferential attachment index is the product of source and target degree, and we found that its performance did not differ from the edge prior or the feature's performance when computed on permuted networks. The four remaining features showed far higher reconstruction performance than the edge prior or feature values computed on permuted networks.

This comparison quantified the performance of degree toward the prediction task and assessed degree's effect on five edge prediction features. The edge prior provided the baseline level of performance attributable to degree alone. Comparing the performances on permuted networks to the performance of the edge prior reveals the extent to which a feature captures degree. Features whose performances on permuted networks were below that of the edge prior only imperfectly captured degree (eg: Jaccard index), whereas features whose performances equaled the edge prior completely captured degree (eg: preferential attachment index).

Features can also capture information beyond degree, and our method can quantify this performance. For example, RWR captured more than degree because it reached all of a node's neighbors in one step after each restart. These results aligned with the definitions of each feature and validated that xswap accurately assessed reliance on degree.

Discussion

Prediction performance and the edge prior

We focus on edge prediction in biomedical networks. Our overall goal is to predict new edges with specificity, so that predictions reflect particular connectivity rather than generic node characteristics. Our permutation framework is designed to capture the predictive performance attributable to degree to provide a baseline expectation for edge pairs. We expect that degree-based non-specificity is not a unique property of biomedical networks. For example, if node A connects to nearly all other nodes in a network, predicting that all remaining nodes share an edge with node A will likely result in many correct–though nonspecific–predictions, regardless of the type of data contained in the network.

Node degree should be accounted for to make correct predictions while being able to distinguish specific from nonspecific predictions.

Prediction without reliance on node degree is challenging because many effective methods for edge prediction are correlated with degree (eg Figure 6). The effects of node degree are obvious when edge prediction features are functions of degree. For example, the resource allocation index is the sum of inverse degree of common neighbors between source and target nodes (in the symmetric case), while preferential attachment is the product of source and target degree [21,22]. However, because many other edge prediction methods are not explicitly degree-based, it is important to have a general method for comparing the effects of node degree on edge prediction methods.

We developed a permutation framework to quantify the edge probability due to degree. We term this probability the “edge prior”, and we have identified two applications. First, a probability associated with every node pair can be treated as a classification score. Ordering these scores provides an assessment of performance based solely on degree, which can be used as a baseline for other classifiers. Second, node pair probabilities can be used to adjust edge prediction features depending on the task. If degree is a desired feature, then the edge prior can be treated like a Bayesian prior probability. Alternatively, if degree is not a desired feature, then the edge prior can be used to calibrate features and thus potentially enhance predictive specificity.

Conclusion

References

1. **Systematic integration of biomedical knowledge prioritizes drugs for repurposing**
Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, Sergio E Baranzini
eLife (2017-09-22) https://doi.org/cdfk
DOI: 10.7554/elife.26726 · PMID: 28936969 · PMCID: PMC5640425

2. **Link Prediction Methods and Their Accuracy for Different Social Networks and Network Metrics**
Fei Gao, Katarzyna Musial, Colin Cooper, Sophia Tsoka
Scientific Programming (2015) https://doi.org/f7hvd9
DOI: 10.1155/2015/172879

3. **Bias tradeoffs in the creation and analysis of protein–protein interaction networks**
Jesse Gillis, Sara Ballouz, Paul Pavlidis
Journal of Proteomics (2014-04) https://doi.org/f3mn5f
DOI: 10.1016/j.jprot.2014.01.020 · PMID: 24480284 · PMCID: PMC3972268

4. **Correcting for the study bias associated with protein–protein interaction measurements reveals differences between protein degree distributions from different cancer types**
Martin H. Schaefer, Luis Serrano, Miguel A. Andrade-Navarro
Frontiers in Genetics (2015-08-04) https://doi.org/gf5t46
DOI: 10.3389/fgene.2015.00260 · PMID: 26300911 · PMCID: PMC4523822

5. **Effect of sampling on topology predictions of protein-protein interaction networks**
Jing-Dong J Han, Denis Dupuy, Nicolas Bertin, Michael E Cusick, Marc Vidal
Nature Biotechnology (2005-07) https://doi.org/dj5cm8
DOI: 10.1038/nbt1116 · PMID: 16003372

6. **Towards a proteome-scale map of the human protein–protein interaction network**
Jean-François Rual, Kavitha Venkatesan, Tong Hao, Tomoko Hirozane-Kishikawa, Amélie Dricot, Ning Li, Gabriel F. Berriz, Francis D. Gibbons, Matija Dreze, Nono Ayivi-Guedehoussou, … Marc Vidal
Nature (2005-09-28) https://doi.org/dw6q23
DOI: 10.1038/nature04209 · PMID: 16189514

7. **A Proteome-Scale Map of the Human Interactome Network**
Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, … Marc Vidal
Cell (2014-11) https://doi.org/f3mn6x
DOI: 10.1016/j.cell.2014.10.050 · PMID: 25416956 · PMCID: PMC4266588

8. **The Impact of Multifunctional Genes on "Guilt by Association" Analysis**
Jesse Gillis, Paul Pavlidis
PLoS ONE (2011-02-18) https://doi.org/bs9
DOI: 10.1371/journal.pone.0017258 · PMID: 21364756 · PMCID: PMC3041792

9. **Biology, Methodology or Chance? The Degree Distributions of Bipartite Ecological Networks**
Richard J. Williams
PLoS ONE (2011-03-03) https://doi.org/fmtk6x
DOI: 10.1371/journal.pone.0017645 · PMID: 21390231 · PMCID: PMC3048397

10. **The Degree Distribution of Networks: Statistical Model Selection**
William P. Kelly, Piers J. Ingram, Michael P. H. Stumpf
Bacterial Molecular Networks (2011-10-28) https://doi.org/ddx5rx
DOI: 10.1007/978-1-61779-361-5_13 · PMID: 22144157

11. **Scale-free networks are rare**
Anna D. Broido, Aaron Clauset
Nature Communications (2019-03-04) https://doi.org/gfztz9
DOI: 10.1038/s41467-019-08746-5 · PMID: 30833554 · PMCID: PMC6399239

12. **Addressing false discoveries in network inference**
Tobias Petri, Stefan Altmann, Ludwig Geistlinger, Ralf Zimmer, Robert Küffner
Bioinformatics (2015-04-24) https://doi.org/f7rwgt
DOI: 10.1093/bioinformatics/btv215 · PMID: 25910697

13. **Evidence of probabilistic behaviour in protein interaction networks**
Joseph Ivanic, Anders Wallqvist, Jaques Reifman
BMC Systems Biology (2008) https://doi.org/dsz4kn
DOI: 10.1186/1752-0509-2-11 · PMID: 18237403 · PMCID: PMC2267158

14. **Randomization Techniques for Graphs**
Sami Hanhijärvi, Gemma C. Garriga, Kai Puolamäki
Proceedings of the 2009 SIAM International Conference on Data Mining (2009-04-30)
https://doi.org/f3mn58
DOI: 10.1137/1.9781611972795.67

15. **STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets**
Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, … Christian von Mering
Nucleic Acids Research (2018-11-22) https://doi.org/gfz2jr
DOI: 10.1093/nar/gky1131 · PMID: 30476243 · PMCID: PMC6323986

16. **TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions**
Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, … Insuk Lee
Nucleic Acids Research (2017-10-26) https://doi.org/gcwpcz
DOI: 10.1093/nar/gkx1013 · PMID: 29087512 · PMCID: PMC5753191

17. **ChEA: transcription factor regulation inferred from integrating genome-wide ChIP-X experiments**
Alexander Lachmann, Huilei Xu, Jayanth Krishnan, Seth I. Berger, Amin R. Mazloom, Avi Ma'ayan
Bioinformatics (2010-08-13) https://doi.org/d2h98v
DOI: 10.1093/bioinformatics/btq466 · PMID: 20709693 · PMCID: PMC2944209

18. **Tracking the popularity and outcomes of all bioRxiv preprints**
Richard J. Abdill, Ran Blekhman
Cold Spring Harbor Laboratory (2019-01-13) https://doi.org/gftzwz
DOI: 10.1101/515643

19. **Complete Rxivist dataset of scraped bioRxiv data**
Richard J. Abdill, Ran Blekhman

Zenodo (2019-03-21) https://doi.org/gfz3fm
DOI: 10.5281/zenodo.2566421

20. **Degree-grouped permtuations by zietzm · Pull Request #96 · greenelab/hetmech**
GitHub
https://github.com/greenelab/hetmech/pull/96

21. **Predicting missing links via local information**
Tao Zhou, Linyuan Lü, Yi-Cheng Zhang
The European Physical Journal B (2009-10) https://doi.org/dd55vr
DOI: 10.1140/epjb/e2009-00335-8

22. **Link prediction approach to collaborative filtering**
Zan Huang, Xin Li, Hsinchun Chen
Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries - JCDL '05 (2005)
https://doi.org/fn39g8
DOI: 10.1145/1065385.1065415

23. **The link-prediction problem for social networks**
David Liben-Nowell, Jon Kleinberg
Journal of the American Society for Information Science and Technology (2007) https://doi.org/c56765
DOI: 10.1002/asi.20591

24. **Friends and neighbors on the Web**
Lada A Adamic, Eytan Adar
Social Networks (2003-07) https://doi.org/br5zd3
DOI: 10.1016/s0378-8733(03)00009-1

25. **Automatic multimedia cross-modal correlation discovery**
Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, Pinar Duygulu
Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04 (2004) https://doi.org/bmhgw4
DOI: 10.1145/1014052.1014135

26. **Learning with local and global consistency**
Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, Bernhard Scholkopf
NIPS 2003 (2003) https://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf

27. **Link prediction in large directed graphs**
Dario Garcia Gasulla
Universitat Politècnica de Catalunya (2015) https://upcommons.upc.edu/handle/2117/95691

Supplemental information

Approximate edge prior

To approximate the edge prior, we began by making two simplifications. First, we assumed independence between node pairs. This assumption does not actually hold for the XSwap algorithm, though it is a reasonable simplification for large, sparse networks. Second, we assumed that the XSwap process is stationary. This assumption also does not actually hold, but it was made because it significantly simplifies the problem. A single node pair has two possible states, “edge” and “no edge”. These states are not transient, and they are not periodic so long as more than one possible swap

exists in the network. In almost all cases, then, our simplified model of the algorithm gives the state of a node pair as an ergodic process, independent of other node pairs.

Let $A_{i,j}$ represent the existence of edge (i, j) For a given node pair, (i, j), then, let $q_{i,j}$ represent the transition probability from the “no edge” state to the “edge” state in one successful iteration of the XSwap algorithm. Let $r_{i,j}$ represent the probability of the opposite transition (“edge” to “no edge”) in one successful iteration. With “no edge” represented as $[1, 0]^T$ and “edge” represented as $[0, 1]^T$, the transition matrix, P, is given by the following:

$$P^T = \begin{bmatrix} 1-q & r \\ q & 1-r \end{bmatrix}$$

The stationary distribution of this system should correspond to the distribution when the number of swaps goes to infinity. It can be found by computing the eigenvectors of the system, as we know that the stationary distribution vector, $\mathbf{v}$ satisfies $P^T\mathbf{v} = \mathbf{v}$. The normalized eigenvector $\mathbf{v}$ is given by

$$\mathbf{v} = \frac{1}{r/q + 1} \begin{bmatrix} r/q \\ 1 \end{bmatrix}$$

The asymptotic edge probability is therefore

$$\frac{1}{r/q + 1}.$$

Since node pairs are being treated as independent, the probability of an edge being created in one successful iteration, given that the edge does not currently exist, is the ratio of the number of edge choices involving nodes i and j to the total number of possible swaps, S. Let $d(u_i)$ represent the degree of source node i and $d(v_j)$ represent the degree of target node j.

$$q_{i,j} = \frac{d(u_i)d(v_j)}{S}$$

Similarly, the probability of an edge being eliminated in one iteration is the ratio of the number of edge choices involving (i, j) and any other valid edge to the total number of possible swaps. Let m be the total number of edges in the network.

$$r_{i,j} = \frac{m - d(u_i) - d(v_j) + 1}{S}$$

The approximate edge prior is, therefore,

$$\frac{d(u_i)d(v_j)}{m - d(u_i) - d(v_j) + 1 + d(u_i)d(v_j)}.$$

Unfortunately, we found that the above edge prior approximation is a poor approximation in many cases. We found that the following modified form (introduced in Methods) affords a superior approximation:

$$P_{i,j} = \frac{d(u_i)d(v_j)}{\sqrt{(d(u_i)d(v_j))^2 + (m - d(u_i) - d(v_j) + 1)^2}}$$

Because the modified form of the approximation offers a much superior fit to the data, we chose to include only the modified version in the Python package released, and we used only the modified form throughout our analysis.

Networks used for comparison

Data	Network	Nodes	Edges
Hetionet	AdG	Source: 402, Target: 20945	102240
	AeG	Source: 402, Target: 20945	526407
	AlD	Source: 402, Target: 137	3602
	AuG	Source: 402, Target: 20945	97848
	BPpG	Source: 11381, Target: 20945	559504
	CCpG	Source: 1391, Target: 20945	73566
	CbG	Source: 1552, Target: 20945	11571
	CcSE	Source: 1552, Target: 5734	138944
	CdG	Source: 1552, Target: 20945	21102
	CrC	1552	6486
	CuG	Source: 1552, Target: 20945	18756
	DaG	Source: 137, Target: 20945	12623
	DdG	Source: 137, Target: 20945	7623
	DpS	Source: 137, Target: 438	3357
	DuG	Source: 137, Target: 20945	7731
	GuG	20945	265672
	GcG	20945	61690
	GiG	20945	147164
	GpMF	Source: 20945, Target: 2884	97222
	GpPW	Source: 20945, Target: 1822	84372
PPI	Sampled	3992	255522

	Literature	3992	364743
	Systematic	3916	12913
	Sampled	4587	30686
bioRxiv	<2018	4615	43691
	All time	4615	44963
TF-TG	Sampled	Source: 142, Target: 1396	2689
	Literature	Source: 144, Target: 1406	3496
	Systematic	Source: 144, Target: 1417	29177

Edge prediction features

In the table that follows, let $k(u)$ denote the set of neighbors of node u. Let $\mathbf{A}$ represent the normalized Laplacian adjacency matrix, and let y_u be a vector with all ones except for a one in the u-th position. x For a directed graph, let $A(u)$ denote the set of nodes that node u points to and $D(u)$ the set of nodes that point to u. All definitions that follow are the score between nodes u and v.

Table 2: Edge prediction features.

Feature	Definition	Citation
Jaccard index	$\frac{\lvert k(u)\cap k(v)\rvert}{\lvert k(u)\cup k(v)\rvert}$	[23]
Preferential attachment score	$\lvert k(u)\rvert\lvert k(v)\rvert$	[23]
Resource allocation index	$\sum_{w\in k(u)\cap k(v)}\frac{1}{\lvert k(w)\rvert}$	[21]
Adamic/Adar index	$\sum_{w\in k(u)\cap k(v)}\frac{1}{log\lvert k(w)\rvert}$	[24]
Random walk with restart score	$c\left[\left(\mathbb{I}-(1-c)\mathbf{A}\right)^{-1}\mathbf{y}_u\right]_v$	[25,26]
Inference score	$\frac{\lvert A(u)\cap D(v)\rvert}{\lvert A(u)\rvert}+\frac{\lvert D(u)\cap D(v)\rvert}{\lvert D(u)\rvert}$	[27]