2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Link prediction: fair and effective evaluation

Ryan Lichtenwalter and Nitesh V. Chawla
Department of Computer Science
The University of Notre Dame
Notre Dame, IN 46556
Email: rlichten@nd.edu, nchawla@nd.edu
Phone: 1-574-631-7095

***Abstract*—Link prediction is a popular area for publication. Papers appear in virtually every conference on data mining or network science with new methods. We argue that the practical performance potential of these methods is generally unknown because of challenges endemic to evaluation in many link prediction contexts. We demonstrate that current methods of evaluation are inadequate and can lead to woefully errant conclusions about practical performance potential. We argue for the use of precision-recall threshold curves and associated areas in lieu of receiver operating characteristic curves due to the extreme imbalance of the link prediction classification problem. We provide empirical examples of how current methods lead to questionable conclusions, how the fallacy of these conclusions is illuminated by methods we propose, and suggest a fair and consistent framework for link prediction evaluation for longitudinal and non-longitudinal network data sets.**

I. Introduction

Link prediction generally stated is the task of predicting relationships in a network. Typically it is approached specifically as the task of predicting new links given some set of existing nodes and links. Existing nodes and links may be present from a prior time period from longitudinal data or they may be some portion of the topology in a network whose exact topology is difficult to measure. In the former case, general link prediction is useful to anticipate future behavior. In the latter case, it can identify or substantially narrow possibilities that are difficult or expensive to determine through direct experimentation [1] [2] [3]. Thus, even in domains where link prediction can seem impossibly difficult or offer a high ratio of false positives to true positives, it may be useful [4].

Link prediction involves all the complexities of evaluating ordinary binary classification, but it includes several new parameters and intricacies that make it fundamentally different. We depict the framework for evaluation [5] [6] [7] in Figure 1. Computations occur within network snapshots based on particular segments of data. Whether the predictor is unsupervised or supervised, it must be evaluated on the same instances defined by the same segment. For supervised methods, the division of the prediction network into a training and label network is not strictly necessary since all edges from the prediction network might be temporarily removed to calculate features for instances with a positive training label, and missing edges in the prediction network might be used to calculate features for instances with negative training labels.

Link prediction in longitudinal data should ideally be performed with a longitudinal approach. Removing and subsequently predicting "test edges" should only be a last resort in networks where multiple snapshots do not exist. Even in protein-protein interaction data, it is possible to use high-confidence and low-confidence edges to construct different networks for evaluation with greater meaning. Removing and predicting edges removes information from the original network in unpredictable ways [8], and the removed information has the potential to affect prediction methods differently. More significantly, randomly sampling edges for testing from a single data set in a supervised approach [9] reflects prediction performance with respect to a random growth process instead of the real growth process underlying the longitudinal network, variable confidence data, or whatever other regulatory mechanisms may pertain.

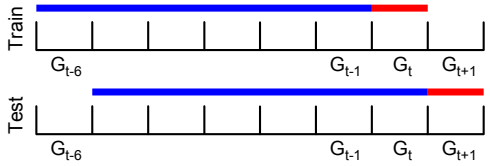


Fig. 1. Link prediction and evaluation.

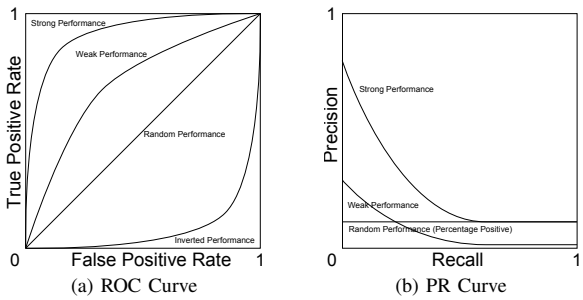
Another issue is directionality, for which there is no analog in typical classification tasks. In undirected networks, the same method may predict two significantly different results for a link between v_a and v_b depending on the order in which vertices are presented. There can only be one final judgment of whether the link will form, but that judgment differs depending on which vertex you ask. We expand upon this in Section IV.

Test set sampling is a ubiquitous practice in link prediction evaluation [5] [7] [9] [10] [11] [12] [13] [14] [15], and we analyze how such sampling should be conducted to produce fair and meaningful results. The reason sampling is necessary, and a primary reason link prediction is such a challenging domain within which to evaluate and interpret performance, is its extreme class imbalance. We extensively analyze issues related to sampling in Section V and cover the significance of class imbalance on evaluation in III-D. Fairly and effectively evaluating a link predictor requires determining which evaluation metric to use (Sections III and VII), whether to restrict the enormous set of potential predictions, and how best to restrict the set if so.

All of these issues stand in the way of producing fair,

978-0-7695-4799-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ASONAM.2012.68

comparable results across published methods. Perhaps even more importantly, they interfere with rendering judgments of performance that indicate what we might really expect of our prediction methods in deployment scenarios. It is virtually impossible to compare from one paper to the next, and some of the currently employed evaluation methods can produce results that are unfairly favorable to a particular method or otherwise unrepresentative of expected deployment performance. We seek to provide a reference for important issues to consider and a set of coherent standards as recommendations to those performing link prediction research.



(a) ROC Curve (b) PR Curve

Fig. 2. A visualization of the two theshold curves used throughout this paper.

II. Data and Methods

We report all results on a single, relatively small publicly available longitudinal data set. The data set, to which we will henceforth refer as condmat, is constructed by moving through a sequence of collaboration events in the condensed matter physics community. Each collaboration of k individuals forms an undirected k-clique with weights in inverse linear proportion to k. The network is thus weighted and undirected.

We illustrate our points using results for the efficacy of only three prediction methods, but each method represents a different modeling approach. The preferential attachment predictor [16] uses degree product and represents predictors based on node statistics. The Adamic/Adar predictor [17] represents the family of common neighbors predictors. The PropFlow predictor [7] represents the family of predictors based on paths and random walks.

We emphasize here that the point of this work is not to illustrate the superiority of one method of link prediction over another. It is instead to point out that the described effects and arguments have real impacts on performance and evaluation. If we show that the effects pertain in at least one network, then they may exist in others and must be considered. We also selected condmat because its small size makes exploring some of these problems computationally easier. Larger and sparser networks will typically exhibit the patterns we describe to a greater degree rather than to a lesser degree.

III. Evaluation Metrics

The evaluation metrics typically used in link prediction are much the same as those used in any binary classification task. They can be divided into two broad categories: fixed-threshold metrics and threshold curves. All fixed-threshold metrics suffer from the limitation that some estimate of a reasonable threshold must be available. This is rarely true in research contexts where we are curious about the performance without necessarily being attached to any particular domain or deployment. Threshold curves such as the receiver operating characteristic (ROC) curve [18] and derived curves like cost curves [19] and precision-recall curves [20] provide alternatives in these cases.

A. Fixed-Threshold

In general, fixed-threshold metrics may rely on different types of thresholds: based on prediction score, based on percentage of instances, or based on number of instances. In link prediction specifically, there are additional constraints. Many link prediction methods do not produce scores with any reasonable correspondence to likelihood. For instance, it is rarely sensible to say that two vertices with a degree product of 10,000 are 10 times as likely to form a new link as two with a degree product of 1000.

When considering links by geodesic distance, which we shall advocate, it makes little sense to speak about the top k results as a fraction of the total number of potential links. The resources to explore potential positives do not change because the potential positives happen to span a larger distance. It is appropriate to use top-k percentages only within the 2-hop distance or the complete listing of scores for all potential links and only when there is a reasonable expectation that k is logical for the problem at hand. When using an absolute number of instances k, if the data does not admit a trivially simple class boundary, we set classifiers up for failure by presenting them with class ratios of millions to one and taking only some few k. Further, if k is small and the number of positives is small, the metric becomes highly unstable.

Therefore, while accuracy [9] [10] [21] [22], precision [5] [9] [6] [21], recall [9] [21] [22], and top-k equivalents [5] [6] [22] are used commonly in link prediction literature, we caution trusting any results that come only in terms of fixed thresholds [9] [21] [22]. In poorly calibrated scores, it is also unclear what these terms mean when they appear unqualified.

B. Threshold Curves

Due to the rarity of cases when researchers are in possession of reasonable fixed thresholds, threshold curves are commonly used in the binary classification community to express results. They are especially popular when the class distribution is highly imbalanced. and hence are used increasingly commonly in link prediction evaluation [5] [7] [23]. Threshold curves also admit scalar measures, which serve as a single summary statistic of performance. The ROC curve shows the true positive rate with respect to the false positive rate at all classification thresholds, and its area (AUC) is equivalent to the probability of a randomly selected positive instance appearing above a randomly selected negative instance. We will use ROC curves and precision-recall curves to illustrate our points and eventually argue for the use of precision-recall

curves and areas. Figure 2 illustrates a depiction of the two curve metrics.

C. Improvement Factors

For any scalar performance indicator, one might examine improvement factors or ratios [12]. Such factors are typically constructed by dividing the performance numbers of competing methods by the performance number of some established baseline. The advantage of this method is that it normalizes the evaluation measure, and all comparisons that use it are fair. The disadvantage is that all information about absolute predictive capability is lost unless clearly provided. This makes it easy to use improvement factors to report stunning numbers that actually indicate very little about performance. Reporting a 1000x factor of improvement in precision over a random model sounds impressive. It sounds much less impressive when the original precision is 10^{-7} and the minimal number of positives makes the random model so unstable that random models can easily achieve large factors of improvement over themselves. It is important to consider the stability of this measure when deploying it. Finally, when used with single-threshold metrics, factors of improvement likewise rely on arbitrary thresholds.

D. Class Imbalance

In typical binary classification tasks, class ratios are approximately balanced. We expect the probability of randomly selecting a positive instance to be approximately equal to the probability of randomly selecting a negative instance. As a result, we can calculate expectations for baseline classifier performance. For instance, the expected accuracy ($\frac{TP+TN}{TP+FP+TN+FN}$), precision ($\frac{TP}{TP+FP}$), recall ($\frac{TP}{TP+FN}$), AUC, and PR curve area of a random classifier, an all-positive classifier, and an all-negative classifier are 0.5 with an even prior distribution.

Binary classification problems that exhibit class imbalance do not share this property, and the link prediction domain is an extreme example. The expectation for each of the classification metrics diverges for random, all-positive, and all-negative classifiers. Accuracy is problematic because its value for all-negative predictors converges to 1. At the same time, correct classification of rare positive instances is usually more important since they tend to represent exceptional cases of high relative interest. Since classification is an exercise in optimizing some measure of performance, we must not select a measure of performance that optimizes toward a useless result. ROC curves offer a baseline random performance of 0.5 and penalize all-positive and all-negative predictors. Optimizing ROC optimizes the production of class boundaries that maximize TP while minimizing FP. Precision and PR curves offer baseline performance calibrated to the class balance ratio, and this can serve to maintain a sobering link to reality in expectations of performance.

IV. Directionality

In undirected networks, an additional methodological parameter pertains in the task of evaluation, which is rarely reported. With directed networks, at least in the case of single-mode predictions, a prediction is uniquely specified by an *ordered* pair of vertices. Order is implied by edge directionality. With undirected networks, no ordering is implied, so for any given pair of edges, there are potentially two different prediction outputs. For some prediction methods, such as those based on node properties or common neighbors, the prediction output remains the same irrespective of ordering. This is not true in general. Most notably, many prediction methods based on paths and walks implicitly depend on notions of source and target even if there is no directionality in the edges.

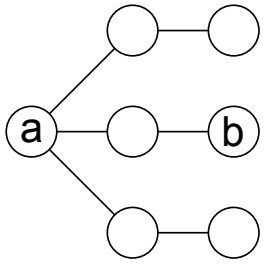


Fig. 3. A simple undirected graph example to illustrate that predictions for a link from v_a to v_b may differ according to which vertex is considered the source.

Contemplate Figure 3 with the goal of predicting a link between v_a and v_b. Consider the percentage of two-hop paths starting from v_a that reach v_b versus the percentage of two-hop paths starting from v_b that reach v_a. Clearly, all two-hop paths originating at v_b reach v_a whereas only a third of the two hop paths originating at v_a reach v_b. In a related vein, consider the probability of reaching one vertex from another in random walks. Clearly all walks starting at v_b that travel at least two hops must reach v_a whereas the probability of reaching v_b from v_a in two hops is lower. Topological prediction outputs may diverge whenever v_a and v_b are in different automorphism orbits within a shared connected component.

This raises the question of how to determine the final output of a method that produces two different outputs depending on the input. Naturally any functional mapping from two values to a single value will suffice. Selection of an optimal method will depend on both the predictor and the scenario and is outside the scope of this paper. Nonetheless, it is important for reasons of reproducibility not to forget or neglect this question when describing results. When a method or a method component respects notions of source and target, researchers must specify how they resolve the question on any undirected networks that they use.

The approach consistent with the process for directed networks would be to generate a ranked list of scores that includes predictions with each node alternately serving as source and destination. This approach is workable in a deployment scenario, since top-ranked outputs may be selected as predicted links regardless of the underlying source and target. It is not feasible as a research method for presenting results, however, because the meaning of the resulting threshold curves is ambiguous. There is also no theoretical reason to suspect any

sort of averaging effect in the construction of threshold curves.

To emphasize this empirically, we computed ROC areas for two methods using the PropFlow predictor. The first method includes a prediction in the output for both underlying orderings, and the resulting area is 0.610. The second method computes the arithmetic mean of the predictions from the two underlying orderings to produce a single final prediction for the rankings, and the resulting area is 0.625.

V. Test Set Sampling

Test set sampling is popular in link prediction domains because sparse networks usually include only a tiny fraction of the $O(|V|^2)$ links supported by the collection of vertices. Each application of link prediction must provide outputs for what is essentially the entire set of $\binom{|V|}{2}$ links. For even moderately sized networks, this is an enormously large number that places unreasonable demands on processing resources and storage. As a result, there are many sampling methods for link prediction testing sets. Each method presents advantages aligned with its goals, but some methods may be fairer than others. All at least conceptually start with the complete set of potential edges that could form between the predictor network and the testing network. One common method is selecting a subset of edges at random from the original complete set [5] [9] [14]. Another is to select only the edges that span a particular geodesic distance [7] [15]. Yet another is to select edges so that the sub-distribution composed by a particular geodesic distance is approximately balanced [5] [13]. Finally any number of potential methods can select edges that present a sufficient amount of information along a particular dimension [11] [12], for instance selecting only the edges where each member vertex has a degree of at least 2.

When working with threshold-based measures, any sampling method that removes negative class instances above the decision threshold score unfairly and potentially unpredictably raises most information retrieval measures. Precision is inflated by the removal of false positives. In top-k measures, recall is inflated by the opportunity for additional positives to appear above the threshold after the negatives are removed. This naturally affects the harmonic mean, F-measure. Accuracy is affected by any test set modification since the number of mis-classifications may change. Clearly we cannot report meaningful results with these threshold-based measures when performing any type of sampling on the test set, but researchers have inadvertently reported such results [5] [9]. The question is whether it is fair to sample the test set when evaluating with ROC curves or other threshold curves based on points in ROC space.

At first it may seem that subsampling positives from the test set has no negative effects on ROC curves and areas. There is a solid theoretical basis for this belief, but issues specific to link prediction relating to extreme imbalance cause problems in practice. We will first describe these problems, and why using evaluation methods involving extreme subsampling are problematic. Then we will show that test set sampling actually reveals what we believe is a much more significant problem with the testing distribution that sampling attacks.

A. ROC Robustness to Class Imbalance

Theoretically, ROC curves and their associated areas are unaffected by changes in class distribution alone. This is one source of great appeal as compared to accuracy since they will render consistent judgments even as imbalance becomes increasingly extreme. As a result, it is theoretically possible to fairly sample negatives from the test set without affecting ROC results. The proper way to model fair random removals of test instances closest to the actual ROC curve construction step is to randomly remove items from the unsorted or sorted list of output scores. As long as the distribution remains stable in the face of random removals, the ROC curve and area will remain unchanged.

We do not want to waste the effort necessary to generate lists of output scores only to actually examine a fractional percentage of them. We must instead find a way to transfer our fair model of random removals in the ranked list of output scores to a network sampling method while theoretically preserving all feature distributions. The solution is to randomly sample edges. So given a network in which our original evaluation strategy was to consider a test set with every potential edge based on the previously observed network, we generate a list of edges that do not exist in the test period until we achieve the number we require.

We demonstrate the results of this strategy empirically in Figure 4. The ROC area remains stable down to 1% negative sampling. Below this, it is affected. While stability to 1% sampling may seem quite good, it is critical to note that the imbalance ratios of link prediction in large networks are such that 1% of the original test set likely still contains an unmanageable number of instances. Notably, the dashed vertical line shows the ROC area for sampling that produces a balanced test set. The area deviates by more than 0.007 for PropFlow and more than 0.01 for preferential attachment, which may exceed significant variations in performance across link predictors. Further sampling causes even greater deviations so that significance is almost certainly lost. These deviations are not a weakness of the ROC area itself but are indicative of instability in score rankings within the samples. This instability does not manifest itself uniformly, and it may be greater for some predictors than for others. In this example, preferential attachment exhibits greater susceptibility to the effect.

This result calls into question the soundness of the practice of undersampling link prediction test sets to balance. Worse news is that there is no necessary lower bound on how bad the effect can be. The greater the imbalance ratio and the smaller the absolute number of positives the more results will skew. One cannot easily argue against the presence of the effect without repeated evaluation or evaluation using a more complete test set.

If the original goal is to reduce data volume for more manageable testing, then clearly sampling is problematic. A

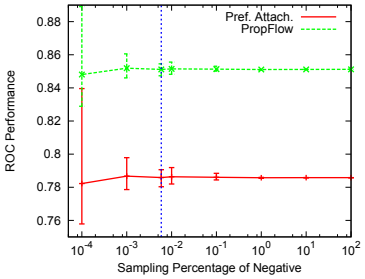
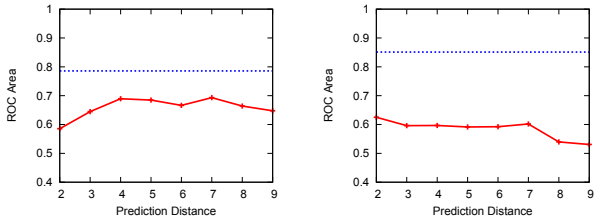


Fig. 4. Effect of test set negative sampling on ROC area. The vertical line indicates class balance. It makes no sense to consider this domain for PR area since the undersampling inherently affects the measure.



(a) Preferential Attachment (b) PropFlow

Fig. 5. The ROC curve performance of the preferential attachment link predictor over each neighborhood. The horizontal line represents the performance apparent by considering all potential links as a corpus.

reduction factor of 2 still results in $O(|V|^2)$ test instances and the same order of magnitude. A reduction factor of 100 or 1000, which may still provide huge data sets in link prediction terms, potentially reduces the stability of results to a degree that may exceed the difference in performance across methods.

B. The Real Testing Distribution

The question should naturally arise of what performance we report when we undersample link prediction test sets. Presumably undersampling is part of an attempt at combating unmanageable test set sizes and describing the performance of the predictor on the network as a whole. This type of report is common, and issues of stability aside, it is clearly theoretically valid. We question, however, whether the results that it produces actually mean anything. Figure 5 compares ROC performance overall to the ROC performance achievable in the distinct sub-problem created by dividing the task by geodesic distance.

We will first consider the results for the preferential attachment predictor. The general conclusion is that the apparent achievable performance is remarkably higher, 13.4%, in the complete set of potential edges than the performance achievable by the same prediction method in data sets restricted by distance. The explanation is that the extreme importance of geodesic distance in determining link formation correlates distance highly with any successful prediction method. The high-distance regions contain very few positives and effectively append a set of trivially recognizable negatives to the end. This increases the probability of a randomly selected positive appearing above a randomly selected negative, the ROC area.

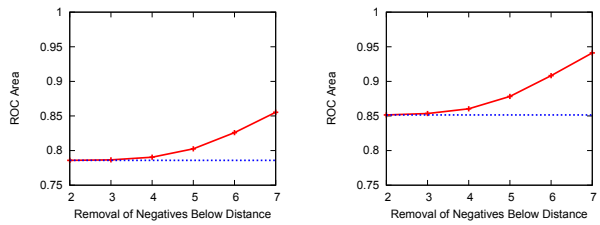
The effect is exaggerated for PropFlow and for other ranking methods that inherently scale according to distance such as rooted PageRank and Katz. In those cases, the ROC curve for the amalgamated data approximates concatenation of the individual ordered outputs, which inherently places the distances with higher imbalance ratios at the end where they inflate the overall area. Figure 5 shows the effect for the PropFlow prediction method on the right. We chose PropFlow only because it is faster to compute and its inherent distance scaling is deterministic.

For PropFlow, the apparent achievable performance is 36.2% higher for the overall score ordering than for the highest of the individual orderings! This result also has important implications from a practical perspective. PropFlow appears to have a higher ROC than preferential attachment, but the *only* distance at which it outperforms preferntial attachment is the 2-hop distance. Preferential attachment is a superior choice for the other distances in cases where the other distances matter. These important details are hidden from view by ROC space. They also illustrate that the performance indicated by overall ROC is not meaningful with respect to deployment expectations and that it conflates performance across neighborhoods within a bias toward rankings that inherently reflect distance.

Consider the data distribution of the link prediction problem used in this paper. There are 148.2 million negatives and 29,898 positives. The ratio of negatives to positives is 4,955 to 1. There are 1196 positives and 214,616 negatives in the 2-hop neighborhood. To achieve a 1 to 1 ratio with random edge sampling, statistical expectation is for 43.3 2-hop negatives to remain. The 2-hop neighborhood contains 30% of all positives, so clearly it presents the highest baseline precision. That border is the most important to capture well in classification, because improvements in 2-hop discrimination are worth much more than improvements at higher distances. 16% of all positives are in the 3-hop neighborhood, so the same argument applies with it versus higher distances.

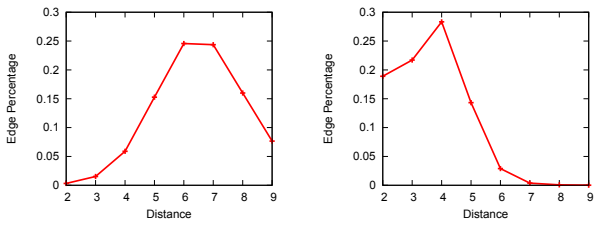
The real data distribution on which we report performance when we perform this sampling is the relatively easy boundary described by the highly disparate 2-hop positives and high-distance negatives. Figure 6 further substantiates this point by illustrating the minimal effect of selectively filtering all negatives from low-distance neighborhoods. We know that performance in the 2-hop neighborhood is most significant because of the favorable class ratio and that improvements in defining this critical boundary offer the greatest rewards. Simultaneously, as the figure shows, we can entirely remove all 2-hop negatives with only a 0.2% effect on the ROC curve area for two predictors from entirely different families. We have to remove all 2-hop negatives, all 3-hop negatives, and even all 4-hop negatives before the alteration becomes highly conspicuous, yet these are the significant boundary instances.

Since we also know that data distributions do not affect ROC curves, we can extend this observation even when no sampling is involved: considering the entire set of potential links in ROC space evaluates prediction performance of low-



(a) PreferentialAttachment (b) PropFlow

Fig. 6. The effect of removing negative instances from increasingly distant potential links. The horizontal line represents the base ROC on the unsampled test set.



(a) condmat (b) Kaggle

Fig. 7. Distribution of distances in test sets.

distance positives versus high-distance negatives. We need to describe performance within the distribution of 2-hop positives and 2-hop negatives and select predictors that optimize this boundary.

Of the many sampling methods introduced, the random removal of edges from the complete set of potential edges is the only way to preserve the testing distribution. However questionable the meaning of this distribution may be with respect to deployment potential, it is at least fair and comparable. One recently employed alternative appears to take another sampling approach: aggressively subsampling negatives from over-represented distances while keeping the same number of low-distance instances in the distribution. The Kaggle link prediction competition [13] undersampled the testing set by manipulating the amount of sampling from each neighborhood to maintain approximate balance within the neighborhoods. The distribution of distances exhibited by the 8960 test edges is shown in Figure 7.

Consider the results of Figure 7 against the results of fair random sampling in the condmat network. Unless Kaggle has an incredibly small effective diameter, it is impossible to obtain this type of distribution. It requires a sampling approach that includes low-distance edges from the testing network with artificially high probability. While this selective sampling approach might seem to better highlight some notion of average boundary performance across neighborhoods, it is instead meaningless because it creates a testing distribution that would never exist in deployment. The Kaggle competition disclosed that the test set was balanced. In a deployment scenario, it is impossible to provide to a prediction method a balance of positives and negatives from each distance, because that would require knowledge of the class labels that are the object of the prediction task. Assuming that the balanced distribution is not maintained within neighborhoods, the test set still suffers from the problem that without knowledge of class labels, samples of similar size across distances would reward a purely distance-based predictor because of the extreme probability of a 100% negative distribution in 3-hop or higher distances.

Worse, it is unfair and incomparable because the original distribution is not preserved, and there is no reason to argue for one arbitrary manipulation of distance prevalence over another. Simultaneously, the ROC area will vary greatly according to each distributional shift. It is even possible to predictably manipulate the test set to achieve a virtually arbitrary ROC area by changing the distribution in this manner. Clearly this approach has problems because any results are inextricably tied to decisions about sampling that require the very results we are predicting in a deployment scenario. This was not a problem for Kaggle because all the players were playing the same game, but it would be a mistake to believe that the results are in general indicative of real achievable performance or even of a proper ranking of models in a genuine prediction task.

VI. New Nodes

There are two fundamentally different ways to generate test sets in link prediction. The first is to create a set of potential links by examining the predictor network and selecting all pairs for which no edge exists. Positives are those among the set that subsequently appear in the testing network, and negatives are all others. The second is to use the testing network to generate the set of potential links. Positives are those that exist in the testing network but not in the training network, and negatives are those that could exist in the testing network but do not. The subtle difference lies in whether or not the prediction method is faced with or penalized for links that involve *nodes* that do not appear in the predictor network.

The choice we should make depends on how the problem is posed. If we are faced with the problem of returning a most confident set of predictions, then new nodes in the testing network are irrelevant. Although we could predict that an existing node will connect to an unobserved node, we cannot possibly predict what node the unobserved node will be. Unless we consider predicting a link to some unspecified new nodes as predicting a link to a specific new node, there is no reasonable way to reward the prediction. Surely such a reward should not be counted at the same time as precision with respect to predicting specific links.

If we are faced with the problem of answering queries, then the ability to handle new nodes is an important aspect of performance. On one hand, we could offer a constant response, either positive or negative, to all queries regarding unfamiliar nodes. The response to offer and its effect on performance depend on the typical factors of cost and expected class distribution. On the other hand, some prediction methods may support natural extensions to provide a lesser amount of

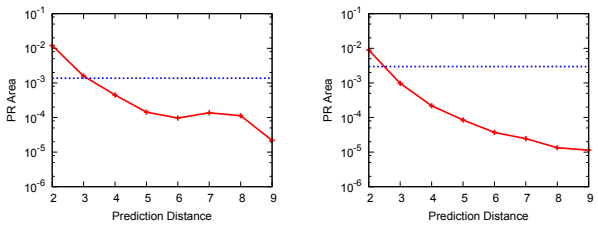
information in such cases. For instance, preferential attachment could be adapted to assume a degree of 1 for unknown nodes. Path-based predictors would have no basis to cope with this scenario whatsoever. In supervised classification, any such features become missing values and the algorithm must support such values.

Evaluating with potential links drawn from the testing network is problematic for decomposing the problem by distance since the distance must be computed from single-source shortest paths based on the pretend removal of the link that appears only in the testing network. Since distance is such a crucial player in determining link likelihood in most networks, this would nonetheless be an early step in making a determination about link formation likelihood in any case, so its computation for creating divided test sets is probably unavoidable. Given the extra complexity introduced by using potential link extraction within the testing network, we opt for determining link pairs for testing based on training data unless there is a compelling reason why this is unsatisfactory.

VII. The Case for Precision-Recall Curves

ROC curves and areas make sense for typical data imbalance scenarios because they optimize for the performance of interest, and because the appearance of the curve provides a reasonable visual indicator of expected performance. It is typical to hear statements by researchers examining ROC curves such as: "Wow! 0.99 AUROC. This is a trivially easy classification problem." That statement is true in most scenarios because data set sizes are relatively small (10^3 to 10^6) and because imbalance ratios are relatively modest (2 to 20). Corresponding precisions are near 1. For complete link prediction in sparse networks, when every potential new edge is classified, the imbalance ratio is *lower* bounded by the number of vertices in the network [7]. ROC curves and areas are deceptive. In a network with millions of vertices, even with an exceptional AUC of 0.99 one could easily suffer small fractions as a maximal precision. The same researchers would usually consider the outcome unacceptable when performance is put in these terms. In most domains, examining several million false positives to find each true positive *is* the classification problem. Even putting aside more concrete theoretical criticisms of ROC curves and areas [24], in link prediction tasks they fail to honestly convey the difficulty of the problem and reasonable performance expectations for deployment.

Precision-recall (PR) curves provide a more discriminative view of classification performance in extremely imbalanced contexts such as link prediction [20]. Like ROC curves, PR curves are threshold curves. Each point corresponds to a different score threshold with a different precision and recall value. In PR curves, the x-axis is recall and the y-axis is precision. We will now revisit a problematic scenario that arose with ROC curve areas and demonstrate that PR curve areas present a less deceptive view of what is actually present in the performance data. Notably, many PR curve construction procedures will require that negatives are not subsampled from the test set. This is not problematic in the consideration of distance-restricted predictions.



(a) Preferential Attachment (b) PropFlow

Fig. 8. The precision-recall curve performance of the preferential attachment link predictor over each neighborhood. The horizontal line represents the performance apparent by considering all potential links as a corpus.

Note in Figure 8 that the PR curve area is higher for the 2-hop distance than it is for the complete data set. In the underlying curves, this is exhibited as much higher precisions throughout but especially for low to moderate values of recall. Performance by distance exhibits expected monotonic decline due to increasing baseline difficulty excluding the instabilities in very high distances. Compare this to Figure 5 where the ROC area for all potential links was much greater than for any neighborhood individually, and the apparent performance was greatest in the 7-hop distance data set.

VIII. Conclusion

To select the best technique, we must know how to evaluate our techniques. Beyond this, we must be sure that readers do not come away from papers with the question of how the method *actually* performs. It is more difficult to specify and explain link prediction evaluation strategies than with standard classification wherein it is sufficient to fully specify a data set, one of a few evaluation methods, and a given performance metric. In link prediction, there are many potential parameters often with many undesirable options. There is no question that the issues raised herein *can* lead to questionable or misleading results. Hopefully the empirical demonstrations convince the reader that they *do* lead to questionable or misleading results. We propose the following guidelines:

1) Use precision-recall curves and curve areas as an evaluation measure. Use ROC curves and areas as optional accompaniment. Avoid fixed thresholds.
2) Render prediction performance evaluation by distance.
3) Do not undersample negatives from test sets, which will be of more manageable size due to consideration by distance.
4) If negative subsampling is undertaken for some reason, it must be based on a purely random sample of edges missing from the test network. It must not modify dimensions in the original distribution. Naturally *any* sampling must be *clearly* reported.
5) In undirected networks, state if the method is invariant to designations of source and target. If it is not, state how the final output is produced.

6) Always take care to use the same testing set instances regardless of the method used for training.
7) In longitudinal data, the final test set on which evaluation is performed should receive labels from a subsequent, unobserved snapshot of the data stream generating the network.
8) Consider whether the task is solving the recommendation problem or the query problem and construct test sets accordingly.

Much of this paper relies upon the premise that the class balance ratio differs, even differs wildly, across distances. There are certainly rare networks where such an expectation is tenuous, but the premise holds without exception in every network with which the authors have worked including networks from the following families: biology, commerce, communication, collaboration, and citation. Alternatively, distance might simply be taken as a parameter to a model and thus incorporated into the predictions. This approach lacks some of the other advantages that distance-restricted testing offers, most notably speed and efficiency but probably also efficacy.

Consider somebody asking you to describe the achievable performance of your link prediction strategy. You answer by reporting the ROC area you can achieve in the amalgamated testing data. The number seems quite high and impresses your listener. You then indicate that deploying the associated classifier will actually result in very low performance in terms of precision. The classifier that achieves the highest precisions relative to recall in the overall data set may not even be the optimal classifier for virtually any threshold your listener would choose in deployment. You indicate that you also have a model that operates only over a 2-hop distance, that it has a much lower ROC area, but that it will provide much better performance. It is natural to question why you did not simply report the second model and number initially instead of another number describing a model that you would never actually use or deploy!

ACKNOWLEDGMENT

Research was sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053, and in part by the National Science Foundation (NSF) Grant BCS-0826958. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] N. Martinez, B. Hawkins, H. Dawah, and B. Feifarek, "Effects of sampling effort on characterization of food-web structure," *Ecology*, vol. 80, no. 3, pp. 1044–1055, 1999.

[2] E. Sprinzak, S. Sattath, and H. Margalit, "How reliable are experimental protein-protein interaction data?" *Journal of Molecular Biology*, vol. 327, no. 5, pp. 919–923, 2003.

[3] A. Szilágyi, V. Grimm, A. Arakaki, and J. Skolnick, "Prediction of physical protein–protein interactions," *Physical biology*, vol. 2, p. S1, 2005.

[4] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[5] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *Proc. of the 2007 7th IEEE ICDM*. Washington, D.C., USA: IEEE Computer Society, 2007, pp. 322–331.

[6] J. O'Madadhain, J. Hutchins, and P. Smyth, "Prediction and ranking algorithms for event-based network data," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 23–30, 2005.

[7] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *KDD '10: Proc. of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2010, pp. 243–252.

[8] M. P. Stumpf, C. Wiuf, and R. M. May, "Subnets of scale-free networks are not scale-free: sampling properties of networks," *Proc. of the Nat Acad. of Sci.*, vol. 102, no. 12, pp. 4221–4224, 2005.

[9] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *Workshop on Link Discovery: Issues, Approaches and Apps.*, 2005.

[10] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. of the 19th international conference on World wide web*. ACM, 2010, pp. 641–650.

[11] T. Murata and S. Moriyasu, "Link prediction of social networks based on weighted proximity measures," in *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 2007, pp. 85–88.

[12] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *CIKM '03: Proc. of the Twelfth International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2003, pp. 556–559.

[13] A. Narayanan, E. Shi, and B. Rubinstein, "Link prediction by de-anonymization: How we won the kaggle social network challenge," *Arxiv preprint arXiv:1102.4374*, 2011.

[14] J. Scripps, P.-N. Tan, F. Chen, and A.-H. Esfahanian, "A matrix alignment approach for link prediction," in *Proc. of the 19th International Conference on Pattern Recognition*, 2008.

[15] S. Scellato, A. Noulas, and C. Mascolo, "Exploring place features in link prediction on location-based social networks," in *KDD '11: Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2011.

[16] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review Letters E*, vol. 64, 2001.

[17] L. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, pp. 211–230, 2001.

[18] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Machine Learning*, vol. 31, pp. 1–38, 2004.

[19] C. Drummond and R. Holte, "Cost curves: An improved method for visualizing classifier performance," in *Machine Learning*, vol. 65, 2006.

[20] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[21] B. Taskar, M. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *NIPS03*, 2003.

[22] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proc. of the 5th ACM/IEEE-CS Joint inproceedings on Digital Libraries*. New York, NY, USA: ACM, 2005, pp. 141–142.

[23] D. Goldberg and F. Roth, "Assessing experimentally derived interactions in a small world," *Proc. of the National Academy of Sciences of the United States of America*, vol. 100, no. 8, p. 4372, 2003.

[24] D. Hand, "Measuring classifier performance: a coherent alternative to the area under the roc curve," *Machine learning*, vol. 77, no. 1, pp. 103–123, 2009.