# New Perspectives and Methods in Link Prediction

Ryan N. Lichtenwalter
Dept. of Computer Science
The University of Notre Dame
Notre Dame, Indiana 46556
rlichten@nd.edu

Jake T. Lussier
Dept. of Computer Science
The University of Notre Dame
Notre Dame, Indiana 46556
jlussier@nd.edu

Nitesh V. Chawla
Dept. of Computer Science
The University of Notre Dame
Notre Dame, Indiana 46556
nchawla@nd.edu

## ABSTRACT

This paper examines important factors for link prediction in networks and provides a general, high-performance framework for the prediction task. Link prediction in sparse networks presents a significant challenge due to the inherent disproportion of links that can form to links that do form. Previous research has typically approached this as an unsupervised problem. While this is not the first work to explore supervised learning, many factors significant in influencing and guiding classification remain unexplored. In this paper, we consider these factors by first motivating the use of a supervised framework through a careful investigation of issues such as network observational period, generality of existing methods, variance reduction, topological causes and degrees of imbalance, and sampling approaches. We also present an effective flow-based predicting algorithm, offer formal bounds on imbalance in sparse network link prediction, and employ an evaluation method appropriate for the observed imbalance. Our careful consideration of the above issues ultimately leads to a completely general framework that outperforms unsupervised link prediction methods by more than 30% AUC.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

## General Terms

Algorithms, Performance, Theory

## Keywords

Link Prediction, Networks, Machine Learning, Class Imbalance

## 1. INTRODUCTION

Link prediction is an important task in network science that offers unique ways whereby the study of networks can benefit researchers and organizations in a variety of fields. Security agencies can more precisely focus their efforts based on probable relationships in malicious networks that have heretofore gone unobserved [10]. In social networks, individuals can efficiently and effectively find companions, assistants, or colleagues [9]. In medicine and biology, link prediction can be used to find relationships and associations that exist, but which might otherwise surface only after arduous and expensive research and study on a huge selection of agents. Finally, researchers can easily adapt link prediction methods to identify links that are surprising given their surrounding network, or which may not belong at all [15]. Put simply, any environment that naturally maps to a network probably has an equally coherent mapping from link prediction in that network back to an important question in the environment.

This broad applicability demands a powerful yet general framework, and we promote supervised learning. Unsupervised methods, which receive the most attention in link prediction literature, are fundamentally unable to cope with dynamics, interdependencies, and other properties in networks. We recognize that this is not the first paper to apply supervised learning to the link prediction problem, but there are important differences versus past work. First, in spite of the excellent intentions of past researchers, they have fallen prey to unique pitfalls endemic to problems with highly imbalanced class distributions. In [2], the holdout test set is undersampled to balance, and the authors of [17] also contribute only a sample of the negative instances to their test set. As researchers familiar with high skew are aware, modifying the data distribution on which testing is performed generates uninterpretable results. The distribution of the resulting testing data no longer presents the same challenges as the real-world distribution, and performance measures in testing no longer reflect the real capabilities and limitations of the model. Additionally, both of these works employ semantic and contextual information that pertains almost exclusively to the bibliographic domain. Finally, these works do not consider the important impact of geodesic distance and the intricacies of class imbalance specific to the task of link prediction.

We demonstrate that decomposition by geodesic distance has important impacts on predictor performance irrespective of the choice of predictor. We also expand the library of unsupervised measures with an intuitive flow-based metric that is over 15% AUC more predictive than baseline methods in certain networks. After illustrating the benefits of supervised learning, we cast link prediction as a problem in class


*KDD'10,* July 25–28, 2010, Washington, DC, USA.

imbalance. The result of these considerations is a framework that improves upon the best baseline unsupervised methods by over 30% AUC in our test networks. Furthermore, the framework is entirely general, operating over any class of network whether it be weighted, unweighted, directed, or undirected. It does not require any node attributes but is capable of accepting them.

In Section 2 we describe the data sources and evaluation measures. Section 3 explains standard unsupervised approaches and defines a new metric, `PropFlow`. Section 4 lays out the rationale for supervised learning. This leads to a discussion of class imbalance in Section 5. In Section 6 we describe our realization of the framework and Section 7 presents results. Finally, Section 8 provides recommendations and concludes.

# 2. DATA AND EVALUATION

In order to make a compelling, novel case for a supervised framework, we offer a comprehensive explanation of the nature of link prediction, primarily through an examination of two real-world data sets. We also report the prediction results relative to an appropriate metric for predicting in imbalanced environments. It is therefore necessary to first present the two data sources and the principal measure of performance that we employ.

## 2.1 Network Data Sources

The first data source is a stream of 712 million cellular phone calls from a major non-American cellular phone service provider. We construct weighted, directed networks from the calls by creating a node $v_i$ for each caller and a weighted, directed link $e_{ij}$ from $v_i$ to $v_j$ if and only if $v_i$ calls $v_j$. Weights correspond to the number of calls over the link. We shall henceforth refer to this network as `phone`. For all experiments except those in section 4.1 we use the first 5 weeks of data (5.5M nodes, 19.7M links) for extracting features and the sixth week (4.4M nodes, 8.5M links) for obtaining ground truth.

The second data source is a stream of 19,464 multi-agent events representing condensed matter physics collaborations from 1995 to 2000. We construct weighted, undirected networks from the collaborations by creating a node for each author in the event and a weighted, undirected link connecting each pair of authors. Weights correspond to the number of collaborations two authors share. We shall henceforth refer to this network as `condmat`. For all experiments involving `condmat`, we use the years 1995 to 1999 (13.9K nodes, 80.6K links) for extracting features and the year 2000 (8.5K nodes, 41.0K links) for obtaining ground truth.

The networks exhibit different quantitative characteristics. Table 1 contains some summary network statistics in order to provide context for the two networks. These statistics result from the complete 6 weeks of data for `phone` and the complete 1995-2000 network for `condmat`. The assortativity coefficient measures the tendency to find highly connected nodes that are connected to each other. The average clustering coefficient measures the tendency of nodes in the network to be connected in dense groups. Strongly-connected components (SCCs), or connected components in the undirected network, are clusters of vertices in the network in which every vertex in the cluster has a path to all other vertices in the cluster. The size and diameter of such components provides some insight into the broad topological structure of the network.

**Table 1: Network Characteristics**

| | `phone` | `condmat` |
|---|---|---|
| Assortativity Coef. | 0.293 | 0.177 |
| Average Clustering Coef. | 0.187 | 0.642 |
| Mean Degree | 3.88 | 6.42 |
| Median Degree | 3 | 4 |
| Number of SCCs | 1,023,044 | 652 |
| Largest SCC | 4,293,751 | 15,081 |
| Largest SCC Diameter | 25 | 19 |

## 2.2 Evaluation

Scalar measures often used in link prediction, such as precision on the top-$N$ predictions and factors of improvement in precision over random models, rely upon the application of an arbitrary and often unjustified threshold. Most of our evaluation relies instead upon receiver operating characteristic (ROC) curves. These curves present achievable true positive rates ($TP$) with respect to all false positive rates ($FP$) by varying the decision threshold on probability estimations or scores. ROC curves provide information about the operating range of classifiers. For example, classifier $A$ may outperform classifier $B$ when we dictate $FP < 20\%$ but $B$ may outperform $A$ when we allow $FP \geq 20\%$. The expected performance of a random classifier is the line $y = x$, and curves below this line indicate an inverted predictor. Finally, we can say that classifier $A$ dominates classifier $B$ in ROC space if all points on the convex hull of $A$ dominate all points on the convex hull of $B$ in the $xy$-plane, and this is a condition known to correlate highly with superiority in many other measures [13]. The area under the ROC curve (AUC) is a related scalar measure of the performance over all thresholds. AUC has classically been used as a measure of performance in imbalanced learning.

# 3. UNSUPERVISED METHODS

Most existing studies in link prediction consider baseline unsupervised methods to assign scores to potential links. The state-of-the-art in these methods is aggregated and compared in [11], and in Section 3.1 we offer a brief explanation of the particular methods that we study. Moreover, since our goal is to derive a robust feature set, we introduce a novel, effective method in Section 3.2.

## 3.1 Baseline Predictors

Most unsupervised methods either generate scores based on node neighborhoods or path information. The *common neighbors* predictor is the number of neighbors, or out-degree neighbors in our directed network, that are shared by nodes $v_i$ and $v_j$. *Jaccard's coefficient* simply divides the number of common neighbors by the number of total neighbors. The *Adamic/Adar* measure [1] weights the importance of a common neighbor $v_k$ by the rarity of relationships between other nodes and $v_k$. Finally, the *preferential attachment* link prediction score [3, 12] is the product of the degrees of $v_i$ and $v_j$. When we observed especially poor performance for this predictor in `phone`, we tried using in-degree, out-degree, and their sum but observed only minor differences. We report our results based on out-degree performance. From the path-based methods we employ the unweighted *Katz*

**Algorithm 1** `PropFlow` Predictor

**Require:** network $G = (V, E)$, node $v_s$, max length $l$
**Ensure:** score $S_{sd}$ for all $n \leq l$-degree neighbors $v_d$ of $v_s$

insert $v_s$ into $Found$
push $v_s$ onto $NewSearch$
insert $(v_s, 1)$ into $S$
**for** $CurrentDegree \leftarrow 0$ to $l$ **do**
  $OldSearch \leftarrow NewSearch$
  empty $NewSearch$
  **while** $OldSearch$ is not empty **do**
    pop $v_i$ from $OldSearch$
    find $NodeInput$ using $v_i$ in $S$
    $SumOutput \leftarrow 0$
    **for** each $v_j$ in neighbors of $v_i$ **do**
      add weight of $e_{ij}$ to $SumOutput$
    **end for**
    $Flow \leftarrow 0$
    **for** each $v_j$ in neighbors of $v_i$ **do**
      $w_{ij} \leftarrow$ weight of $e_{ij}$
      $Flow \leftarrow NodeInput \times \frac{w_{ij}}{SumOutput}$
      insert or sum $(v_j, Flow)$ into $S$
      **if** $v_j$ is not in $Found$ **then**
        insert $v_j$ into $Found$
        push $v_j$ onto $NewSearch$
      **end if**
    **end for**
  **end while**
**end for**

measure [8], which had better, more stable performance in the networks than the weighted variant. This method contributes each path to a sum with an influence damped in exponential proportion to its length, $l$, using the parameter $\beta$. We select $\beta = 0.005$ and, for performance reasons, we restrict our examination of the measure such that $l \leq 5$.

## 3.2 The `PropFlow` Method

We introduce a new unsupervised prediction method on networks, `PropFlow`, which corresponds to the probability that a restricted random walk starting at $v_i$ ends at $v_j$ in $l$ steps or fewer using link weights as transition probabilities. The restrictions are that the walk terminates upon reaching $v_j$ or upon revisiting any node including $v_i$. The walk selects links based on their weights. This produces a score $s_{ij}$ that can serve as an estimation of the likelihood of new links. `PropFlow` is somewhat similar to *Rooted PageRank*, but it is a more localized measure of propagation, and is insensitive to topological noise far from the source node. Unlike Rooted PageRank, the computation of `PropFlow` does not require walk restarts or convergence but simply employs a modified breadth-first search restricted to height $l$. It is thus much faster to compute. It may be used on weighted, unweighted, directed, or undirected networks. We supply the detailed procedure for weighted, directed networks in Algorithm 1.

In the `phone` network, `PropFlow` outperforms baseline unsupervised methods by $> 15\%$ AUC on average. It outperforms Rooted PageRank by more than 8.75% AUC. We attribute this success to the nature of the mechanisms in `phone` underlying the appearance of new links. Although it may be used in any network, `PropFlow` has special intuitive significance as a link predictor in networks where some resource such as information flows, propagates, or cascades. In transportation networks, when a resource frequently travels from one node through neighbors to another, there is often some cost for the intermediaries. When the expected cost inherent in traveling through intermediaries overcomes the cost of establishing a new link, one can expect formation of that particular link. In transmission networks, the measure represents the link-weighted probability that a randomly outward-propagated transmission sent by one node will reach another. In `condmat`, there is no strong analogy and `PropFlow` is not as effective. Later in the paper, we will explore the utility of `PropFlow` both as an individual predictor and as a feature in our supervised classification framework.

# 4. A CASE FOR SUPERVISED LEARNING

While past studies on link prediction have focused on unsupervised single metrics, some recent works have used a supervised classification scheme, and rightly so. If one accepts the basic premise that ground truth, whether a link forms or not, is always available from prior incarnations of the network, there is no practical disadvantage to using a supervised framework. Even training classifiers based on a single unsupervised method has the potential to outperform rankings generated by sorting the scores of the method if there are multiple differentiating boundaries in the domain of scores. Supervised algorithms are also able to capture important interdependency relationships between topological properties. While past studies simply acknowledged this fact and trained classifiers, we probe more deeply into the relevant issues so that we can fully understand *how* to frame the prediction problem and *why* a supervised framework is best for the task. We first address the *how* question in Section 4.1 by examining how to best transform network data into standard data sets. We then address the *why* question in sections 4.2, 4.3, and throughout section 5. More specifically, Section 4.2 explains that supervised approaches are adaptive and may be more general whereas unsupervised methods are invariant. Section 4.3 demonstrates that unsupervised methods cannot be, or at least have not been, combined into ensembles to reduce variance. Section 5 explains that unsupervised methods are inherently incapable of combating extreme class imbalance, a natural characteristic of link prediction in nearly any network.

## 4.1 Constructing Data Sets

Some networks may always be observable, such as WWW, the Internet, and electricity grids. Others are observable only through events that indicate the presence of links. In the former, one need only select a moment at which to observe the structure directly. In the latter, one must collect events to construct an approximation of the underlying structure. Regardless, the network evolves through time to present a longitudinal source of data. We see then that link prediction, a domain in which unsupervised topological measures receive much attention, is very often suitable for supervised learning. The acquisition of ground truth for constructing models does not mitigate the necessity of the task; future forms of the static network will raise the same questions that exist in the present.

In a typical supervised learning task, we are given a unified set of data with each instance of the form $(\vec{x}, y)$. To convert networks such as `phone` or `condmat` into this format, we have

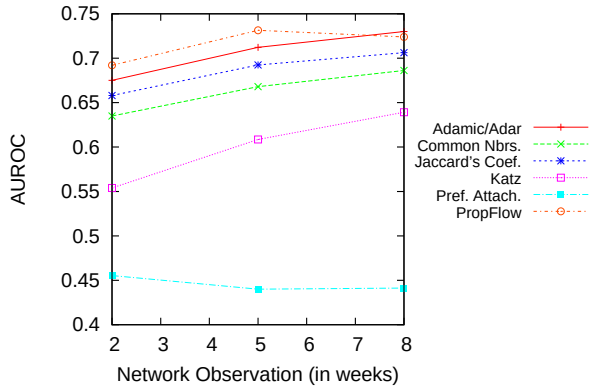



**Figure 1: Performance in the second-degree neighborhood as a function of $\tau_x$.**

to select two values $\tau_x$ and $\tau_y$. These values correspond to the lengths of two adjacent periods over which we want to record events to construct networks. From the first network, $G_x = (V_x, E_x)$, constructed from $t_0$ to $t_{0+\tau_x}$, we extract topological measures, and potentially node attributes, that serve as features for each pair of nodes $(v_i, v_j)$. From the second network, $G_y = (V_y, E_y)$, constructed from $t_{\tau_x+1}$ to $t_{\tau_x+\tau_y}$, we examine $(v_i, v_j)$ to discover whether $e_{ij}$ exists and determine the class label. This yields a data set in the standard $(\vec{x}, y)$ format with $|V_x|^2 - |E_x|$ instances.

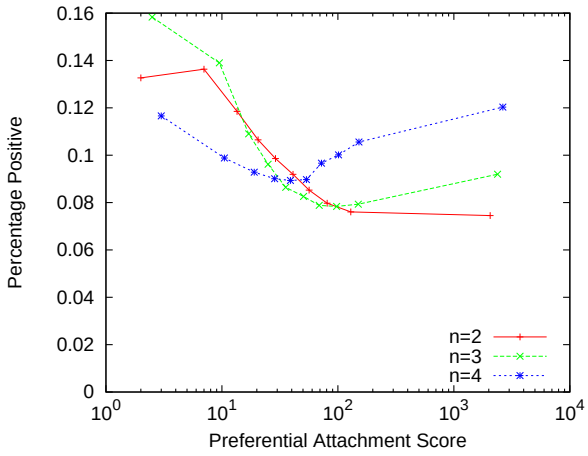
The two parameters $\tau_x$ and $\tau_y$ have important but predictable influences on the success of models. We can expect that increasing $\tau_x$ will increase the quality of topological measures as the network reaches saturation. This is the point at which $\tau_x$ is large enough that the observed events form a topology that closely reflects the underlying static network. As $\tau_x$ approaches this point, the topological measures converge to their actual unobservable static network values, thus allowing improved individual predictive capacity. We can expect that increasing $\tau_y$ will increase the number of positives. We investigate $\tau_x$ on the `phone` network in Figure 1.

Increasing $\tau_x$ has the expected result. The strength of the predictors increases greatly from $\tau_x = 2$ weeks to $\tau_x = 5$ weeks and again from $\tau_x = 5$ weeks to $\tau_x = 8$ weeks. Although measures of network saturation and convergence are outside the scope of this paper, we can remark that they are highly correlated with performance. Since we observe an increase in the predictive power of unsupervised methods, we can expect increases in supervised classification performance too. In effect, the features more closely reflect actual relationships underlying observable events, so models are more closely related to reality.

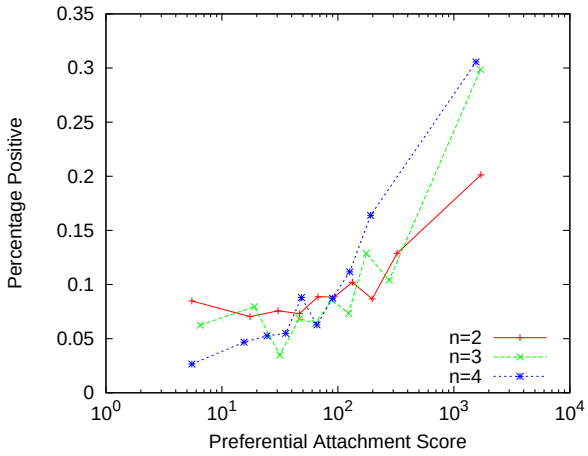
Although this suggests that the results in Section 7 could be even higher with $\tau_x = 8$ weeks, we choose to present the rest of the paper based on $\tau_x = 5$ weeks and $\tau_y = 1$ week. This observational period corresponds to a network that is only partially approaching saturation and might more realistically represent the data available for training in real-world environments.

## 4.2 Generality

While classifiers can generalize well to many environments in the sense that they can adjust models depending on poste-




(a) `phone`




(b) `condmat`

**Figure 2: Preferential attachment performance by scoring region.**

rior information, unsupervised methods are domain-specific. The figures in [11] show that predictors that serve well in one network do not necessarily serve well in all networks; our observations concur. It is clear throughout our results that the performance of the unsupervised methods is unstable not only from one network to the other, but from one graph-distance to another. The preferential attachment predictor is a particularly clear example in Figure 2. The figure shows the percentage of a given score that is positive. Intuitively, the model serves as a good predictor when low scores produce low percentages and high scores produce high percentages and an inverted predictor when the opposite is true.

In the `phone` network, we see that the predictions are inverted, with a higher percentage of positives falling into low scores than high scores. In the `condmat` network, the predictions are much better, with the highest scores corresponding to much higher incidences of links. Finally, for both networks, we observe a similar trend with increasing geodesic distance $n$. The greater the distance, the better preferential attachment models the appearance of links. That is, we see lower percentages for lower scores and higher percentages

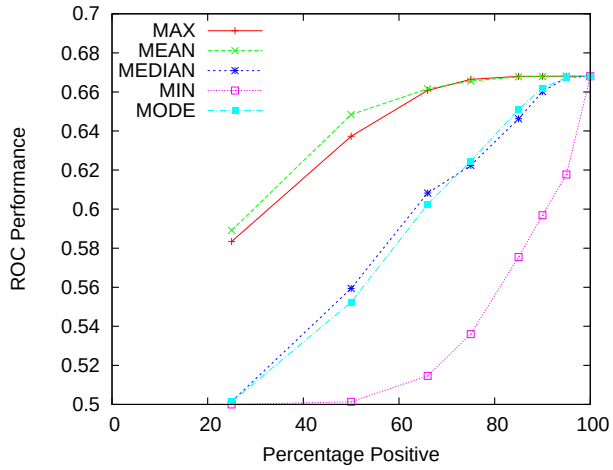



**Figure 3: AUC performance variation in common neighbors 'ensemble' created by randomly selecting $p$ percent of edges for removal 10 times, obtaining the measure, and combining it using the series statistic.**

for higher scores as we move from $n = 2$ to $n = 4$. This supports the intuition that preferential attachment is better as a global indicator where underlying local mechanisms such as neighbor recommendations are weaker.

## 4.3 Variance Reduction and Sampling Issues

Yet another benefit of supervised learning is that classification algorithms, especially unstable algorithms like decision trees, can benefit from reduced variance by placing them in an ensemble framework. Ensembles consist of many models that have been trained on slightly perturbed variations of the data. It is difficult or impossible to accomplish the same goal with unsupervised methods common in link prediction because the score is invariant for a given potential link. Furthermore, it is likely that network analogs to common ensemble sampling techniques are fundamentally flawed as a rough corollary of work in [16], where samples of networks with ill-behaving distributions produce new networks with different properties. Nevertheless, we wanted to explore the potential for one method of ensemble construction using unsupervised methods. To achieve the values in Figure 3, we construct 10 new networks, randomly selecting $p$ percent of the edges in the original network for each. Then, we compute a common neighbors score for the pair $(v_i, v_j)$ in each network and combine the scores using a summary statistic.

The figure shows that the attempt at constructing an ensemble out of an unsupervised method fails. The best AUC appears at 100%, where the network is unsampled and there is no ensemble, which suggests that sampling the network to construct the ensemble does nothing but remove important information, a result we find unsurprising. What the figure does not show is that the $p = 100$ ROC curve dominates all ROC curves for $p < 100$, including mean and max, and that transformations of the ROC curves into precision-recall space show $p = 100$ greatly outperforms even $p = 95$. We performed these experiments only for the common neighbors classifier, but expect the same results for other unsupervised methods.

Supervised classification, on the other hand, offers many strong options for reducing variance such as bagging [4] and random forests for decision trees [5], the latter of which also increases classification efficiency. While a single classifier that incorporates several of the unsupervised methods can greatly improve classification versus those methods, variance reduction techniques can further improve it.

# 5. GRAPH DISTANCE AND IMBALANCE

A significant novelty of link prediction as a supervised learning problem is its extreme imbalance, which reaches past the most skewed distributions studied by the imbalance community. While unsupervised methods cannot combat this imbalance because they are agnostic to class distributions by definition, supervised learning schemes are able to balance data and focus on class boundaries. In this section, we will study some of the properties of that imbalance, especially as it relates to graph distance.

## 5.1 Sparse Networks

We proceed by constructing a formal proof of the lower bound on the class imbalance ratio for link prediction in sparse networks. The proof operates on two reasonable, almost ubiquitously satisfied assumptions. First, the network maintains the property of sparseness throughout the period of interest. Second, the network growth is limited such that the number of nodes may only double during the period of interest, although the theorem holds for any factor of growth $g$ such that $g \ll |V|$.

*Definition 1.* Let a network $G = (V, E)$ be described as *sparse* if it maintains the property $|E| = k|V|$ for some constant $k \ll |V|$.

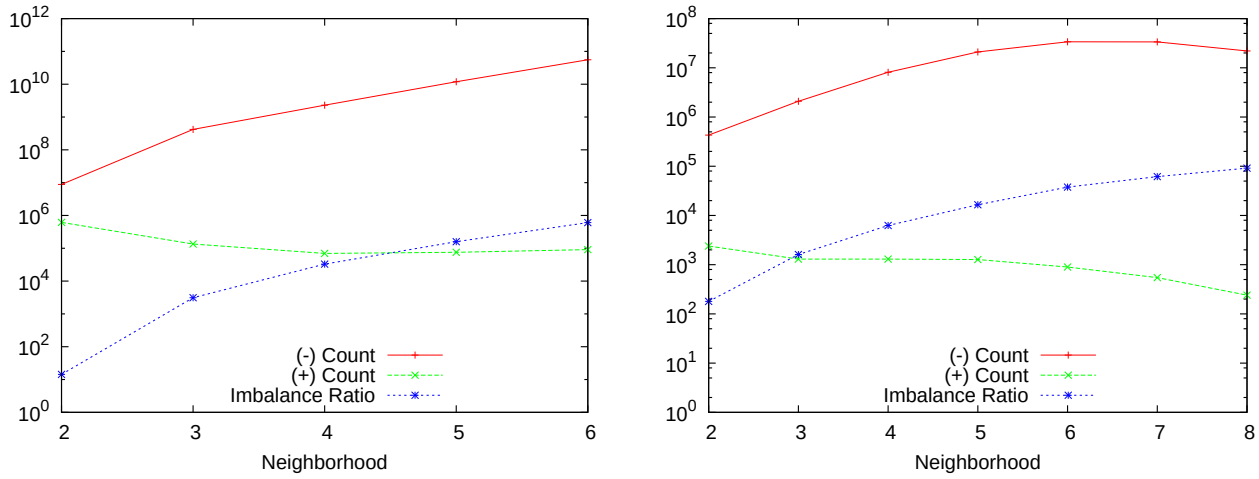
THEOREM 1. *The class imbalance ratio for link prediction in a sparse network $G$ is $\Omega\left(\frac{|V|}{1}\right)$ when at most $|V|$ nodes may join the network.*

PROOF. The number of possible links in $G$ is $|V|^2$. Then the number of missing links, $|E^C|$, is $|V|^2 - k|V| \in \Theta(|V|^2)$. Let $|V'|$ nodes and $|E'|$ links join the network. Since $|V| + |V'| \leq 2|V| \in \Theta(|V|)$, $|E| + |E'| \in \Theta(|V|)$, which requires that $|E'| \in O(|V|)$. The number of positives is $|E'|$, and there are $\left|(E \cup E')^C\right| \in \Theta(|V|^2)$ negatives. This gives us $\frac{\Theta(|V|^2)}{O(|V|)}$, equivalent to $\Omega\left(\frac{|V|}{1}\right)$, as the class ratio. □

Thus the imbalance issue in the general link imbalance problem becomes clear. No matter how many links we hope to anticipate, $TP$, we must accept a baseline random model that produces $FP$ such that $FP \propto TP \times |V|$. Even a model thousands of times better than random performs poorly. The severity of the problem is exacerbated by the fact that positives often represent occurrences of greater interest.

## 5.2 Graph Distance and Neighborhoods

In link prediction, graph distance plays a primary role in determining the imbalance ratio. We define the $n$-degree neighborhood of a node $v_i$ as the set of nodes exactly $n$ hops away from $v_i$. As $n$ increases, the number of potential links will increase in proportion to the superlinear increase




(a) `phone` (b) `condmat`

**Figure 4: Neighborhood imbalance properties.**

in the number of neighbors. Simultaneously, it is reasonable to expect that the new links will tend to form between nodes that are close, such as in `phone` where local influences such as recommendations and common neighbors pertain. Figure 4 illustrates the imbalance behavior of the `phone` and `condmat` networks. It also demonstrates the distribution of distances between pairs of nodes for all distances where the underlying computation is feasible. It is important to note here that `phone` has a diameter in its largest strongly connected component of 25 while `condmat` has a diameter of only 19 in its largest connected component. Further, the $n \leq 6$-degree neighborhood of any given node in `phone` still includes only a moderate fraction of nodes in the network. In the much smaller `condmat` network, the $n \leq 6$-degree neighborhood of any given node often approaches the periphery and includes almost every node in the network.

The simultaneous severe increase in unformed potential links and severe decrease in links that actually form causes even more dramatic increases in imbalance ratios. `phone` imbalance goes from 131:1 at $n = 2$ to 32,880:1 at $n = 4$ and 606,926:1 at $n = 6$. `condmat` imbalance goes from 179:1 at $n = 2$ to 6,247:1 at $n = 4$. Fortunately, there is often little reason to believe that the benefit of successfully predicting links to nodes at high $n$ is greater than the benefit of predicting them at low $n$.

Given that imbalance increases so sharply between neighborhoods, and local mechanisms quickly give way to global mechanisms at higher values of $n$, we suggest that each neighborhood should be treated as a separate problem in supervised learning. This also allows us to avoid the $V$:1 imbalance of the general problem. Additionally, in the case of the entire class of neighbor-based models, there is null output for $n \geq 3$ in undirected networks because there is no sense in which such nodes can have common neighbors. Many unsupervised methods have implicit or explicit adjustments for graph distance, but the fundamental distinction of neighborhood comes for free. Supervised models may benefit from a decrease in noise and, for networks in which the distance spanned by the predicted link is inconsequential, the consideration of low $n$ saves computational time.

# 6. CLASSIFICATION

With the nature of the problem and the advantages of supervised learning carefully considered, we now present the details of the high-performance link prediction (HPLP) framework. In most cases, we reserved two-thirds of the labeled data for training the model and the remaining third for testing, but for the presentation of significance results, we employed 10-fold cross-validation with care to use unmodified folds for testing. At no time do we change the class distribution in any testing data. Due to computational complexity, we restricted our consideration of classifiers to the `WEKA` [18] C4.5 [14] equivalent, J48 (parameters -A -U), naïve Bayes (default parameters), and `WEKA` bagging (10 bags, default parameters) with random forests (10 trees, default parameters). The last is easy to parallelize.

## 6.1 General Feature Extraction

Any network, no matter its type or source, necessarily supports basic topological measures such as $v_i$ and $v_j$ in-degree and out-degree, $v_i$ and $v_j$ in-volume and out-volume, or their undirected equivalents in the case of `condmat`. We also employ the baseline unsupervised models from Section 3, including `PropFlow`, and path-oriented measures such as the number of shortest paths from $v_i$ to $v_j$ and the maximum flow that can travel from $v_i$ to $v_j$ within 5 steps. Though we use only these features for generality, we could use any other available features, including measures of reciprocity, or node attributes such as age and gender.

To illustrate that we are able not only to achieve performance that vastly exceeds baseline methods, but that we do so without using them as features, we include both a restricted feature set that does not use the existing unsupervised methods (HPLP) and the full feature set (HPLP+). Table 2 contains details of the features.

## 6.2 Ensemble of Classifiers

We capitalize on the ability of supervised frameworks to reduce variance, as described in Section 4.3, by using ensembles of classifiers. We use two different ensemble methods: bagging and random forests. Random forests is an excel-

**Table 2: Feature Listing**

| Name | Parameters | HPLP | HPLP+ |
|---|---|---|---|
| In-Degree($i$) | - | ✓ | ✓ |
| In-Volume($i$) | - | ✓ | ✓ |
| In-Degree($j$) | - | ✓ | ✓ |
| In-Volume($j$) | - | ✓ | ✓ |
| Out-Degree($i$) | - | ✓ | ✓ |
| Out-Volume($i$) | - | ✓ | ✓ |
| Out-Degree($j$) | - | ✓ | ✓ |
| Out-Volume($j$) | - | ✓ | ✓ |
| Common Nbrs($i$,$j$) | - | ✓ | ✓ |
| Max. Flow($i$,$j$) | $l = 5$ | ✓ | ✓ |
| Shortest Paths($i$,$j$) | $l = 5$ | ✓ | ✓ |
| PropFlow($i$,$j$) | $l = 5$ | ✓ | ✓ |
| Adamic/Adar($i$,$j$) | - | | ✓ |
| Jaccard's Coef($i$,$j$) | - | | ✓ |
| Katz($i$,$j$) | $l = 5$, $\beta = 0.005$ | | ✓ |
| Pref Attach($i$,$j$) | - | | ✓ |

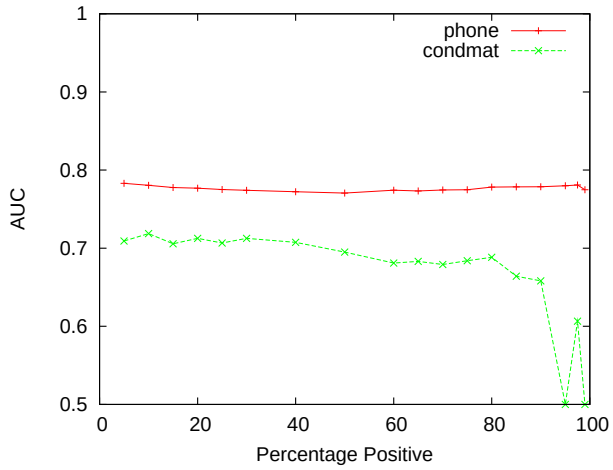



**Figure 5: Performance reaction of a single C4.5 decision tree to different undersampling levels. The x-axis is in terms of the percentage of all training examples that are positive.**

lent method for these data sets for two reasons. First, the data sets are composed of a combination of strong features and weak features. While the weak features are occasionally helpful, random forests is an excellent method to prevent overfitting them. Second, the decreased training time for each single tree counters the increased training time to build the forest, making it an especially efficient method of variance reduction for these large data sets.

In both data sets, after undersampling to balance, we found on average a 4.04% AUC improvement moving from a single tree to 10 bagged trees and an additional 3.91% improvement moving from 10 bagged trees to 10 bagged random forests. The total average improvement of 8.11% justifies the use of these variance reduction methods, and the sheer size of the testing sets lends significance to even fractional percentage improvements. We found that neither 100 bagged trees without random forests nor 100 random forest trees without bagging offered the same improvements. Finally, we note that the improvements we observed after the application of these techniques are impressive, but still suboptimal. We constructed our ensembles from the same selection of undersampled negative class instances. With a minimal penalty in computational time, each member of the ensemble could make use of a random selection of the entire set of negative class instances.

## 6.3 Overcoming Imbalance

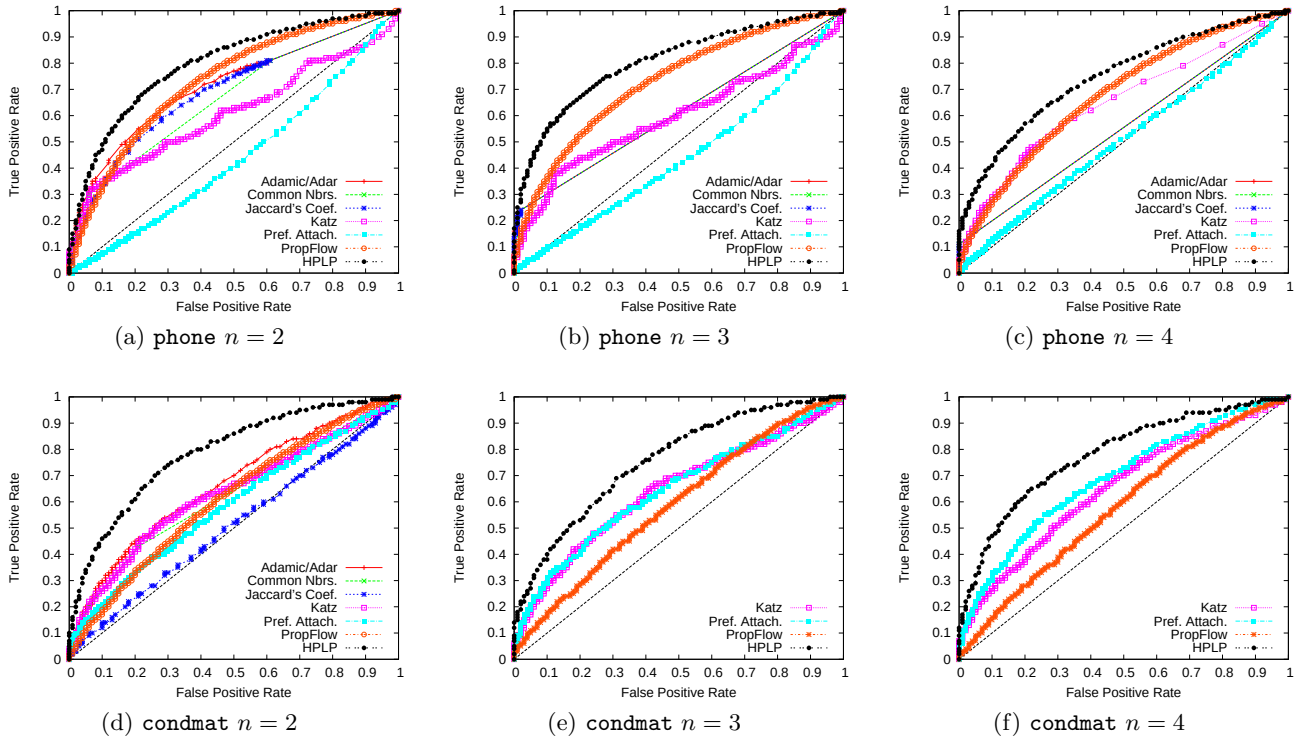
Aside from reducing variance, we considered different approaches to overcome the imbalance described in Section 5. In doing so, we had to carefully consider the enormity of the data sets, especially for large values of $n$. As Figure 4 shows, in `phone` $n = 2$ produces 81.4 million instances and $n = 4$ produces 2.2 billion instances. Even in the smaller `condmat` network, $n = 2$ produces 431.6 thousand instances and $n = 4$ produces 8.1 million instances.

One of the best oversampling strategies, SMOTE [6], is $O(p^2 \cdot |\vec{x}|)$, the product of the number of positive class instances $p$ and the length of the feature vector. While this may work for `condmat` where $p$ is on the order of thousands, it certainly will not work for `phone`. We also theorize that `phone`, with $p$ in the order of tens or hundreds of thousands, does not suffer as much from lack of definition in the positive class as from a strong classifier bias toward $f(x) = 0$ from prior information. Furthermore, oversampling approaches only increase data set size and training time. We considered training skew-insensitive decision trees based on Hellinger distance [7]. Such trees are best when trained on the original training set distribution, however, and performed poorly with undersampled data. Without undersampling, the training set sizes for the data often render training with these trees infeasible. Undersampling, on the other hand, can help to mitigate the problem of class imbalance while also reducing the size of the training set.

In section 5.2 we argue for treating each neighborhood as a separate problem. This also allows for skew-combating methods that are appropriate to the particular neighborhood. If $n \geq 2$ is combined into a single data set and subsequently uniformly undersampled, negative representatives of $n = 2$ will be underrepresented causing a distortion of the real $n = 2$ class boundary.

The class ratio to which the data set is undersampled serves as a significant parameter to our framework. In Figure 5 we explore a wide range of possible sampling parameters using a single C4.5 decision tree evaluated according to AUC. The performance of the `phone` data set is relatively stable, but it exhibits an interesting trend wherein AUC drops slightly when the class distribution is balanced. `condmat` achieves AUC values that are higher with increasing negative class representation through the ratios we tested. Despite these results, in Section 7 we undersample the training sets to balance to present a consistent view of performance.

We would like to mention that the beneficial relationship between link prediction researchers and class imbalance researchers is mutual. Class imbalance research contributes many options to the link prediction community. Simultaneously, link prediction offers the potential for a wide variety of data sets that match or surpass the imbalance ratios of the most demanding publicly available data. Any large network can become an authentic source for data with selectable features, selectable imbalance ratios depending on the chosen value of $n$, and a large pool of positive instances from which to draw the desired positive class cardinality.




(a) `phone` $n = 2$ (b) `phone` $n = 3$ (c) `phone` $n = 4$

(d) `condmat` $n = 2$ (e) `condmat` $n = 3$ (f) `condmat` $n = 4$

**Figure 6: The ROC curves for `phone` (top) and `condmat` (bottom).**

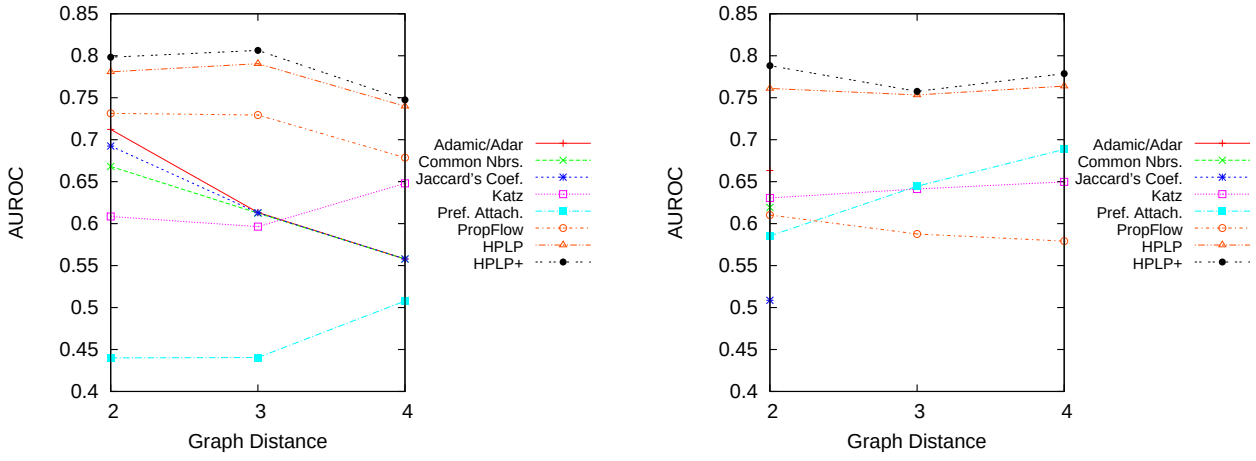
## 7. DETAILED ANALYSIS

To achieve the following results, we trained bagged random forests, which exhibited universally superior performance. For uniformity of reporting, all training sets are undersampled to balance rather than to a level optimized for each network and $n$. Because AUC alone can sometimes be misleading, we also include ROC curves. Figure 6 contains curves describing the performance of unsupervised methods and the supervised framework.

The `phone` and `condmat` curves illustrate that the mechanism by which links arise is indeed different both across networks and geodesic distances. In fact, this leads to an interesting broad observation about mechanisms of link formation. In the `condmat` network, individuals have a global view of the topology through a variety of means. In essence, researchers know of other eminent researchers in the field however remote they may be in terms of geodesic distance. In the `phone` network, there is little reason to suspect that individuals have much knowledge of other individuals at remote locations in the network. The performance of the preferential attachment method supports this theory in the two networks; it is much stronger for `condmat` than for `phone`. Additionally, it shows performance that increases with $n$ in both networks. The more distance potential links span in a network, the weaker local influences such as neighbor recommendations or path-based considerations become. The discriminative power of methods based on these principles generally drops accordingly. On the other hand, global influences such as degree have the same interpretation at any distance in the network. As the local influences lose their meaning with increasing $n$, the preferential attachment method becomes an increasingly pure estimation of link formation biases.

We can clearly see the deterioration in predictive capacity of the local methods. Neighbor-based methods perform worse for $n = 3$ than for $n = 2$ `phone`, and they perform much worse for $n = 4$ than for $n = 3$. Neighbor-based methods have no meaning for $n \geq 3$ in directed networks such as `condmat`; there is no sense in which two unconnected individuals greater than two hops away from each other can share a neighbor. The `PropFlow` predictor degrades more gracefully than neighbor-based methods in `phone` but suffers mediocre performance on `condmat`. Despite much greater curve areas in `phone`, `PropFlow` does not dominate other measures. Instead, methods based on common neighbors achieve slightly higher $TP$ rates at very low $FP$ rates, but `PropFlow` rapidly surpasses them. Importantly, the HPLP dominates all other methods in ROC space in every case except `phone` $n = 2$, where `PropFlow` actually crosses at $FP = 0.99$. On a more general note, especially in `phone` where imbalance ratios grow higher, the increasing difficulty of more distant neighborhoods is exhibited in the form of ROC curves that converge toward $TP = FP$.

We now move to the discussion of AUC values and Figure 7. HPLP achieves performance levels as much as 30% higher than the best unsupervised methods. The difference in performance between HPLP and HPLP+ averages $< 1\%$ AUC. Though it does not appear in the figures, in `phone` we also created a data set with all unsupervised methods except `PropFlow`. We found that `PropFlow` alone achieves higher performance than using all other unsupervised methods put together when using the same basic supporting features, such as node degree. To substantiate the hypothe-




(a) `phone` (b) `condmat`

**Figure 7: The AUC values for `phone` and `condmat`.**

sis that there are useful dependencies between features, we compared naïve Bayes to a single C4.5 tree and confirmed that the latter wins by $> 2.3\%$. Nonetheless, even using a single, fast naïve Bayes classifier, HPLP always greatly outperforms the strongest of the unsupervised methods. To provide statistical significance to this statement, we used two-tailed paired t-tests. The paired samples come from 10-fold cross validation AUC scores. For all values of $n$, HPLP outperforms unsupervised methods at over 99% confidence and HPLP+ outperforms them at over 99.99% confidence.

## 8. CONCLUSION

The general framework we propose in this paper achieves major improvements over existing methods. Although it outperforms such methods by $> 30\%$ in terms of AUC, it does not require any domain-specific node attributes to do so. It can be applied in any domain exactly as described or it can accept any number of domain-specific features. It is also highly scalable; feature computation for a single path-based method requires more time than the entire classification framework. The feature computation itself is embarrassingly parallel.

In addition to the results, the supporting study allows for some recommendations. Unsurprisingly, for networks where topological convergence and saturation may be a concern, the training observation period, $\tau_x$, should be as long as possible. The parameter $\tau_y$ for the static network from which labels are gathered should match the size of the real-world prediction window so that testing and real-world prior distributions are as similar as possible. In link prediction on networks such as the Internet or electricity grids, these concerns are moot since snapshots contain the entire network structure.

Optimal class ratios for undersampling are specific to the problem at hand, but the results we obtained for both networks indicate that undersampling to balance may not be ideal in the link prediction domain. Those employing this classification framework should be aware of this fact and should investigate other ratios as resources permit. For small networks where computational resources are not problematic, we advise the use of skew-insensitive classifiers such as Hellinger trees in an ensemble framework. We encourage the community to consider the link prediction task as a separate problem for each desired neighborhood in domains where local mechanisms are likely to pertain. This not only decreases computational time by considering those links most likely to rank highly regardless but has the potential to sensitize supervised classification to the specific mechanisms and boundaries present for predictions within the target graph distances.

In general, the application of unsupervised methods, at least without due study and consideration, is highly suboptimal. No such method, no matter how high its performance in some subset of our data, provides acceptable performance for the entire range of problems. Where there is some limitation on supervised learning due to the unavailability of labels for training, we hope that the included study of unsupervised performance measures proves helpful in the selection of an appropriate option. For networks where one expects local mechanisms to dominate, especially local mechanisms related to flow or propagation, we highly encourage the use of the unsupervised method, `PropFlow`, proposed in this paper.

We have published all scripts and source code for prediction and evaluation at `http://www.nd.edu/~rlichten/linkpred` along with the `condmat` data set. We regret that we are unable to make the `phone` data set available due to non-disclosure agreements.

## 9. ACKNOWLEDGMENTS

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.

We sincerely thank Mark E.J. Newman for the `condmat` data set and Albert-László Barabási for the `phone` data set. Finally, we thank our colleagues in the University of Notre Dame Interdiscplinary Center for Network Science and Applications (iCeNSA).

## 10. REFERENCES

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[2] 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.

[3] A.-L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaboration. *Physica A*, 311(3-4):590–614, 2002.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and P. W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of A.I. Research*, 16:341–378, 2002.

[7] D. A. Cieslak and N. V. Chawla. Learning decision trees for unbalanced data. In *Proc. of the ECML*. Springer, 2008.

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

[9] H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63, 1997.

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

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

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

[13] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[15] M. J. Rattigan and D. Jensen. The case for anomalous link discovery. *SIGKDD Explorations Newsletter*, 7(2):41–47, 2005.

[16] 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.*, 102(12):4221–4224, 2005.

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

[18] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, California, USA, second edition, 2005.