Node Classification with Bounded Error Rates[*]

Pivithuru Wijegunawardana[1], Ralucca Gera[2], and Sucheta Soundarajan[1]

[1] Syracuse University, Department of Electrical Engineering & Computer Science,
{ppwijegu,susounda}@syr.edu

[2] Nhttps://www.overleaf.com/projectaval Postgraduate School, Department of Applied Mathematics, Monterey, CA
RGera@nps.edu

Abstract. Node classification algorithms are widely used for the task of node label prediction in partially labeled graph data. In many problems, a user may wish to associate a confidence level with a prediction such that the error in the prediction is guaranteed. We propose adopting the Conformal Prediction framework [17] to obtain guaranteed error bounds in node classification problem. We show how this framework can be applied to 1) obtain predictions with guaranteed error bounds, and 2) improve the accuracy of the prediction algorithms. Our experimental results show that the Conformal Prediction framework can provide up to a 30% improvement in node classification algorithm accuracy while maintaining guaranteed error bounds on predictions.

Keywords: Node classification · Conformal prediction · Bounded error rates.

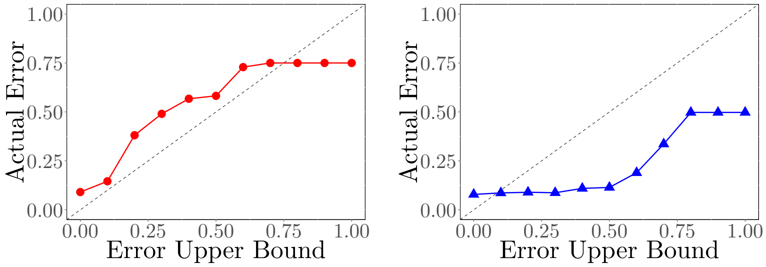
1 Introduction

In real world network analysis problems, it is common for data to be incomplete. In such cases, node classification algorithms play an important role: given a partially labeled graph, these algorithms predict labels for unlabeled nodes by using known node's labels and connections between nodes. For example, consider a criminal group hidden inside a general social network. If some criminals and non-criminals are identified, can an algorithm predict whether the unlabeled nodes are criminals? By taking advantage of connections, algorithms specifically designed for node classification generally perform better on semi-supervised graph classification tasks as compared to traditional classification algorithms [11, 19].

The user of a node classification algorithm often may wish to associate a confidence with each prediction. For example, when predicting whether a node in a social network is a criminal or not, a prediction may lead to a criminal investigation. In such applications, it is thus essential to have prediction algorithms that can provide guaranteed error rates on unseen data.

The performance of a classification algorithm is generally measured with metrics such as accuracy, precision, and recall. Such metrics describe the algorithm performance in aggregate, but do not measure certainty of individual predictions.

[*] This material is based upon work supported by the U.S. Army Research Office under grant number W911NF1810047.



(a) Class Probability (b) Conformal Prediction

Fig. 1: Comparison of using Conformal Prediction framework vs class probability as upper error bound for Cora citation dataset classification using Iterative Classification Algorithm. Conformal prediction actual error is always no greater than the error bound where as the actual error for class probability does not follow the error bound.

Node classification algorithms can generally output a vector indicating the probability that a node belongs to each class. One can consider the probability of a node belonging to some class as the confidence of the label (1-probability would be the upper error bound). Figure 1a shows that actual errors are larger than the upper error bounds when using the label probability as error bound. Therefore, label probability should not be interpreted as confidence values

In this work, we demonstrate how the *Conformal Prediction* framework can be used to obtain error bounds for the node classification task. Conformal Prediction (CP) is a framework to provide guaranteed error bounds for prediction algorithms [16]. This framework works on top of a prediction algorithm (e.g. SVM, Neural Network) and for a specified error bound, considers how unusual a data instance is in consideration to training data. Since CP framework is a very mathematical framework, application to prediction algorithms require customizing the framework according to the algorithm. The CP framework has been applied to provide guaranteed error bounds for machine learning algorithms [4, 7, 12]. However, to our knowledge, CP has not been applied to the network setting. Figure 1b shows an example application of the CP framework to node classification problem. The prediction error rate is always lower then the expected error the CP framework suggests.

Our contributions in this paper are: 1) We show conditions under which node classification problem would satisfy the CP framework assumptions to obtain valid error bounds, 2) We show how to apply the CP framework to node classification algorithms from different categories 3) We conduct an experimental analysis over various types of node attributes and graphs and show that the CP framework can improve node classification algorithm accuracy.

2 Background

2.1 Node Classification Algorithms

Node classification algorithms consider both node attributes and node connectivity patterns when making predictions. There are three main categories of node classification algorithms. The first category contains local classifier based algorithms, where a local classifier is iteratively trained using node attributes and network information to predict labels for unlabeled nodes, such as logistic regression local classifier. Iterative Classification Algorithm (ICA) [11] and Link Based Classification algorithm [6] are examples of such algorithms. These algorithms iteratively predict labels for the unlabeled nodes in the graph using predicted labels in the previous round of predictions.

The second category of algorithms are label propagation based algorithms, where the algorithms use random walks to learn a global labeling function across the network [19]. These algorithms predict labels for nodes in the graph by considering hitting probability of each label in a random walk.

The third, and newest, category of node classification algorithms learn a deep representation of the network and labeling function. There are two approaches to learning this representation. The first approach uses network embedding-based algorithms, which generate feature vectors for nodes in a graph in an unsupervised manner. These algorithms use multiple random walks starting at each node, and trains a prediction model based on these features [2, 13]. The second approach is learning a labeling function using deep neural networks based on graph representation. Graph Convolutional Neural Networks (Graph CNN) are widely used to conduct node classification under this category [5].

In the current work, we consider node classification algorithms from each category mentioned above and show how the `CP` framework can be applied to obtain predictions with guaranteed error bounds.

2.2 Conformal Prediction Framework

The `CP` framework outputs a set of predictions for a given sample with a bounded error rate by comparing "how typical" the sample is as contrasted to other samples [16]. Suppose that we are given a data set $Z = \{z_1, z_2, \ldots, z_n\}$ where $z_i = (x_i, y_i)$; $x_i \in \mathbb{R}^d$ is the feature vector of the sample i, and $y_i \in Y$ is the class label for i^{th} sample. Here, Y is the set of class labels, i.e. $Y = \{y^1, y^2, \ldots, y^\ell\}$.

Given a new sample with feature vector x_{n+1}, the `CP` framework measures how typical the following sequence is: $(z_1, z_2, \ldots, z_n, (x_{n+1}, y^k))$, where $y^k \in Y$. Since we already know the labels for $z_1, z_2, \ldots, z_n$, we are in effect measuring how typical the sequence is when label y^k is assigned to the new sample, and how likely is that $n+1$'s true label is y^k [12].

The `CP` framework uses a test for randomness to measure how likely a sequence is, where the $p_{cp}-$value for a given sequence is calculated using Equation (1). A given conformity measure calculates the "typicalness" of a data instance (α values). The α_i is the conformity score for i^{th} data instance [12].[3]

$$p_{cp}(z_1, z_2, \ldots, (x_{n+1}, y^k)) = \frac{|\{i = 1, \ldots, n : \alpha_i \leq \alpha_{n+1}\}|}{n} \tag{1}$$

$p-$values vs. $p_{cp}-$values: We adopt notation $p_{cp}-$value instead of $p-$value in this paper to avoid any confusion since a higher $p_{cp}-$value in `CP` framework

[3] Note that the "$\leq$" sign in Equation 1 changes to "$\geq$" if we are using a non-conformity function instead of conformity.

means the label in consideration is highly likely. Conversely, a higher $p-$value in general means that there is stronger evidence towards the alternative hypothesis.

Given some significance value ϵ, `CP` framework first calculates $p_{cp}-$values for all sequences considering all possible class labels; then the prediction set of $n+1$ sample at ϵ significance is calculated using Equation (2).

$$P(n+1,\epsilon) = \{y^k : y^k \in Y \quad \& \quad p_{cp}(z_1, z_2, \dots, (x_{n+1}, y^k)) > \epsilon\} \tag{2}$$

For example, consider we are predicting hobbies in a social network. A node can have one of the hobbies among the following; {reading, singing, dancing, cooking}. When we use the `CP` framework, for an unlabeled node v, we observe that the $p_{cp}-$values for each label are $\{0.2, 0.1, 0.01, 0.02\}$. If we set significance to 0.05, both reading and singing will be predicted as hobbies of node v since both these labels have $p_{cp}-$values higher than the significance level. This also shows that the chance of generating sequences including dancing and cooking as labels is less than 5%, implying that these labels are highly unlikely.

Note that in Equation (2), the `CP` framework outputs the set of labels that satisfy the specified significance rather than a single prediction. Therefore, `CP` framework predictions can have one prediction, multiple predictions, or zero predictions, in case none of the labels satisfy the significance requirement. The probability of not including true labels in the prediction set is less than the specified threshold, providing an error rate bounded by the significance level. If we are to predict labels at significance ϵ, the probability of not including the correct label in the prediction set is ϵ and the confidence in the prediction is $1-\epsilon$.

The `CP` framework provides guaranteed error bounds for predictions under the assumption that the data is exchangeable, meaning any permutation of the sequence $(z_1, z_2, \dots, z_n, (x_{n+1}, y^k))$ should result in the same $p_{cp}-$value. This assumption is necessary to obtain the $p_{cp}-$value using Equation (1).

The `CP` framework was originally introduced in the *transductive* setting, where the true label of the current sample is revealed before the arrival of the next sample [16]. In this setting, the given model is trained considering each possible label for new data instance and the framework measures how typical the model is. Since this setting requires training the model for each new data instance and each possible label, applying this in a real world setting would be very inefficient.

The *Inductive Conformal Prediction* (`ICP`) is an alternative approach which splits the training data into actual training set and a calibration set, and uses the calibration set to conduct `CP` [12]. The `ICP` framework uses the training set to train the underlying prediction model, and the calibration set to calculate the $p_{cp}-$value. In the `ICP` setting, we only consider the calibration set when calculating the $p_{cp}-$value of Equation (1).

2.3 Related Work

Bayesian Framework, Probably Approximately Correct Learning theory (PAC theory) [3] and generalization error bounds [9] are other frameworks that provide bounded error rates in machine learning applications. Bayesian Framework error rates are dependent on the priors that are used in the estimation. Hence, the error bounds are not guaranteed in case priors are wrong. PAC theory and generalization error bounds provides upper bounds on the trained model rather than individual samples. The only assumption that the CP framework makes is that the data is exchangeable, which is valid for most machine learning data. Dashevskiy et al. [1] show that even in cases where exchangeability assumption is violated (e.g., time series data), the CP framework still provides reasonable error bounds. The CP framework, unlike PAC theory and generalization error bounds, can provide error bounds for individual samples rather than the algorithm.

Initial work on the CP framework was primarily theoretical, and focused on proving the error bounds. Applying the CP framework to machine learning algorithms required defining conformity measures specific to algorithms, showing that the data is in fact exchangeable. Research in this area shows how the CP framework can be applied to various algorithms including decision trees [4], neural networks [12], and SVM [7] etc.

To best our knowledge, this is the first work that considers providing guaranteed error bounds for node classification algorithms, and shows how the CP framework can be applied to obtain those error bounds

3 Methodology

We now introduce the details of how the ICP framework can be applied to node classification algorithms. In the $z_n = (x_n, y_n)$ a node classification problem, we have that $x_n \in \mathbb{R}^d$ is the d-dimensional feature vector for node n, and y_n is the label of node n.

To show that the ICP framework applies, we must demonstrate that the data is exchangeable. Note that this does *not* require that the data is i.i.d., simply that all permutations of each sequence of training samples are equally likely. Since we are drawing training samples uniformly at random from the set of nodes, exchangeability holds. We also considered sampling training data using a network crawling algorithm such as random walk or snowball sampling. Resulting error bounds are not valid in these cases since any training node ordering is not equally likely (not exchangeable) for random walk or snowball sampling.

The conformity function is an integral part of the ICP framework, measuring how different the data instance in consideration from the calibration set. Any real valued conformity function that measures how different a sample is can be used to produce valid nested prediction regions [17], but the efficiency (smaller prediction regions) of the algorithm depends on how well the nonconformity function measures differences between data instances. For example, an efficient prediction according to our hobby prediction example in Section 2 would be predicting one hobby as the label. An inefficient prediction would have no hobby or more than one hobby in the prediction set.

Consider a prediction algorithm that outputs a vector $\sigma_n \in \mathbb{R}^{|Y|}$ for some unlabeled node n, indicating the probability that n would belong to each class in Y. One possible conformity measure for such an algorithm is the probability margin, which is the difference between the label in consideration and the highest probability of any other label [14]. We can calculate the probability margin conformity score for some label $y^k \in Y$ using Equation 3.

$$C(n, y^k) = \sigma_n(y^k) - \max_{y^i \in Y : y^i \neq y^k} (\sigma_n(y^i)). \tag{3}$$

Given a node classification algorithm M, a graph G, set L of labeled nodes, set U of unlabeled nodes, a significance level ϵ, and a conformity function C, we introduce Algorithm 1 to show how the ICP framework can be applied to node classification problem.

Algorithm 1 ICP for Node Classification

```
Input: $G$= Graph, $L$ = $labeled\_nodes$, $U$ = $unlabeled\_nodes$, $M$ = prediction algorithm, $C$= Conformity function, $\epsilon$ = significance
Output: Prediction set for each node in $U$ at significance $\epsilon$
procedure ICP
    Divide $L$ into $T = training\_set$ and $S = calibration\_set$
    Train $M$ using $G$ and $T$                         ▷ Train prediction model $M$
    for $s \in S$ do
        $\sigma_s = M(s)$                               ▷ Get prediction probability vector $\sigma_s$ for $s$
        $\alpha_s = C(\sigma_s, y_s)$                   ▷ Calculate conformity score for $s$ and $s$'s label $y_s$
    for $u \in U$ do
        $P_u = \{\}$                                    ▷ $u$'s prediction set at significance $\epsilon$
        for $y^k \in Y$ do
            $\sigma_u = M(u)$
            $\alpha_u = C(\sigma_u, y^k)$
            $p = \frac{|\{s \in S : \alpha_s \leq \alpha_u\}|}{|S|}$   ▷ Calculate p-value for label $y^k$
            if $p > \epsilon$ then
                $P_u.add(y^k)$                          ▷ Add $y^k$ to $u$'s prediction set
```

4 Experiments

We conduct experiments to evaluate whether the ICP framework predictions meet the specified error bounds. We consider node classification algorithms from different categories: Iterative Classification Algorithm (ICA), Label Propagation (LP), Graph Convolutional Network (GCN), and Deepwalk (DW). We now introduce the performance metrics and data sets used in our research.

4.1 Performance Metrics

Our evaluation closely follows the evaluation criteria in [4]. We use several measures to evaluate the quality of predictions made by ICP framework for the node classification problem:

1. We check whether the ICP framework predictions meet the specified maximum error bounds. Since node classification graph data meets the exchangeability assumption, the specified error bounds should be met.

Table 1: Network dataset statistics

Dataset	Type	Nodes	Edges	Label	Classes	Label Assortavity
Cora	Citation	2708	5278	Research area	7	0.771
PubMed	Citation	19717	44327	Research area	3	0.686
Blogcatalog	Social	10312	333983	Blogger group	39	0.05
Facebook100	Social	2235	90954	Year	9	0.409
PPI	Biological	3890	38739	Biological state	50	0.05

2. We evaluate the `ICP` framework predictions based on their efficiency. Since the ICP framework outputs a set of predictions for a node based on its conformity score, an efficient prediction would have only a single class in the prediction set. We consider the fraction of predictions with only one class (OneC), multiple classes (MultiC) and zero classes (ZeroC) to evaluate the efficiency of the ICP framework.
3. We compare the accuracy of the baseline prediction model (BaselineAcc) with the accuracy of one class predictions (OneAcc) from the ICP framework to show that the ICP framework enhances performance of the baseline prediction model.

4.2 Datasets

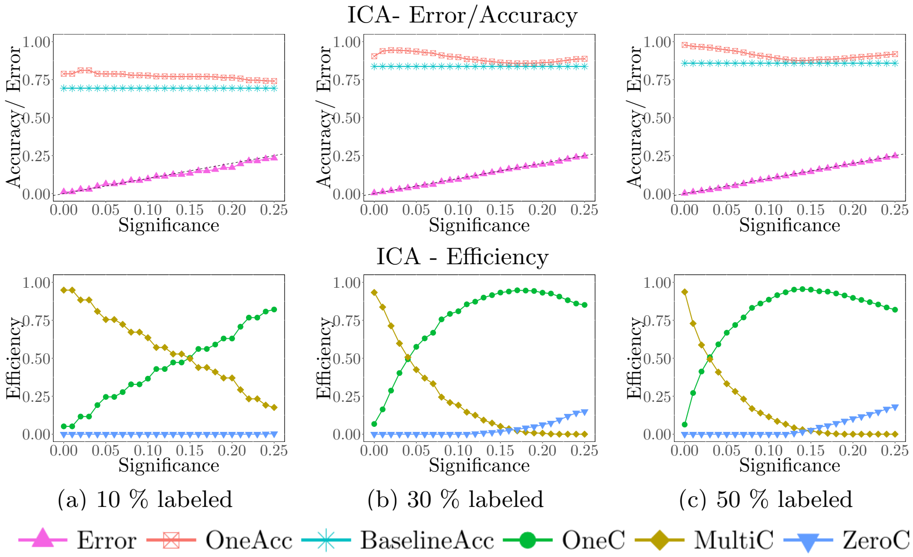
Node classification algorithms generally perform well on assortative networks, but less well on nodes are not assortative. Accordingly, we have selected graph datasets with varying levels of assortativity. For each network, we use the largest connected component. Cora [8,15] and PubMed [10] are citation networks, showing citation relationships between papers. Facebook100 [4] is the Amhrest college Facebook friendship network. BlogCatalog [18] is a blogger friendship network. The Protein-Protein Interaction network [2] is a subgraph of the PPI Homo Sapiens network. Networks are described in Table 1.

4.3 Experimental Setup

We run experiments as a multi-class prediction problem where we vary the percentage of labeled nodes in the network from 10% up to 50%. We randomly sample the labeled data from each class proportional to the size of the class and report average performance over 10 runs. We used 25% of the training data as the calibration set to conduct conformal prediction.

For ICA and Deepwalk, we use a multi-class logistic regression classifier as the base classifier. We set Deepwalk hyper parameters for all data sets as follows: 80 walks, 128 dimension representation, window size 10, and walk length 40 according to [13]. GCN hyper parameters are set at 0.5 dropout rate, 5.10^{-4} $L2$ regularization and 16 hidden units, according to [5].

[4] Obtained from https://archive.org/download/oxford-2005-facebook-matrix.



(a) 10 % labeled (b) 30 % labeled (c) 50 % labeled

Fig. 2: Accuracy and efficiency for Cora citation data set using ICA as the baseline algorithm when 10%, 30% and 50% of the nodes are labeled. The actual error is always no greater than the specified significance level.

5 Results

Figure 2 shows results of ICP using ICA on the Cora citation network with $10\%, 30\%$ and 50% of the nodes labeled, using the performance metrics discussed in Section 4.1. First, we see that the actual errors in all algorithms are very close to the given significance level, demonstrating that the ICP framework in fact provides accurate error bounds for node classification algorithms.

Second, as expected, the percentages of OneC (one-class predictions) and MultiC (multiple-class predictions) increase and decrease as we increase the significance level, respectively. Recall our hobby prediction example in Section 2 where $p_{cp}-$values of node v for labels; reading, singing, dancing and cooking are $\{0.2, 0.1, 0.01, 0.02\}$ respectively. If we set significance to 0.01, all label $p_{cp}-$values will satisfy the significance requirement and hence will be included in the prediction set. Therefore, we can observe many multiple-class predictions at lower significance values. When we increase significance level to, e.g., 0.15, only one label satisfies the significance requirement, increasing the number of one-class predictions. ZeroC (zero class predictions) slightly increases at higher significance values causing OneC to reduce slightly, because if we set significance to 0.2, none of the labels meet significance.

Finally, we see that the accuracy of the ICP framework is higher than the accuracy of the baseline node classification algorithm, showing that the predictions from ICP are more reliable than those from the baseline prediction algorithm. Results are consistent across different algorithms and labeled node percentages.

Tables 2 and 3 summarize the performance of the ICP framework applied to ICA and Label Propagation, respectively. Both algorithms closely maintain

Table 2: Conformal prediction framework performance using ICA as the baseline algorithm. Average performance over 10 runs where randomly selected 30% of the nodes are labeled in each run.

Dataset	Significance	Error	OneC	MultiC	ZeroC	OneAcc	BaseAcc
Fb100 Amherst	0.05	**0.045 ± 0.01**	0.58	0.42	0	0.94	0.82
	0.15	**0.133 ± 0.02**	0.88	0.12	0	0.87	
	0.25	**0.25 ± 0.02**	0.85	0	0.15	0.88	
BlogCatalog	0.05	**0.05 ± 0.01**	0.05	0.95	0	0.52	0.23
	0.15	**0.15 ± 0.01**	0.11	0.89	0	0.48	
	0.25	**0.25 ± 0.01**	0.15	0.85	0	0.44	
PubMed	0.05	**0.046 ± 0.01**	0.52	0.48	0	0.92	0.83
	0.15	0.154 ± 0.01	0.97	0.03	0	0.85	
	0.25	**0.257 ± 0.01**	0.84	0	0.16	0.89	
PPI	0.05	**0.051 ± 0.01**	0.03	0.97	0	0.21	0.10
	0.15	0.147 ± 0.01	0.08	0.92	0	0.18	
	0.25	**0.248 ± 0.02**	0.14	0.86	0	0.17	

Table 3: Conformal prediction framework performance using Label Propagation as the baseline algorithm. Average performance over 10 runs where randomly selected 30% of the nodes are labeled in each run.

Dataset	Significance	Error	OneC	MultiC	ZeroC	OneAcc	BaseAcc
Cora	0.05	**0.043 ± 0.01**	0.60	0.40	0	0.94	0.83
	0.15	**0.141 ± 0.03**	0.89	0.09	0.01	0.87	
	0.25	0.252 ± 0.04	0.85	0	0.15	0.89	
PubMed	0.05	0.052 ± 0.01	0.54	0.46	0	0.91	0.82
	0.15	**0.149 ± 0.01**	0.92	0.08	0	0.85	
	0.25	0.253 ± 0.01	0.87	0	0.13	0.86	
Fb100 Amherst	0.05	0.052 ± 0.01	0.38	0.62	0	0.93	0.78
	0.15	**0.144 ± 0.01**	0.71	0.29	0	0.88	
	0.25	0.252 ± 0.03	0.92	0.01	0.07	0.81	
BlogCatalog	0.05	**0.049 ± 0.01**	0.04	0.96	0	0.36	0.22
	0.15	**0.149 ± 0.01**	0.10	0.90	0	0.38	
	0.25	0.253 ± 0.01	0.15	0.85	0	0.39	
PPI	0.05	**0.048 ± 0.01**	0.04	0.96	0	0.12	0.10
	0.15	**0.146 ± 0.01**	0.10	0.90	0	0.11	
	0.25	**0.239 ± 0.03**	0.15	0.85	0	0.11	

the given error bounds. ICP framework can cause the prediction errors to be slightly higher than the given error bound since the predictions are based on the calibration set rather than the whole training set.

Further, applying ICP improves baseline accuracy of both algorithms in all data sets. The ICP improves ICA accuracy in FB100 data from 0.82 to 0.94, while predicting singleton labels for 58% of the nodes with a guaranteed error rate of 5%. In the Label Propagation algorithm, ICP improves accuracy for Facebook100 data from 0.78 to 0.93, while predicting singleton labels for 38% of the nodes with a guaranteed error rate of 5%. When the baseline predictor accuracy is reasonable, ICP provides efficient predictions (more singleton pre-

Table 4: Conformal prediction framework performance using GCN as the baseline algorithm. Average performance over 10 runs where randomly selected 30% of the nodes are labeled in each run.

Dataset	Significance	Error	OneC	MultiC	ZeroC	OneAcc	BaseAcc
Cora	0.05	**0.045 ± 0.01**	0.70	0.30	0	0.95	0.83
	0.15	**0.141 ± 0.02**	0.93	0.07	0	0.86	
	0.25	**0.248 ± 0.03**	0.83	0	0.17	0.91	
PubMed	0.05	**0.047 ± 0.01**	0.72	0.28	0	0.94	0.85
	0.15	0.151 ± 0.01	0.98	0.01	0.01	0.86	
	0.25	**0.249 ± 0.01**	0.82	0	0.18	0.92	
Fb100 Amherst	0.05	**0.048 ± 0.01**	0.43	0.57	0	0.96	0.72
	0.15	**0.139 ± 0.02**	0.59	0.41	0	0.9	
	0.25	0.251 ± 0.02	0.90	0.10	0	0.77	
BlogCatalog	0.05	**0.049 ± 0.01**	0.001	0.999	0.05	0	0.12
	0.15	**0.139 ± 0.01**	0.005	0.995	0	0.05	
	0.25	0.238 ± 0.01	0.01	0.99	0	0.11	
PPI	0.05	**0.049 ± 0.01**	0.0002	0.9998	0	0.03	0.05
	0.15	**0.143 ± 0.02**	0.002	0.998	0	0.18	
	0.25	0.239 ± 0.02	0.005	0.995	0	0.11	

dictions). When the baseline predictor does not perform well, conformity scores also become less meaningful leading ICP to make more multiple predictions. In blogcatalog, at significance level 0.15, only 11% of the predictions are singletons.

Tables 4 and 5 summarize results for GCN and DeepWalk respectively. GCN algorithm works well when node labels show homophily (Cora, FB100 and PubMed). In the Blogcatalog and PPI data sets, GCN algorithm baseline accuracy is 0.12 and 0.05, making it impractical to get meaningful predictions. The Deepwalk algorithm only considers network structure when predicting labels. If node labels are not correlated with the structure, even if data shows high homophily, Deepwalk baseline accuracy is low. In general, both these algorithms maintain the error bounds but provide inefficient predictions in some cases.

5.1 Perturbation Analysis

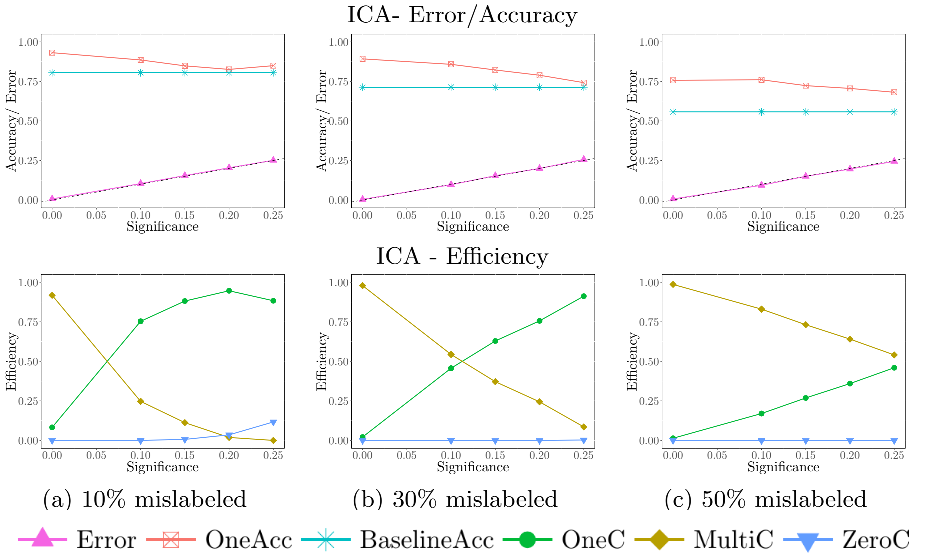
Real world network data collection can be prone to errors. In Figure 3, we show the effect of mislabeled data on ICP framework predictions. We consider the CORA data set with 30% of the nodes initially labeled and change labels randomly for $10\%, 30\%$ and 50% of the nodes in the training data. Figure 3 shows that mislabeled training data does not affect ICP error bounds. As we increase the percentage of mislabeled data, the efficiency of predictions decreases, since the percentage of singleton predictions decreases.

6 Discussion and Conclusion

In this work we consider the problem of providing guaranteed error bounds for predictions in node classification algorithms. We use the CP framework, which works with a given prediction model to provide bounded error rates. We use ICP a more efficient variant of the CP framework and show how this can be applied to

Table 5: Conformal prediction framework performance using DeepWalk as the baseline algorithm. Average performance over 10 runs where randomly selected 30% of the nodes are labeled in each run.

Dataset	Significance	Error	OneC	MultiC	ZeroC	OneAcc	BaseAcc
Cora	0.05	**0.048** ± **0.01**	0.59	0.41	0	0.94	0.82
	0.15	**0.150** ± **0.02**	0.90	0.09	0.01	0.86	
	0.25	**0.239** ± **0.03**	0.88	0	0.12	0.87	
PubMed	0.05	**0.048** ± **0.01**	0.53	0.47	0	0.92	0.80
	0.15	**0.149** ± **0.01**	0.89	0.11	0	0.84	
	0.25	**0.245** ± **0.01**	0.90	0	0.10	0.84	
Fb100 Amherst	0.05	**0.047** ± **0.02**	0.001	0.999	0	0.14	0.15
	0.15	0.155 ± 0.02	0.005	0.995	0	0.12	
	0.25	0.268 ± 0.04	0.01	0.99	0	0.13	
BlogCatalog	0.05	0.057 ± 0.01	0.012	0.988	0	0.82	0.28
	0.15	0.153 ± 0.01	0.02	0.98	0	0.79	
	0.25	0.255 ± 0.01	0.03	0.97	0	0.76	
PPI	0.05	0.051 ± 0.01	0.0002	0.9998	0	0.43	0.11
	0.15	0.151 ± 0.02	0.005	0.995	0	0.32	
	0.25	**0.250** ± **0.02**	0.007	0.993	0	0.31	



(a) 10% mislabeled (b) 30% mislabeled (c) 50% mislabeled

Fig. 3: Performance of `ICP` applied to CORA citation data with 30% of nodes initially labeled. ICA is used as the baseline. Note that 10%, 30% and 50% of training data mislabeled. `ICP` maintains the error bounds even when 50% of the training data is mislabeled. But the efficiency decrease as there are more errors.

ICA, Label Propagation, GCN and DeepWalk algorithms to improve prediction accuracy and provide more reliable predictions. We evaluate performance of this framework using citation, social and biological networks and show that 1) Specified significance levels are maintained across all data sets and Algorithms, and 2) ICP can in fact improve accuracy of baseline algorithms. We conduct a perturbation analysis to show that ICP framework error bounds are not affected by the perturbations, rather the efficiency is affected.

References

1. Dashevskiy, M., Luo, Z.: Time series prediction with performance guarantee. IET communications **5**(8), 1044–1051 (2011)
2. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
3. Haussler, D.: Probably approximately correct learning. University of California, Santa Cruz, Computer Research Laboratory (1990)
4. Johansson, U., Boström, H., Löfström, T.: Conformal prediction using decision trees. In: 2013 IEEE 13th international conference on data mining (2013)
5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
6. Lu, Q., Getoor, L.: Link-based classification. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03). pp. 496–503 (2003)
7. Makili, L., Vega, J., Dormido-Canto, S., Pastor, I., Murari, A.: Computationally efficient svm multi-class image recognition with confidence measures. Fusion Engineering and Design **86**(6-8), 1213–1216 (2011)
8. Motl, J., Schulte, O.: The ctu prague relational learning repository. arXiv preprint arXiv:1511.03086 (2015)
9. Nadeau, C., Bengio, Y.: Inference for the generalization error. In: Advances in neural information processing systems. pp. 307–313 (2000)
10. Namata, G., London, B., Getoor, L., Huang, B., EDU, U.: Query-driven active surveying for collective classification. In: 10th International Workshop on Mining and Learning with Graphs. p. 8 (2012)
11. Neville, J., Jensen, D.: Iterative classification in relational data. In: AAAI-2000 Workshop on Learning Statistical Models from Relational Data (2000)
12. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. In: Tools in artificial intelligence. IntechOpen (2008)
13. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
14. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S., et al.: Boosting the margin: A new explanation for the effectiveness of voting methods. The annals of statistics **26**(5), 1651–1686 (1998)
15. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008)
16. Shafer, G., Vovk, V.: A tutorial on conformal prediction. Journal of Machine Learning Research **9**(Mar), 371–421 (2008)
17. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer Science & Business Media (2005)
18. Zafarani, R., Liu, H.: Social computing data repository at ASU (2009), http://socialcomputing.asu.edu
19. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International conference on Machine learning (ICML-03). pp. 912–919 (2003)