Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

OMNI-Prop: Seamless Node Classification on Arbitrary Label Correlation

Yuto Yamaguchi[†], **Christos Faloutsos**[‡] and **Hiroyuki Kitagawa**[†]
†University of Tsukuba; ‡Carnegie Mellon University
yuto_ymgc@kde.cs.tsukuba.ac.jp, christos@cs.cmu.edu, kitagawa@cs.tsukuba.ac.jp

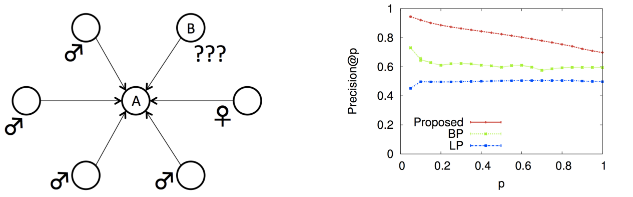
Abstract

If we know most of Smith's friends are from Boston, what can we say about the rest of Smith's friends? In this paper, we focus on the node classification problem on networks, which is one of the most important topics in AI and Web communities. Our proposed algorithm which is referred to as *OMNI-Prop* has the following properties: (a) *seamless* and *accurate*; it works well on any label correlations (i.e., homophily, heterophily, and mixture of them) (b) *fast*; it is efficient and guaranteed to converge on arbitrary graphs (c) *quasi-parameter free*; it has just one well-interpretable parameter with heuristic default value of 1. We also prove the theoretical connections of our algorithm to the semi-supervised learning (SSL) algorithms and to random-walks. Experiments on four real, different network datasets demonstrate the benefits of the proposed algorithm, where OMNI-Prop outperforms the top competitors.

Introduction

If most of Smith's friends are from Boston, what would you say about the rest of Smith's friends? Most people would say that the rest of friends are also from Boston. In this paper, we address the problem of *node classification* on networks where nodes are classified into one of the predefined labels. There are various kinds of labels people may concern: demographic labels of people on social networks such as gender, locations, and education (Mislove et al. 2010); research areas of researchers on co-authorship networks (Sun et al. 2009); and political leanings of blogs on blogosphere (Adamic and Glance 2005). Although these labels are important for many practical applications (e.g., marketing products and forecasting the election outcome), a significant part of labels on networks are typically unavailable, meaning that a lot of practical applications benefit from node classification.

The node classification problem can be formulated as follows: A number of important tasks have been solved in the form of node classification problem: user profiling (Jurgens 2013), blog classification (Ikeda, Takamura, and Okumura 2008), recommendations (Ha et al. 2012), and word sense disambiguation (Pham, Ng, and Lee 2005). For this reason, the node classification on networks, which is also known as the collective classification, has been one of the most active and important research fields in AI and Web communities (Neville and Jensen 2000)(Zhu et al. 2003)(Sen et al. 2008).



(a) The main idea.

(b) The main result on POKEC-G network.

Figure 1: (a) **Hypothesis**: If the most of followers are males, then the rest is also male. (b) **Proposed OMNI-Prop wins**: against top competitors on the network with neither homophily nor heterophily.

There are two major research issues on this problem, which we tackle with in this paper.

Label correlation types. Node classification is often solved based on the *label correlations* known as *homophily* (i.e., love of similar) or *heterophily* (i.e., love of different). On the homophily networks, nodes with similar characteristics tend to connect to each other, while the reverse is true on the heterophily networks. As well as the above two popular cases, there is also the case of mixture of homophily and heterophily. For example, on social networks, talkative people tend to prefer talkative friends but some talkative people prefer silent friends. In contrast to homophily and heterophily, this mixed correlation depends on the preference of each node and therefore no clear correlation is observed at the network level. Most of node classification algorithms (Zhu et al. 2003)(Goldberg, Zhu, and Wright 2007) assume one of these label correlations and propagate signals on the graph. However, as pointed out by (Bilgic and Getoor 2010), propagating signals in the wrong way causes misclassifications. Hence, we need the *seamless* node classification algorithm that can be applied to any kind of label correlations without any assumption about it.

Evidence sufficiency. The majority of nodes have the small number of degrees on most of real-world networks with *power law* degree distributions (Faloutsos, Faloutsos, and

Faloutsos 1999). Unfortunately, it is inherently difficult to classify such nodes because we cannot obtain sufficient evidences from their neighbors. In some practical applications, one's first priority is the accuracy of the classification while the recall is not that important (e.g., a situation that false alarms cause a serious problem). Therefore, we should choose not to classify such nodes to achieve reasonable classifications, which is known as the *reject option*.

Our main ideas. In this paper, we propose a node classification algorithm called *OMNI-Prop*, which tackles with the above mentioned two research issues. Our algorithm is based on the very intuitive and natural hypothesis that *if the most of followers*[1] *have the same label, then the rest also have the same label*. For example, Figure 1(a) illustrates this hypothesis; node B is probably a male because node A has a lot of males as its followers. In this way, OMNI-Prop propagates labels on the graph. Thanks to this idea, OMNI-Prop can handle any types of label correlations because we do not mind the label of node A.

Furthermore, we adopt *Bayesian-like* fashion to consider the evidence sufficiency. OMNI-Prop assigns every node with the *prior belief* about its label and then it is updated using the evidence from its neighbors. After the priors are updated, we can choose to take a reject option or not for each node by considering the shape of the probability distribution.

Figure 1(b) shows our main result, which illustrates that OMNI-Prop achieves substantial improvement against top competitors to infer the genders of users on POKEC-G dataset (see Experiments Section for the detail). The important point is that OMNI-Prop works well although there is no clear label correlation (neither homophily nor heterophily) observed on this network.

Contributions. Our contributions of OMNI-Prop algorithm are summarized as follows:

1. *Seamless* and *Accurate*: OMNI-Prop achieves good accuracy on arbitrary label correlation types.
2. *Fast*: each iteration of the propagation computation of OMNI-Prop is linear on the input size and is proved to converge on any graph structures.
3. *Quasi-Parameter Free*: OMNI-Prop has just one parameter with default value of 1, meaning no parameter tuning is needed.

Also, we offer a set of theorems that state theoretical connections to the *semi-supervised learning* (*SSL*) algorithms and the *random-walk* on the specific type of graph which we call *twin graph*. These connections indicate that our algorithm has solid theoretical foundations.

Furthermore, we performed the extensive experiments using four different labeled networks: a blog citation network, a co-authorship network, and two social networks. The results show that our algorithm outperforms top competitors in terms of classification accuracy on all networks that have different label correlation types.

Table 1: Qualitative comparison

	LP	RWR	BP	OMNI-Prop
Homophily	✓	✓	✓	✓
Heterophily			✓	✓
Mix				✓
No parameters to tune	✓	✓		✓
Convergence	✓	✓	?	✓

To the best of our knowledge, our algorithm is the first solution to handle arbitrary label correlation without any parameters to tune. We believe that our proposal helps practitioners solve the node classification problem without laborious preliminary investigation on label correlations and parameter tuning.

Outline. The rest of the paper is organized as standard: problem definition, method description, theoretical analysis, experimental analysis, and conclusion.

Related work

As major algorithms that solve the node classification problem, we briefly survey Random Walk with Restarts (RWR), Label Propagation (LP), and Belief Propagation (BP).

RWR is based on the *Personalized PageRank* (Jeh and Widom 2003) (Page et al. 1999), where the random walk goes back to seed nodes with a certain probability. RWR has applied to many practical applications, including recommendations (Liu et al. 2010) and link prediction (Liben-Nowell and Kleinberg 2007).

LP is proposed by (Zhu et al. 2003), which is one of the most well-known graph-based SSL algorithms. Graph-based SSL algorithms have been actively studied in the AI community (Zhou et al. 2004)(Baluja et al. 2008)(Gong et al. 2014). Although there are some algorithms that can address the heterophily networks (Goldberg, Zhu, and Wright 2007)(Tong and Jin 2007), no algorithm can be applied to arbitrary networks (including mixture) without parameter tuning. Since LP and RWR have similar characteristics, we mainly compare our algorithm with LP theoretically and experimentally.

BP is originally proposed as an efficient inference algorithm for probabilistic graphical models (Pearl 1982), but it has been applied to various problems like RWR and LP. Although BP is very effective, its recursive calculation has no guarantee to converge on arbitrary graphs.

Table 1 shows the qualitative comparison between our algorithm and these major competitors. BP can handle heterophily, but it requires K^2 parameters to tune where K is the number of different labels. None of these three can handle mixture of homophily and heterophily, which is addressed by our algorithm.

Problem Formulation

This section defines some terminologies and formulates the node classification problem. Let $G = (V, E)$ be a partially labeled graph where V is set of N nodes and E is set of M edges. The set of nodes is composed of two types of components $V = V^L \cup V^N$ where $V^L = \{v_1, \cdots, v_L\}$ is a set of L *labeled nodes* whose labels are known, while

[1] We adopt the term followers for in-neighbors, and followees for out-neighbors.

Table 2: Symbols and Definitions

Symbols	Definitions
A	Adjacency matrix.
N, M, K	# of nodes, edges, and labels.
L	# of labeled nodes.
U	# of unlabeled nodes.
s_{ij}	Self-score that node i has label j.
t_{ij}	Follower-score that node i's followers have label j.
b_j	Prior belief that nodes have label j
λ	Prior strength parameter

$V^N = \{v_{L+1}, \cdots, v_{L+U}\}$ is a set of U *unlabeled noes* whose labels are unavailable. Let $\mathcal{Y}$ be the set of K possible labels, and $Y_L = \{y_1, y_2, \cdots, y_L\}$ be the label assignments for the corresponding nodes in V^L. Using these terminologies, the node classification problem is formulated as follows:

Problem 1 (Node Classification).

- **Given**: a partially labeled graph $G = (V, E)$.
- **Find**: score s_{ij} that unlabeled node i has label j.
- so that the classification function $\mathcal{C}(v_i) = \arg\max_j s_{ij}$ outputs the same label as the ground truth.

Note that a larger value of s_{ij} indicates that node i is more likely to have label j. If we can acquire well calibrated scores, we can make a reasonable decision to take a reject option or not.

Proposed Algorithm

In this section we propose OMNI-Prop, which is based on the following hypothesis:

Hypothesis 1. *If the most of followers of a node have the same label, then the rest also have the same label.*

By adopting *followers* rather than *followees*, OMNI-Prop is robust to the suspicious connecting behaviors (Sun et al. 2005) because it is difficult for fraudulent nodes to control followers of a node. OMNI-Prop formulates this hypothesis by the following two basic ideas:

- *Mutual-induction*: OMNI-Prop considers that each node has two scores: *self-scores* s_{ij} and *follower-scores* t_{ij}. The self-score s_{ij} denotes how likely node i has label j, while the follower-score t_{ij} denotes how likely node i's followers have label j. These two scores are calculated on the mutually-induced fashion.
- *Bayesian-like inference*: OMNI-Prop assigns each node with the prior scores for both self-scores and follower-scores, which are updated by the evidence from its neighbors.

Here we assume we are given a directed graph, but the OMNI-Prop is also applicable to undirected graphs if we consider there are two opposite directed edges between connected nodes. Table 2 gives the list of symbols we use.

Iterative algorithm. The basic scheme of OMNI-Prop is to perform *self-score update* step and *follower-score update* step iteratively. In the self-score update step, s_{ik} of unlabeled nodes are calculated by aggregating follower-scores of their followees as follows:

$$s_{ik} \leftarrow \frac{\sum_{j=1}^{N} A_{ij} t_{jk} + \lambda b_k}{\sum_{j=1}^{N} A_{ij} + \lambda}, \tag{1}$$

where b_k is the prior score and $\lambda \geq 0$ is the parameter. Each labeled node has fixed value as $s_{ik} = \delta(y_i, k)$ that takes 1 if y_i equals to k, 0 otherwise.

In the follower-score update step, each node adaptively adjusts the follower-score by looking around its followers' self-scores as follows:

$$t_{jk} \leftarrow \frac{\sum_{i=1}^{N} A_{ij} s_{ik} + \lambda b_k}{\sum_{i=1}^{N} A_{ij} + \lambda}. \tag{2}$$

Intuitively, if node i has a lot of followers with label k, follower-score t_{ik} gets large and in turn every follower of node i will gets large self-score s_{ik}. This mutually-induced calculation formulates our hypothesis.

Priors and the parameter. It is worth noting that the priors and parameter λ play a crucial role. The parameter controls how much evidence is needed to update the priors, which can be regarded as the *strength of the prior* in analogous to Bayesian inference. Although one can use arbitrary values (e.g., the class mass ratio) for priors, we adopt the uniform prior (i.e., $b_k = 1/K$) in this paper according to our preliminary study, which leads to the good result. Note that we use the same priors and parameter for s_{ij} and t_{ij} because there is no reason to differentiate them. As discussed in our experiments later, we can always use $\lambda = 1.0$, meaning that no parameter tuning is needed.

Matrix form of iterative algorithm. Hereafter, we formulate the matrix form of OMNI-Prop. Let $S = \{s_{ik}\}$ and $T = \{t_{ik}\}$ be row normalized $N \times K$ matrices. We write S and the adjacency matrix A as sub-matrices as follows:

$$S = \begin{pmatrix} S_L \\ S_U \end{pmatrix}, A = \begin{pmatrix} A_L \\ A_U \end{pmatrix}, \tag{3}$$

where S_L and S_U are $L \times K$ and $U \times K$ matrices, and A_L and A_U are $L \times N$ and $U \times N$ matrices, respectively. The subscript L and U mean that the sub-matrices correspond to the labeled nodes in V^L and unlabeled nodes in V^U, respectively. Recall that each labeled node has $s_{ik} = \delta(y_i, k)$, which corresponds to the elements of S_L.

Using these matrices, the self-score and follower-score update steps can be written in matrix form as follows:

$$S_U \leftarrow (D_U + \lambda I)^{-1} \left(A_U T + \lambda \mathbf{1}_U \boldsymbol{b}^T \right), \tag{4}$$

$$T \leftarrow (F + \lambda I)^{-1} \left(A^T S + \lambda \mathbf{1}_N \boldsymbol{b}^T \right), \tag{5}$$

where $\mathbf{1}_U$ and $\mathbf{1}_N$ is U and N dimensional column vectors where each component is 1, and $\boldsymbol{b}^T = (b_1, \cdots, b_K)$. D_U and F are $U \times U$ and $N \times N$ diagonal matrices defined as follows:

$$[D_U]_{ii} = \sum_{j=1}^{N} [A_U]_{ij}, \; F_{ii} = \sum_{j=1}^{N} A_{ji}.$$

Algorithm 1 Iterative Algorithm

```
Require: explicit labels $Y_L$, adjacency matrix $A$, parameter $\lambda$
$\boldsymbol{b} \leftarrow uniform()$
$S^0 \leftarrow initializeS(Y_L)$
$T^0 \leftarrow initializeT()$
$k \leftarrow 0$
repeat
    $S_U^{k+1} \leftarrow (D_U + \lambda I)^{-1} \left( A_U T^k + \lambda \mathbf{1}_U \boldsymbol{b}^T \right)$
    $T^{k+1} \leftarrow (F + \lambda I)^{-1} \left( A^T S^k + \lambda \mathbf{1}_N \boldsymbol{b}^T \right)$
    $k \leftarrow k + 1$
until error between $S_U^{k+1}$ and $S_U^k$ becomes sufficiently small
return $S_U^k$ and $T^k$
```

OMNI-Prop repeats these two steps iteratively until it converges. As stated in the later section, it is proved that OMNI-Prop with the positive parameter always converges. The iterative algorithm of OMNI-Prop is shown in Algorithm 1. s_{ij} of unlabeled nodes and t_{ij} of all nodes are initialized as arbitrary values.

Reject option After we obtain S_U, we can choose not to classify the node v_i if the value of $\max_k s_{ik}$ is small, which means that there is not sufficient evidence to classify v_i. In our experiments, we report the *precision@p* that is the precision for top $p\%$ nodes ordered by $\max_k s_{ik}$.

Theoretical Analysis

In this section, we offer a set of theorems that show the theoretical foundation of OMNI-Prop. All omitted proofs are on the supplementary file.

Complexity Analysis

This section shows the time complexity of OMNI-Prop.

Theorem 1. *The time complexity of OMNI-Prop is $O(hK(N+M))$ where h is the number of iterations.*

Proof . *Omitted for brevity. See the supplementary file.* □

Convergence and Fixed Point Solution

In this section, we prove the convergence of OMNI-Prop and show the fixed point solution.

Lemma 1. *The iterative algorithm of OMNI-Prop always converges on arbitrary graphs if $\lambda > 0$.*

Proof . *Omitted for brevity. See the supplementary file.*

Theorem 2. *The fixed point solution of OMNI-Prop is written as:*

$$S_U = (I - Q_{UU})^{-1} \left(Q_{UL} S_L + \boldsymbol{r} \boldsymbol{b}^T \right), \qquad (6)$$

where

$$\begin{aligned} Q &= (D + \lambda I)^{-1} A (F + \lambda I)^{-1} A^T, \\ r &= (D_U + \lambda I_U)^{-1} \left(\lambda \mathbf{1}_U + \lambda A_U (F + \lambda I_N)^{-1} \mathbf{1}_N \right), \end{aligned}$$

Q_{UU} is the right-most and bottom-most $U \times U$ sub-matrix of Q, and Q_{UL} is the left-most and bottom-most $U \times L$ sub-matrix of Q

Proof . *Omitted for brevity. See the supplementary file.*

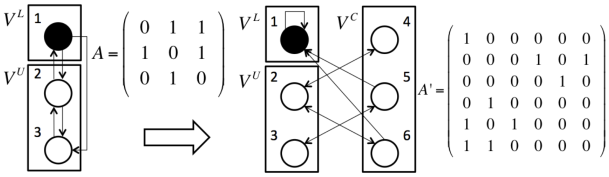
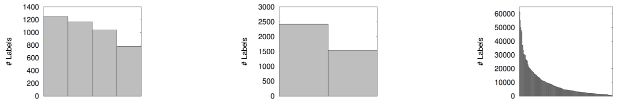


Figure 2: Twin graph construction.



(a) COAUTHOR (b) FACEBOOK (c) POKEC-L

Figure 3: **Label distributions.**

Theoretical Connection to SSL

In this section, we show the theoretical connection between the OMNI-Prop algorithm and LP, which is one of the well-known SSL algorithms. To show that, we first introduce the special graph that we call *twin graph*.

Definition 1 (Twin graph). *Twin graph $TG = (V', E')$ is a graph with $2N$ nodes where $V' = V^L \cup V^U \cup V^C$. V^C is the set of N **twin nodes** that correspond to original nodes in $V^L \cup V^U$ one by one. Nodes in twin graph TG are connected as denoted by the adjacency matrix below:*

$$A' = \begin{pmatrix} I & O & O \\ O & O & A_U \\ A_L^T & A_U^T & O \end{pmatrix}, \qquad (7)$$

where A is the adjacency matrix of the original graph G. The first L rows of A' correspond to V^L, the next U rows correspond to V^U, and the last N rows corresponds to V^C.

The intuition of the twin graph is to separate two scores of each node by assigning self-score to the original node and follower-score to its twin node. By doing so, we can discuss our algorithm in the same scheme as LP where each node has just one variable. Figure 2 illustrates an example of the twin graph construction. There are only edges between $V^L \cup V^U$ and V^C except for the self loops in V^L.

Lemma 2. *The special case of OMNI-Prop with parameter $\lambda = 0$ on graph G is equivalent to LP on twin graph TG.*

Proof . *Omitted for brevity. See the supplementary file.*

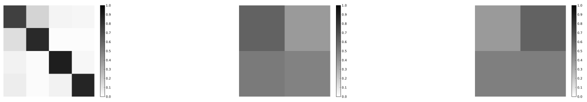
Corollary 1. *The special case of OMNI-Prop with parameter $\lambda = 0$ converges if there is at least one labeled node in each connected component on twin graph TG.*

Proof . *It follows directly from Lemma 2 and the convergence condition of LP (see (Zhu et al. 2003)).* □

Theoretical Connection to Random Walk

In this section, we show the theoretical connection between OMNI-Prop and the random walk.

Lemma 3. *The probability a random walk on twin graph TG starting at unlabeled node $v_i \in V^U$ ends at a labeled node with label k is equal to s_{ik} obtained on G with $\lambda = 0$.*



(a) COAUTHOR (b) FACEBOOK (c) POKEC-G

Figure 4: **Label correlations.**

Proof . *It follows directly from Lemma 2 and the random walk interpretation of LP (see (Zhu et al. 2003)).* □

Now, we show that the OMNI-Prop with $\lambda > 0$ can also be explained by the random walk. We introduce K *adsorbing nodes* on twin graph TG, where each adsorbing node is regarded as a labeled node with the different label out of K. Let the probability to reach the adsorbing node with label k from node $v_i \in V^U$ be $\lambda b_k/(k_i^{out} + \lambda)$, and the probability to reach it from node $v_j \in V^C$ be $\lambda b_k/(k_j^{in} + \lambda)$, where k_i^{out} and k_i^{in} are the numbers of followees and followers, respectively. On the other hand, all the other transition probabilities from node $v_i \in V^U$ or $v_j \in V^C$ are discounted by $\lambda/(k_i^{out} + \lambda)$ or $\lambda/(k_j^{in} + \lambda)$, respectively. If a random walk reaches an adsorbing node, it ends there. Using this modified twin graph TG^*, we can state the following theorem:

Theorem 3. *The probability that a random walk on modified twin graph TG^* starting at unlabeled node $v_i \in V^U$ ends at a labeled node with label k including adsorbing nodes is equal to s_{ik} obtained on G with $\lambda > 0$.*

Proof . *Omitted for brevity. See the supplementary file.*

The line of theorems shown in this section indicates that OMNI-Prop has the solid theoretical foundation and allow us to discuss it in the scheme of SSL and the random walk.

Experiments

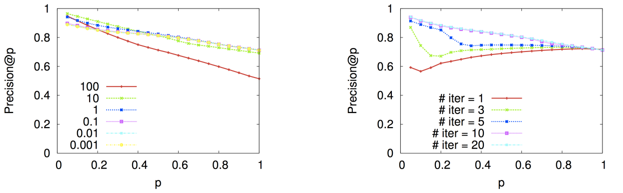
Here we answer the following questions:

- *Q1 - Parameter*: How does the parameter affect the performance of OMNI-Prop?
- *Q2 - Convergence*: How many iterations does OMNI-Prop need to converge?
- *Q3 - Accuracy*: How accurate OMNI-Prop is compared to LP and BP?

Datasets. Five network datasets[2] used in our experiments are described in Table 3. POLBLOGS is a blog-citation network where the labels are political leanings of blogs. COAUTHOR is a co-authorship network where node i and j are connected if they co-write a paper. Labels are the research fields of authors (DB, DM, ML, and AI). FACEBOOK, POKEC-G, and POKEC-L are SNSs where labels on FACEBOOK and POKEC-G are genders and labels on POKEC-L are locations of users. Figure 3 shows the label distributions (i.e., the distribution of the number of different labels) for each dataset. We omit the figures for POLBLOGS and POKEC-G because they are almost the uniform. Figure 4 shows the label correlations where cell ij illustrates the ratio of the edges from label i to j. White and black cells denote small and large values, respectively. We omit the figures for POLBLOGS and POKEC-L because the label correlations of them are clear homophily. We can see that there are varieties of labels with different characteristics.

Table 3: Datasets

	N	M	K	Directed	Correlation
POLBLOGS (Adamic and Glance 2005)	1,490	19,090	2	✓	homophily
COAUTHOR (Sun et al. 2009)	27,644	66,832	4		homophily
FACEBOOK (Leskovec and Mcauley 2012)	4,039	88,239	2		mixture slight homophily
POKEC-G (Takac and Zabovsky 2012)	1,632,803	30,622,564	2	✓	mixture slight heterophily
POKEC-L (Takac and Zabovsky 2012)	1,632,803	30,622,564	183	✓	homophily



(a) Parameter (b) Convergence

Figure 5: (a) **1.0 default**: Our algorithm achieves good result with the default parameter of 1, which means it needs no parameters to tune. (b) **10 iterations**: Our algorithm converges after about 10 iterations.

Evaluation. In our experiments, we hide labels of 70% of labeled nodes on each network. Then we perform node classification algorithm to infer hidden labels. We report the precision@p that is the precision of top p% nodes ordered by their maximum self-scores $\max_k s_{ik}$.

Reproducibility. The datasets we use in this experiments are all available on the web. Our code is also made available on the web[3].

Q1 - Parameter

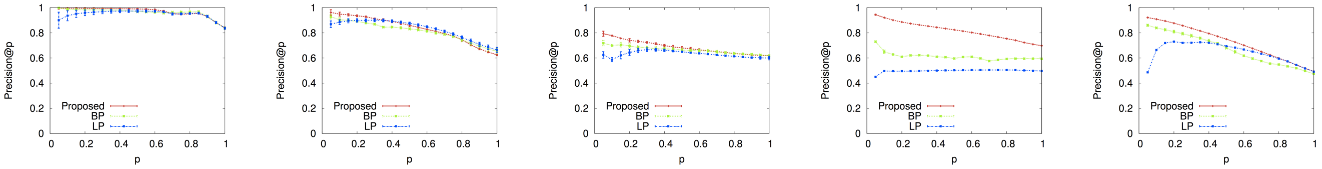
We varied the parameter from 0.001 to 100. We report only the result on POKEC-G in Figure 5(a) because the results of the rest of networks show the same trend. The x-axis and y-axis of this figure indicate the value of p and precision@p, respectively. The large value of the parameter leads to high precision for small p, but low precision for large p, whereas the small value of the parameter leads to high precision for large p, but low precision for small p. Although it is not the best result for all p, we can say that the setting of $\lambda = 1$ achieves high precision overall on all networks.

Observation 1 (1.0 default). *OMNI-Prop achieves good results with the default parameter $\lambda = 1$ on all networks.*

This observation means that practitioners do not need to tune the parameter when using our algorithm. Hence, hereafter, we use this default value for all experiments.

[2] POKEC-G and POKEC-L are the same network but the different labels.

[3] http://www.kde.cs.tsukuba.ac.jp/˜yuto_ymgc



(a) POLBLOGS (b) COAUTHOR (c) FACEBOOK (d) POKEC-G (e) POKEC-L

Figure 6: **Proposed OMNI-Prop wins**: against top competitors on all datasets, especially for the small value of p.

Table 4: Summary of the results. Each cell shows the precision@p.

	POLBLOGS			COAUTHOR			FACEBOOK			POKEC-G			POKEC-L		
	10%	50%	100%	10%	50%	100%	10%	50%	100%	10%	50%	100%	10%	50%	100%
OMNI-Prop	**1.00**	**0.99**	**0.84**	**0.95**	0.86	0.63	**0.78**	**0.68**	**0.62**	**0.92**	**0.82**	**0.70**	**0.91**	**0.75**	**0.49**
BP	0.99	0.97	**0.84**	0.90	0.83	0.65	0.70	0.66	**0.62**	0.65	0.59	0.59	0.84	0.68	0.47
LP	0.94	0.97	**0.84**	0.89	**0.88**	**0.67**	0.59	0.65	0.60	0.50	0.50	0.50	0.66	0.69	**0.49**

Q2 - Convergence

We varied the number of iterations of the iterative algorithm of OMNI-Prop. Figure 5(b) shows the results on POKEC-G. Figure 5(b) shows the results on POKEC-G. Here, we also omit the results of the rest of networks for the same reason. According to the results, we can say that OMNI-Prop almost converges after 10 iterations on all networks.

Observation 2 (10 iterations). *OMNI-Prop converges after about* 10 *iterations on all networks.*

Q3 - Accuracy

We compared OMNI-Prop with LP and BP by 5 trials using randomly constructed 5 test sets for all networks to compute the standard deviations of the precision.

BP requires the $K \times K$ *propagation matrix* as a parameter that specifies label correlations between each pair of labels (Pandit et al. 2007). We use the matrix with diagonal elements $1/K + \alpha$, and off-diagonal elements $1/K - \alpha/(K-1)$. According to our preliminary study, we determine the value of α as 0.001 for POLBLOGS, COAUTHOR, FACEBOOK, and POKEC-L, while -0.001 for POKEC-G. With such propagation matrices, BP converges for all these datasets and shows good results. Note that we also tried propagation matrices constructed from the label correlation shown in Figure 4, but they lead to poor results, which indicates that the parameter tuning for BP is not trivial.

Figure 6 and Table 4 show the results. Error bars in the figure show the standard deviation. Overall, we can see that OMNI-Prop performs either at least equaling or surpassing two competitors on all types of label correlations in terms of precision. Moreover, OMNI-Prop achieves significant improvement against tow competitors (1) on networks with the mixture label correlation, and (2) for small p value (i.e., nodes with much evidence). These results demonstrate that OMNI-Prop works well on various types of label correlations, and can consider the amount of evidence.

Observation 3. *OMNI-Prop outperforms LP and BP on various kinds of label correlations, especially for nodes with much evidence.*

Conclusion

We presented OMNI-Prop, which addresses the node classification problem on networks. The main idea is to assign each node with two scores and to calculate them on mutually-induced and Bayesian-like fashion. The advantages of our algorithm are:

1. *Seamless* and *accurate*: it outperforms top competitors on arbitrary label correlations (Figure 6 and Table 4).
2. *Fast*: each iteration of the algorithm scales linearly with the input size, and is proved to converge on any graphs (Theorem 1, Lemma 1, and Figure 5(b)).
3. *Quasi-parameter free*: it has just one parameter with the effective default value 1 (Figure 5(a)).

We also showed the set of theorems that state theoretical connections to the SSL algorithms (Lemma 2) and the random-walk (Theorem 3). We believe our proposal provides practitioners in the AI and Web community with simple but effective way to perform node classification.

Acknowledgements: This research was partly supported by the program *Research and Development on Real World Big Data Integration and Analysis* of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and by JSPS KAKENHI, Grant-in-Aid for JSPS Fellows #242322. This material is also based upon work supported by the National Science Foundation (Grants No. IIS-1217559 and CNS-1314632), and by the Army Research Laboratory (under Cooperative Agreement Number W911NF-09-2-0053). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Adamic, L. A., and Glance, N. 2005. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, 36–43. ACM.

Baluja, S.; Seth, R.; Sivakumar, D.; Jing, Y.; Yagnik, J.; Kumar, S.; Ravichandran, D.; and Aly, M. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, 895–904. ACM.

Bilgic, M., and Getoor, L. 2010. Active inference for collective classification. In *AAAI*.

Faloutsos, M.; Faloutsos, P.; and Faloutsos, C. 1999. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, 251–262. ACM.

Goldberg, A. B.; Zhu, X.; and Wright, S. J. 2007. Dissimilarity in graph-based semi-supervised classification. In *International Conference on Artificial Intelligence and Statistics*, 155–162.

Gong, C.; Tao, D.; Fu, K.; and Yang, J. 2014. Relish: Reliable label inference via smoothness hypothesis. In *AAAI*, 1840–1846.

Ha, J.; Kwon, S.-H.; Kim, S.-W.; Faloutsos, C.; and Park, S. 2012. Top-n recommendation through belief propagation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2343–2346. ACM.

Ikeda, D.; Takamura, H.; and Okumura, M. 2008. Semi-supervised learning for blog classification. In *AAAI*, 1156–1161.

Jeh, G., and Widom, J. 2003. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, 271–279. ACM.

Jurgens, D. 2013. That's what friends are for: Inferring location in online social media platforms based on social relationships. In *ICWSM*. AAAI Press.

Leskovec, J., and Mcauley, J. J. 2012. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, 539–547.

Liben-Nowell, D., and Kleinberg, J. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58(7):1019–1031.

Liu, Q.; Chen, E.; Xiong, H.; and Ding, C. H. 2010. Exploiting user interests for collaborative filtering: interests expansion via personalized ranking. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 1697–1700. ACM.

Mislove, A.; Viswanath, B.; Gummadi, K. P.; and Druschel, P. 2010. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, 251–260. ACM.

Neville, J., and Jensen, D. 2000. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, 13–20. AAAI Press.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web.

Pandit, S.; Chau, D. H.; Wang, S.; and Faloutsos, C. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*, 201–210. ACM.

Pearl, J. 1982. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, 133–136.

Pham, T. P.; Ng, H. T.; and Lee, W. S. 2005. Word sense disambiguation with semi-supervised learning. In *AAAI*, 1093–1098.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine* 29(3):93.

Sun, J.; Qu, H.; Chakrabarti, D.; and Faloutsos, C. 2005. Neighborhood formation and anomaly detection in bipartite graphs. In *Data Mining, Fifth IEEE International Conference on*, 8–pp. IEEE.

Sun, Y.; Han, J.; Gao, J.; and Yu, Y. 2009. itopicmodel: Information network-integrated topic modeling. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, 493–502. IEEE.

Takac, L., and Zabovsky, M. 2012. Data analysis in public social networks. In *International Scientific Conference AND International Workshop Present Day Trends of Innovations*.

Tong, W., and Jin, R. 2007. Semi-supervised learning by mixed label propagation. In *AAAI*, 651–656.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. *Advances in neural information processing systems* 16(16):321–328.

Zhu, X.; Ghahramani, Z.; Lafferty, J.; et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, 912–919. AAAI Press.