The Power of Certainty: A Dirichlet-Multinomial Model for Belief Propagation

Dhivya Eswaran
Carnegie Mellon University
Pittsburgh, USA
deswaran@cs.cmu.edu

Stephan Günnemann
Technical University of Munich
Munich, Germany
guennemann@in.tum.de

Christos Faloutsos
Carnegie Mellon University
Pittsburgh, USA
christos@cs.cmu.edu

Abstract

Given a friendship network, how certain are we that Smith is a progressive (vs. conservative)? How can we propagate these certainties through the network? While Belief propagation marked the beginning of principled label-propagation to classify nodes in a graph, its numerous variants proposed in the literature fail to take into account uncertainty during the propagation process. As we show, this limitation leads to counter-intuitive results for even simple graphs. Motivated by these observations, we formalize axioms that any node classification algorithm should obey and propose NetConf which satisfies these axioms and handles arbitrary network effects (homophily/heterophily) at scale. Our contributions are: (1) *Axioms*: We state axioms that any node classification algorithm should satisfy; (2) *Theory*: NetConf is grounded in a Bayesian-theoretic framework to model uncertainties, has a closed-form solution and comes with precise convergence guarantees; (3) *Practice*: Our method is easy to implement and scales linearly with the number of edges in the graph. On experiments using real world data, we always match or outperform BP while taking less processing time.

1 Introduction

Suppose Smith has to choose between iOS and android phones based on inputs from Alice and Bob (Figure 1). Alice (pink/dotted), a stubborn tech-geek, after some research believes that iOS is (60-40) better than android. Non-techie Bob (green/solid) favors android (65-35). Which phone would Smith buy? If Smith takes into account only friends' beliefs, he would be swayed by Bob towards android; however, considering their certainty/stubbornness, he would choose iOS. In an online setting, knowing the browsing and buying patterns of Alice and Bob, what ad (iOS/android phone) should we show Smith? The fundamental question is: how can we capture these notions of *certainty/stubbornness* and leverage them to classify nodes in a network?

Network effects appear in many real life scenarios, usually as homophily ("birds of a feather flock together"), or heterophily ("opposites attract") and occasionally a combination of both. Knowing the nature of network effects that apply in a given scenario, we may reason from observed training cases directly to test cases; this is called *transductive* inference. Belief Propagation (BP) [26] has been successfully used to perform such inference in numerous areas [2, 6].

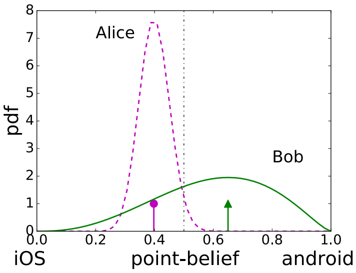
However, BP still suffers from one big limitation: it does not take the uncertainty of beliefs into account. Mathematically, BP computes point estimates only, as opposed to full distributions capturing the uncertainty in the beliefs. Thus, when propagating information, BP treats certain and uncertain nodes with equal weight, resulting in counter-intuitive responses, like recommending android to Smith in Figure 1.

The intuition pays off, as is seen from Figure 1b,c. Our method, NetConf (NETwork effects with CONfidence) takes certainty into account, and produces a sound ranking of database authors (from the DBLP co-authorship network – see Section 5 for more details).

The list of top five authors using NetConf (Figure 1b) includes authors who wrote many milestone database papers and collaborated with many well-known DB authors. In contrast, BP (Figure 1c) ignores certainty and results in *numerous* authors having perfect belief score and tying in first place; for several of them we could not find the h-index ('dash'). Informally, the problem we address is the following:

Problem 1. ***Node classification (with certainty)***
Given *a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, labels*[1] $l_v \in \{1, 2, \ldots, k\}$ *for a subset of the nodes* $v \in \mathcal{V}$ *(with their uncertainties) and the nature of network effects (e.g., homophily),*
find *the probability (belief/leaning)* $b_u(i)$ *that node* u *has label* i *along with a measure of certainty (stubbornness).*

[1] We use the terms label and class interchangeably throughout the paper.



(a)

Author	Score	H-index
Michael J. Carey	2.23	48
Rakesh Agrawal	2.20	96
Jiawei Han	2.00	139
Hamid Pirahesh	1.94	40
David J. DeWitt	1.84	81
Serge Abiteboul	1.80	77

(b)

Author	Score	H-index
Jiawei Han	1.00	139
Annie W. Shum	1.00	-
Werner Kießling	1.00	-
Xiaofang Zhou	1.00	36
Bertram Ludäscher	1.00	45
Amarnath Gupta	1.00	-

(c)

Figure 1: (a) Motivation: Who sways our opinion? Alice (certain, 60-40 iOS) or Bob (uncertain, 65-35 android)? (b) Top DB authors using NetConf (c) Top DB authors using BP (ties broken randomly). H-index was obtained from google scholar or `http://web.cs.ucla.edu`.

The main ideas behind our method are to: (i) model beliefs as Dirichlet distributions to capture uncertainty and (ii) use multinomial counts as messages to propagate these uncertainties along the edges of the network. Our contributions are as follows:

- *Theory*: We propose axioms that every network-effect method should obey; and a Bayesian theoretic model for uncertainty. These lead to our proposed NetConf, which has a closed-form solution (Theorem 4.2) and precise convergence guarantees (Theorem 4.3).
- *Practice*: NetConf is more accurate than BP, as we show with real data; it scales linearly with the number of edges and is usually faster than BP.

Reproducibility: The datasets we used are already public; our source code is available at `http://www.cs.cmu.edu/~deswaran/code/netconf.zip`.

The outline of the paper is typical (background, method, analysis, experiments and conclusions).

2 Background: Belief Propagation

Belief propagation (BP, in short), introduced by Judea Pearl [19] is a general technique to perform *approximate inference* in various graphical models such as Bayesian networks, pairwise Markov random fields and factor graphs [26]. Due to our interest in solving the node classification problem in an undirected graph, we will restrict our discussion of BP to pairwise Markov random fields.

The core idea in BP is for each node u to maintain its *belief* $\mathbf{b}_u$, a k-dimensional vector (where k is the number of classes) in which the i^{th} entry indicates the probability that node u belongs to class i. The belief of a node evolves as it receives *messages* from its neighbors. A message $\mathbf{m}_{vu}$ sent from v to u encodes v's belief about what class the node u should belong to.

	Conservative	Progressive
Conservative	0.9	0.1
Progressive	0.1	0.9

(a)

	A	B
A	0.5	0.5
B	0.5	0.5

(b)

	Silent	Talkative
Silent	0.2	0.8
Talkative	0.8	0.2

(c)

Figure 2: Example edge compatibility matrices $\mathbf{H}$ for a binary class problem. (a) Homophily: friendship (b) No network effects: blood group (c) Heterophily: dating

Beginning with *prior beliefs* $\mathbf{e}_u$ for each node $u \in \mathcal{V}$, the algorithm iteratively propagates messages and computes beliefs guided by the following update rules.

$$(2.1)\quad b_u(i) \leftarrow \frac{1}{Z_u} e_u(i) \prod_{v \in \mathcal{N}(u)} m_{vu}(i)$$

$$(2.2)\quad m_{vu}(i) \leftarrow \sum_{j=1}^{k} H(i,j) e_v(j) \prod_{w \in \mathcal{N}(v) \setminus u} m_{wv}(j)$$

Here, Z_u is a normalization constant which ensures that the beliefs sum up to 1. The $k \times k$ matrix $\mathbf{H}$ is the *edge potential* or *compatibility matrix*, which captures the *affinity* between the classes. The larger an entry $H(i,j)$, the more likely a node with class i connects to a node with class j. Thus, it can encode any kind of *network effects* such as (a) *homophily* (Figure 2a), (b) *heterophily* (Figure 2c) or (c) a combination there of, for more than two classes.

Further, observe that, when v sends a message to u, it does not take into account the message it previously received from u. This is known as *echo-cancellation*.

The method converges to exact marginals only in graphs without loops [19] and in certain special cases [15]. In the presence of loops, the algorithm is not guaranteed to converge to the true marginals, or even

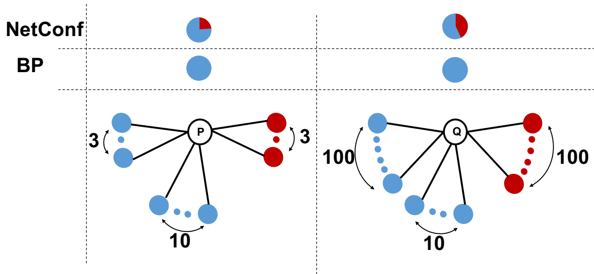


Figure 3: Ratio vs. Difference: BP gives a strong *blue* prediction for Q even though Q has an fairly equal number of *red* and *blue* neighbors.

converge at all. However, in practice, *loopy* belief propagation has been found to approximate the true marginals well [16] in a variety of applications [5, 9, 18].

3 Axioms

Figure 1a demonstrated that the direct application of BP (or similar algorithms) to node classification problems in a graph often *leads to counter-intuitive results*. This phenomenon is common; an another example is Figure 3. BP's results depend on the difference in the number of blue and red neighbors, but not the actual ratio, as one would desire.

The key to address these problems is to quantify the *uncertainty* in beliefs using distributions. In this section, we set up three axioms that our proposed method, operating on belief distributions, must obey.

Axiom 3.1. (**No network effects**) *In the absence of network effects, i.e., when the class labels are indifferent to each other, the final belief distribution of every node should match its prior belief distribution.*

Axiom 3.2. (**Certainty pulls**) *In the presence of network effects, all else being equal, neighbors with more certain belief distributions have a greater influence on a node's belief distribution. Informally, stubborn neighbors are more convincing.*

Axiom 3.3. (**Certainty pools**) *In the presence of network effects, all else being equal, an increase in certainty of a neighbor's belief distribution makes a node's belief distribution more certain. Informally, stubborn neighbors make you more stubborn.*

As we will see later, Eq. (4.7) ensures that our proposed NetConf obeys Axiom 3.1, by propagating flat (uninformative) distributions. Our update rules together ensure that a node with high certainty sends a heavy-weight (as measured by L_1 norm) message according to Eq. (4.8), which in turn has a greater influence on its neighbors' beliefs (Axiom 3.2) and increases

Entity/Operator	Notation
Scalar	lowercase, italics; e.g., n, k
Vector	bold, lowercase, without tilde; e.g., $\mathbf{b}_u, \breve{\mathbf{e}}_u$
Distribution	bold, lowercase, with tilde; e.g., $\tilde{\mathbf{b}}_u, \tilde{\mathbf{m}}_{vu}$
Matrix	bold, uppercase; e.g., $\breve{\mathbf{B}}, \mathbf{H}$
Vectorization	$vec(.)$
Set	calligraphic, capital; e.g., $\mathcal{V}, \mathcal{E}$
Kronecker product	$\otimes$
Vector/matrix entry	Not bold; e.g., $b_u(i), H(i,j)$
Spectral radius	$\rho(.)$

Table 1: Notation

Symbol	Meaning
n	$\lvert\mathcal{V}\rvert$, #nodes in the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
k	number of classes
u, v, w	nodes
i, j	classes
$\mathbf{b}_u, \mathbf{e}_u$	k-dim final, prior belief vectors of u
$\mathbf{m}_{vu}$	k-dim message vector from v to u
$\tilde{\mathbf{b}}_u, \tilde{\mathbf{e}}_u$	final, prior belief distributions of u
$\tilde{\mathbf{m}}_{vu}$	message distribution from v to u
$\breve{\mathbf{b}}_u, \breve{\mathbf{e}}_u$	k-dim final, prior D-belief vectors of u
$\breve{\mathbf{m}}_{vu}$	k-dim D-message vector from v to u
$\breve{\mathbf{B}}, \breve{\mathbf{E}}$	$n \times k$ final, prior D-belief matrices
$vec(\breve{\mathbf{B}}), vec(\breve{\mathbf{E}})$	$nk \times 1$ vectorized matrices $\breve{\mathbf{B}}, \breve{\mathbf{E}}$
$\mathbf{x}_u$	k-dim point belief from $\breve{\mathbf{b}}_u$ or $\breve{\mathbf{e}}_u$
ϕ	continuous potential function
$\mathbf{H}$	$k \times k$ compatibility matrix
$\mathbf{M}$	$k \times k$ modulation matrix
$\mathbf{A}$	$n \times n$ adjacency matrix
$\mathbf{D}$	$n \times n$ diagonal degree matrix

Table 2: Nomenclature

their certainty (Axiom 3.3) according to Eq. (4.9). These are further illustrated using an example in Section 5.1. We now describe our approach.

4 Proposed Approach

In Figure 1, Alice is lukewarm towards iOS but very certain about her opinion, while Bob is the reverse. Thus, we need to capture both the leaning/belief of a node (e.g., preference to iOS vs android) as well as its stubbornness/certainty. At a high level, the heart of our idea is to use a Beta distribution with two parameters $(\alpha+1, \beta+1)$ as depicted in Figure 1. The leaning of a node is the ratio $\frac{\alpha}{\alpha+\beta}$, while its certainty is the height of the spike of the Beta distribution captured through $\alpha+\beta$. For a multi-class case, we generalize this to the *Dirichlet* distribution. Our approach is based on the following steps:

- **Dirichlet Beliefs:** The *D-belief* (*Dirichlet-belief*) $\breve{\mathbf{b}}_u$ of a node u is a k-d vector of reals which parameterize its belief distribution.
- **Multinomial Messages:** The *D-message* $\breve{\mathbf{m}}_{vu}$ from node v to u is a k-d vector of multinomial counts.
- **Network effects:** The *modulation* matrix $\mathbf{M}$ is derived carefully from the compatibility matrix $\mathbf{H}$ (to obey Axiom 3.1).
- **NetConf update rules:** We derive update rules (Eq. (4.8) and Eq. (4.9)) in terms of D-beliefs, D-messages and modulation matrix from Yedidia's update rules (Eq. (2.1) and Eq. (2.2)).
- **Closed-form solution:** From these update rules, we derive NetConf's recursive matrix equation (Theorem 4.1), compute the closed-form solution (Theorem 4.2), and provide *necessary and sufficient* convergence guarantees (Theorem 4.3).

Table 1 summarizes the notation and Table 2 lists the frequently used symbols. The rest of the section describes the above steps in detail.

4.1 Dirichlet beliefs. A principled way to model the uncertainty in k-d beliefs is through a distribution having a $k-1$-d simplex as support, namely, the *Dirichlet distribution*. Its probability density function is given by: $p(\mathbf{x};\alpha) \propto \prod_{i=1}^{k} x_i^{\alpha_i - 1}$. The *concentration parameters* $\alpha_1, \ldots, \alpha_k$ are k real-valued numbers which control the spread of the distribution in space. Let us use D-belief ($\breve{\mathbf{b}}_u$) (analogously, D-prior $\breve{\mathbf{e}}_u$) to denote the parameters of u's belief distribution minus 1.

$$\tilde{\mathbf{b}}_u(\mathbf{x}_u) = Dir(\mathbf{x}_u; \breve{\mathbf{b}}_u + \mathbf{1}) \tag{4.3}$$

$$\tilde{\mathbf{e}}_u(\mathbf{x}_u) = Dir(\mathbf{x}_u; \breve{\mathbf{e}}_u + \mathbf{1}) \tag{4.4}$$

As the scale of D-belief increases, the distribution begins to get peakier (*certain*) around its mean; hence, we may quantify the *certainty in belief* as $\text{Certainty}(\breve{\mathbf{b}}_u) = \sum_i \breve{b}_u(i)$. Our richer model for beliefs maintains only k-parameters at every node, similar to BP.

4.2 Multinomial messages. If beliefs are distributions, how should we characterize messages? The key lies in interpreting Eq. (2.1) as an equation that guides *Bayesian posterior estimation*:

$$\tilde{\mathbf{b}}_u(\mathbf{x}_u) \propto \tilde{\mathbf{e}}_u(\mathbf{x}_u) \prod_{v \in \mathcal{N}(u)} \tilde{\mathbf{m}}_{vu}(\mathbf{x}_u) \tag{4.5}$$

We hypothesize that the message distributions are the likelihood of *observations* made by a node about its neighbors. For tractability of estimation (using conjugacy of Dirichlet-Multinomial distributions), we let observations be *multinomial counts*. Accordingly, the message distribution $\tilde{\mathbf{m}}_{vu}$ from v to u is the likelihood of the message counts (D-message $\breve{\mathbf{m}}_{vu}$) under the belief distribution $\tilde{\mathbf{b}}_u(\mathbf{x}_u)$ of the node which receives the message:

$$\tilde{\mathbf{m}}_{vu}(\mathbf{x}_u; \breve{\mathbf{m}}_u) \propto \prod_{i=1}^{k} x_u(i)^{\breve{m}_u(i)}$$

Plugging this in Eq. (4.5), we derive NetConf's first update rule:

$$\breve{\mathbf{b}}_u \leftarrow \breve{\mathbf{e}}_u + \sum_{v \in \mathcal{N}(u)} \breve{\mathbf{m}}_{vu}$$

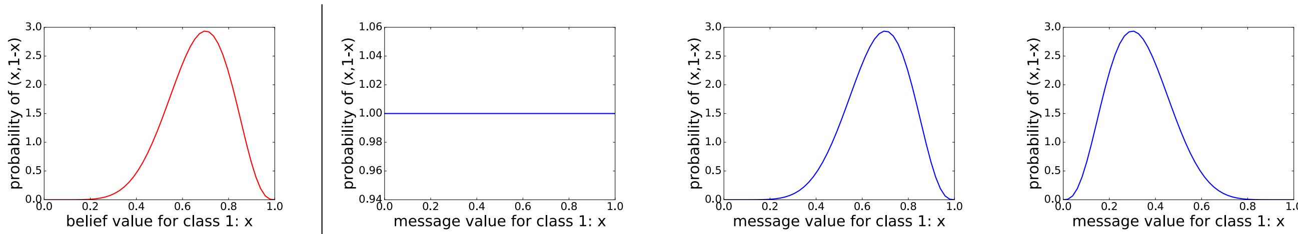
4.3 Network effects. Since now messages and beliefs are (continuous) distributions instead of vectors, the message update rule in Eq. (2.2) needs to be adapted. We use a *continuous potential function* analogous to the compatibility matrix $\mathbf{H}$ in the discrete setting.

$$\tilde{\mathbf{m}}_{vu}(\mathbf{x}_u) \propto \int_{\mathbf{x}_v} \phi(\mathbf{x}_u, \mathbf{x}_v) \underbrace{\tilde{\mathbf{e}}_v(\mathbf{x}_v) \prod_{w \in \mathcal{N}(v) \backslash u} \tilde{\mathbf{m}}_{wv}(\mathbf{x}_v)}_{\tilde{\mathbf{b}}_{v \backslash u}} \tag{4.6}$$

Suppose the compatibility matrix $\mathbf{H}$ for a two class problem is $\begin{pmatrix} 0.5+\epsilon & 0.5-\epsilon \\ 0.5-\epsilon & 0.5+\epsilon \end{pmatrix}$ where ϵ indicates the nature of network effects. $\epsilon = 0.5$ is perfect homophily; $\epsilon = -0.5$ is perfect heterophily; $\epsilon = 0$ is the case of no network effects. Intermediate positive and negative values correspond to varying degrees of homophily and heterophily respectively.

Denote with ϕ_ϵ the (unknown) continuous potential function that reflects the corresponding scenario for a specific value of ϵ. Let $\tilde{\mathbf{b}}_{v\backslash u}$ be the echo-cancelled belief distribution from Eq. (4.6). It is desirable that the checkpoints in Table 3 hold, as also illustrated in Figure. 4. The intuition is as follows: (1) for no network effects, the message should not prefer any belief value over the other (flat distribution); (2) for perfect homophily, a node believes about its neighbors what it believes about itself; (3) for perfect heterophily, a node believes about its neighbors the opposite of what it believes about itself.

Despite the mathematical niceness of the above formulation, it has proved hard to define a potential function that (i) preserves the functional form of message distributions, (ii) satisfies the checkpoints in Table 3, and (iii) ensures efficient computation. Thus, we propose to approximate the continuous potential function



(a) Echo-cancelled belief distribution $\tilde{\mathbf{b}}_{v\backslash u}$

(b) $\tilde{\mathbf{m}}_{vu}$ is flat for no network effects ($\epsilon = 0$)

(c) $\tilde{\mathbf{m}}_{vu} \equiv \tilde{\mathbf{b}}_{v\backslash u}$, for perfect homophily ($\epsilon = +0.5$)

(d) $\tilde{\mathbf{m}}_{vu} \equiv$ flipped $\tilde{\mathbf{b}}_{v\backslash u}$, for perfect heterophily ($\epsilon = -0.5$)

Figure 4: Understanding the corner cases of the continuous potential function: A sample 2-class echo-cancelled belief distribution and the corresponding message distributions for the different network effects

Network effects	ϵ	Checkpoint for continuous potential function ϕ	Modulation matrix $\mathbf{M}$	Geometry (Figure 5)
None	0	$\int_{\mathbf{x}_v} \phi_{\epsilon=0}(\mathbf{x}_u, \mathbf{x}_v)\tilde{\mathbf{b}}_{v\backslash u}(\mathbf{x}_v) \propto 1$	$\mathbf{M} = \mathbf{0}$	Point O (origin)
Perfect homophily	0.5	$\int_{\mathbf{x}_v} \phi_{\epsilon=0.5}(\mathbf{x}_u, \mathbf{x}_v)\tilde{\mathbf{b}}_{v\backslash u}(\mathbf{x}_v) \propto \tilde{\mathbf{b}}_{v\backslash u}(\mathbf{x}_u)$	$\mathbf{M} = \mathbf{I}$	Point A (identical to $\tilde{\mathbf{b}}_{v\backslash u}$)
Perfect heterophily	-0.5	$\int_{\mathbf{x}_v} \phi_{\epsilon=-0.5}(\mathbf{x}_u, \mathbf{x}_v)\tilde{\mathbf{b}}_{v\backslash u}(\mathbf{x}_v) \propto \tilde{\mathbf{b}}_{v\backslash u}(\mathbf{1} - \mathbf{x}_u)$	$\mathbf{M} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Point B (image about $x = y$)

Table 3: NetConf corner cases of network effects: Checkpoints for ϕ and corresponding instantiations of $\mathbf{M}$

ϕ that operates on the distributions by a *modulation matrix* $\mathbf{M}$ that operates on the corresponding hyperparameters. Following the update rule of the belief distribution, the message update for the hyperparameters is defined by $\breve{\mathbf{m}}_{vu} \leftarrow \mathbf{M}(\breve{\mathbf{e}}_v + \sum_{w \in \mathcal{N}(v)\backslash u} \breve{\mathbf{m}}_{wv})$.

To formally define the modulation matrix $\mathbf{M}$, let us visualize the D-beliefs and D-messages for a two class problem as points on a 2D plot, as shown in Figure 5. The x-axis represents the D-score of a belief or message for the first class, while the y-axis represents the D-score for the second class. Let A represent the D-scores of the echo-cancelled belief of node u, i.e., $\breve{\mathbf{b}}_v - \breve{\mathbf{m}}_{uv}$. The three conditions on ϕ determine how the modulation matrix $\mathbf{M}$ is defined for the corner cases of $\epsilon = 0.5, 0, +0.5$ – these correspond to points A, O and B in Figure 5 respectively (see also Table 3). For any intermediate positive value of ϵ, we propose a linear interpolation and transmit the D-message lying on the line AO. Similarly, for intermediate negative values of ϵ, the D-message takes a value lying on OB. Hence, the modulation matrix for a two-class problem is given by

$$\mathbf{M} = 2 \begin{pmatrix} L(\epsilon) & L(-\epsilon) \\ L(-\epsilon) & L(\epsilon) \end{pmatrix}$$

where $L(.)$ is the *Lasso operator* defined as $L(x) = x$ for $x > 0$, and 0 otherwise. This can be generalized to the k-class case as:

$$\mathbf{M} = \frac{k}{k-1} L\left(\mathbf{H} - \frac{1}{k}\right) \quad (4.7)$$

NetConf obeys Axiom 3.1: In the absence of network effects, $\mathbf{M} = \mathbf{0}$. This makes all message counts

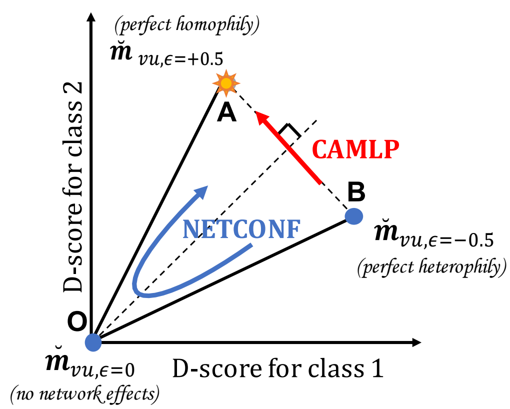


Figure 5: Illustration of modulated messages as a function of ϵ: Proposed NetConf follows blue arrow (as ϵ increases from -0.5 to +0.5) and sends (0,0) message for no network effects. Messages according to our competitor CAMLP [25] follow the red arrow and violate Axiom 3.1 (no network effects).

zero (i.e., message distributions flat), hence leaving the belief distributions of all nodes unchanged.

4.4 Putting things together - NetConf. The update rules for NetConf, in terms of the modulation matrix $\mathbf{M}$ (Eq. (4.7)) can be summarized as:

$$\breve{\mathbf{b}}_u \leftarrow \breve{\mathbf{e}}_u + \sum_{v \in \mathcal{N}(u)} \breve{\mathbf{m}}_{vu} \quad (4.8)$$

$$\breve{\mathbf{m}}_{vu} \leftarrow \mathbf{M}(\breve{\mathbf{e}}_v + \sum_{w \in \mathcal{N}(v)\backslash u} \breve{\mathbf{m}}_{wv}) \quad (4.9)$$

NetConf obeys Axiom 3.2 and Axiom 3.3: A node with high certainty sends a heavy-weight (as measured by L_1 norm) message due to Eq. (4.8). This increases its influence on its neighbors' beliefs (Axiom 3.2) and hence their certainty (Axiom 3.3) according to Eq. (4.9).

While in principle one can simply invoke the previous two update equations several times until the messages and beliefs converge, we infer a more efficient variant that avoids computing messages at all. We will use the following notation. Let $\mathcal{G}$ be an unweighted undirected graph on n nodes, with adjacency matrix $\mathbf{A}$. Let $\mathbf{D}$ be the diagonal degree matrix, where $D(q,q) = d_q$, the degree of the q^{th} node. Also, suppose that k is the number of classes. Then, we construct the $n \times k$ D-belief matrix $\breve{\mathbf{B}}$ (and correspondingly, the D-prior matrix $\breve{\mathbf{E}}$), by stacking D-belief (resp., D-prior) row vectors of all nodes one below the other. Now, we are ready to state our main theorem.

Theorem 4.1. (NetConf) *For matrices* $\mathbf{A}$, $\mathbf{D}$, $\breve{\mathbf{B}}$, $\breve{\mathbf{E}}$ *and* $\mathbf{M}$ *described as above, the final D-beliefs of nodes are given by the equation system:*

$$\breve{\mathbf{B}} = \breve{\mathbf{E}} + (\mathbf{A}\breve{\mathbf{B}}\mathbf{M} - \mathbf{D}\breve{\mathbf{B}}\mathbf{M}^2)(\mathbf{I} - \mathbf{M}^2)^{-1} \tag{4.10}$$

Proof. Rewriting the D-message update rule from Eq. (4.9) in terms of D-belief $\breve{\mathbf{b}}_u$, we have

$$\breve{\mathbf{m}}_{vu} \leftarrow \mathbf{M}(\breve{\mathbf{b}}_v - \breve{\mathbf{m}}_{uv})$$

Plugging the message update rule for $\breve{\mathbf{m}}_{uv}$ into the above yields

$$\breve{\mathbf{m}}_{vu} \quad \leftarrow \quad \mathbf{M}(\breve{\mathbf{b}}_v - \mathbf{M}(\breve{\mathbf{b}}_u - \breve{\mathbf{m}}_{vu}))$$

At steady state, we can replace the update sign with an equality and solve for $\breve{\mathbf{m}}_{vu}$, in terms of the steady state D-beliefs $\breve{\mathbf{b}}_u, \breve{\mathbf{b}}_v$. This gives us

$$\breve{\mathbf{m}}_{vu} \quad = \quad (\mathbf{I} - \mathbf{M}^2)^{-1}(\mathbf{M}\breve{\mathbf{b}}_v - \mathbf{M}^2\breve{\mathbf{b}}_u) \tag{4.11}$$

Now, the steady state D-beliefs can be calculated from the steady state D-messages using Eq. (4.8).

$$\breve{\mathbf{b}}_u \quad = \quad \breve{\mathbf{e}}_u + (\mathbf{I} - \mathbf{M}^2)^{-1} \sum_{v \in \mathcal{N}(u)} (\mathbf{M}\breve{\mathbf{b}}_v - \mathbf{M}^2\breve{\mathbf{b}}_u)$$

Rewriting this in matrix form using the previously defined matrices ($\breve{\mathbf{B}}, \breve{\mathbf{E}}, \mathbf{A}$ and $\mathbf{D}$) yields Eq. (4.10).

As shown, Eq. (4.10) operates on beliefs only; the messages are not explicitly required. In practice, we can use the above result to compute the final belief matrix via an efficient iterative update of the following form:

$$\breve{\mathbf{B}}^{(t+1)} = \breve{\mathbf{E}} + (\mathbf{A}\breve{\mathbf{B}}^{(t)}\mathbf{M} - \mathbf{D}\breve{\mathbf{B}}^{(t)}\mathbf{M}^2)(\mathbf{I} - \mathbf{M}^2)^{-1} \tag{4.12}$$

Weighted edges: Although our proof assumes unweighted edges, it can be easily shown that all our theorems hold for weighted adjacency matrix $\mathbf{A}$ as well.

4.5 Closed-form solution and convergence Before providing theoretical guarantees for our algorithm, we review two useful matrix algebra concepts.

Definition 4.1. (Matrix Vectorization [12]) *Vectorization of an* $m \times n$ *matrix converts it into a* $mn \times 1$ *vector given by:*

$$vec(\mathbf{X}) = [x_{11}, \ldots, x_{n1}, x_{12}, \ldots, x_{n2}, \ldots, x_{1n}, \ldots, x_{nn}]^T$$

where x_{ij} *denotes the element in the* i^{th} *row and* j^{th} *column of matrix* $\mathbf{X}$.

Lemma 4.1. (Roth's column lemma [12]) *For any three matrices* $\mathbf{X}, \mathbf{Y}$ *and* $\mathbf{Z}$,

$$vec(\mathbf{XYZ}) \quad = \quad (\mathbf{Z}^T \otimes \mathbf{X})vec(\mathbf{Y}) \tag{4.13}$$

where $\otimes$ *is the Kronecker product [12].*

Theorem 4.2. (Closed Form Solution) *For matrices* $\mathbf{A}$, $\mathbf{D}$, $\mathbf{M}$ *and vectors* $vec(\breve{\mathbf{B}})$ *and* $vec(\breve{\mathbf{E}})$ *described as above, the closed form solution for D-beliefs is*

$$vec(\breve{\mathbf{B}}) = (\mathbf{I} - (\mathbf{M}\hat{\mathbf{M}})^T \otimes \mathbf{A} + (\mathbf{M}^2\hat{\mathbf{M}})^T \otimes \mathbf{D})^{-1} vec(\breve{\mathbf{E}}) \tag{4.14}$$

where $\hat{\mathbf{M}} = (\mathbf{I} - \mathbf{M}^2)^{-1}$.

Proof. The theorem can be proved by vectorizing Eq. (4.10) and applying Roth's column lemma.

Theorem 4.3. (Fixed Point and Convergence) *The iterative updates in Eq.* (4.12) *converge to a unique fixed point, for arbitrary initialization of the D-belief matrix, if and only if the spectral norm of* $(\mathbf{M}\hat{\mathbf{M}})^T \otimes \mathbf{A} + (\mathbf{M}^2\hat{\mathbf{M}})^T \otimes \mathbf{D}$ *is less than 1.*

$$\text{NetConf } converges \Leftrightarrow$$
$$\rho\left((\mathbf{M}\hat{\mathbf{M}})^T \otimes \mathbf{A} + (\mathbf{M}^2\hat{\mathbf{M}})^T \otimes \mathbf{D})\right) < 1 \tag{4.15}$$

Here, $\hat{\mathbf{M}} = (\mathbf{I} - \mathbf{M}^2)^{-1}$.

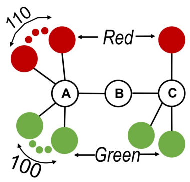
Proof. The Jacobi method of solving a system of linear equations [20] states that a linear equation system of form $\mathbf{x} = (\mathbf{I} - \mathbf{P})\mathbf{y}$ converges if and only if $\rho(\mathbf{P}) < 1$.

Rewriting the update rule in Eq. (4.12) in terms of vectorized D-priors and D-beliefs and applying the above result proves the theorem.

In practice, convergence may be ensured by setting $\mathbf{M}$ as $c\mathbf{M}$, where $c > 0$ is an appropriately chosen constant according to Theorem 4.3. Here, c can be interpreted as the *modulation decay factor* for message propagation.

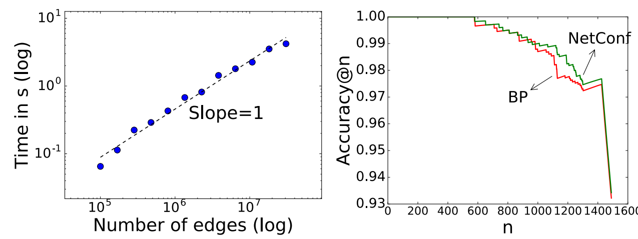
Dataset	Nodes	Edges	Description	Classes
POLBLOGS [1]	1490	19090	Political blog hyperlink network	Democrat/Republican
COAUTHOR [21]	28702	66832	Citation network	4 areas - DB, DM, AI, IR
POKEC [22]	1632803	30622564	Friendship network in Slovakia	Male/female (slight heterophily)

Table 4: Datasets used



Method	A	B	C
BP	**6.20e-9**	**0.2561**	0.5^+
CAMLP	**0.4860**	0.5005	0.5210
NETCONF	**0.4833**	**0.4967**	0.5118

Figure 6: Case Study: (a) graph with $k = 2$ classes (b) BP vs CAMLP vs NETCONF: final belief/leaning for class green (bold = red; 0.5^+ is slightly above 0.5)



(a) Scalability

(b) Accuracy@n

Figure 7: NETCONF is (a) scalable (b) outperforms the baseline, achieving better accuracy and precision

5 Experiments

In this section, we (1) present a case study to demonstrate how the top competitors, unlike NETCONF, violate our axioms and (2) experimentally verify the scalability and effectiveness of NETCONF.

5.1 Case study using synthetic data. We present a case study (Figure 6a) to illustrate how major competitors disobey our axioms. Here, A, B and C are the core nodes (unlabeled). Given the labels for the remaining peripheral nodes (red/green) and homophily network effects, we investigate the belief/leaning scores assigned by NETCONF, BP and CAMLP.

In experiments, we use $[0.1, 0.9]$ and $[0.9, 0.1]$ as prior for the red (top) and green (bottom) with nodes. The core nodes are given uniform prior $[0.5, 0.5]$. Compatibility matrix from Eq. 5.16 with $\epsilon = 0.4$ is used, with CAMLP's β set to the recommended default of 0.1. The belief/leaning returned by the three methods are tabulated in Figure 6b.

$$\mathbf{H} = \begin{pmatrix} 0.5+\epsilon & 0.5-\epsilon \\ 0.5-\epsilon & 0.5+\epsilon \end{pmatrix} \tag{5.16}$$

All three methods label A correctly as red. CAMLP and NETCONF result in a belief value which is close to 0.5 as is desirable. However, BP yields a red belief (≈ 1) despite the comparable number of red and green neighbors, which is counter-intuitive.

The classification of node B illustrates the importance of certainty well. B has two neighbors – the red A and the green C. CAMLP, which does not store/propagate certainty, compute B's belief from those of A and C, resulting in a misclassification (violation of Axiom 3.2). However, NETCONF recognizes the high certainty of A ($\approx 60\times$ neighbors) and by giving it higher weight, correctly classifies B.

Similar results were obtained for $\epsilon \in (0, 0.5)$ and $\beta \in (0, 1)$. In sum, NETCONF obeys axioms and results in intuitive classification unlike major competitors.

5.2 Experiments on real data. Our experiments use three diverse publicly available real-world datasets (Table 4). We implemented NETCONF (iterative version from Eq. (4.12)) in MATLAB, as it is well-optimized to handle sparse matrix operations. The modulation decay factor was chosen according to Theorem 4.3. Due to lack of prior work which incorporates certainty in a scalable manner, we resorted to the widely used BP as baseline. All experiments were conducted on 2.7 GHz Intel Core i5 with 16 GB main memory. Our experimental findings can be summarized under the following three categories.

Q1. Scalability: How fast and scalable is NetConf with #edges? We uniformly sampled 150K-30M edges from POKEC network and timed NETCONF and BP for 5 iterations (computations only) to allow comparability. In each case, we seeded 20% nodes and used $\mathbf{H}$ from Eq. (5.16) with $\epsilon = -0.4$ (heterophily). Fig. 7a plots running time (in seconds; averaged over 10 trials) with the network size in log-log scale.

The plot shows our algorithm scales linearly with the graph size. It was also found to be $\sim 600\times$ faster than a MATLAB implementation of BP by avoiding loops and heavy-weight operations (similar to [10]), processing upto $\sim 30M$ edges in a few seconds. This suggests that NetConf is fast and is expected to scale well to large graph applications.

Q2. Effectiveness: How accurate is NetConf? We compare overall accuracy and accuracy@n curve of NetConf against that of BP on three datasets. In all cases, we seeded 30% nodes with their true labels and prior certainty of 1 (due to lack of richer ground truth). Unlabeled nodes were initialized to $[\frac{1}{k}, \ldots, \frac{1}{k}]$ where k = #classes. The compatibility matrix $\mathbf{H}$ from Eq.(5.16) with $\epsilon = 0.4$ and -0.4 were used for homophily (Polblogs/Coauthor) and heterophily (Pokec) respectively.

(a) Overall Accuracy: The class with the highest belief/D-belief (BP/NetConf) was assigned as the class for a node, breaking any ties arbitrarily. The accuracy results (Table 5) show that NetConf consistently matches or outperforms BP and the differences are statistically significant.

(b) Accuracy@n: We compute the accuracy on top n nodes in a ranking based on the confidence of classification and plotted it as a function of n. The difference in top two beliefs was used as the ranking mechanism for BP; for NetConf, difference in top two D-beliefs was used as it incorporates certainty as well. NetConf emerged as the clear winner on Polblogs dataset, as is evident from Figure 7b. Similar trends were observed in other datasets. These results suggest that NetConf is ideal for *precision-critical* applications, e.g., fraud detection [11, 13].

Q3. Certainty Scores: Do they make sense? On the Coauthor network, we rank the authors on their score for class DB (*databases*) and list the top-5 by NetConf (Figure 1b) and by BP (Figure 1c). Authors in the former list, with high D-belief for 'DB', have several DB publications and coauthors, a high H-index and several DB-related distinctions. In contrast, BP ignores certainty and produces perfect scores for many authors, as long as they have exclusively DB coauthors and publications, no matter how many or how few. Thus, they all tie in first place; we broke ties arbitrarily and only Prof. Jiawei Han is in both lists.

In summary, our empirical studies show that NetConf (i) obeys axioms and leads to intuitive classification (ii) is faster than BP and has linear scalability; (ii) never loses to BP and usually outperforms it; (iii) produces certainty scores that reflect our expectations.

6 Related Work

Table 6 gives an overview of the differences between the methods. In summary, our proposed NetConf is the first method that (i) handles arbitrary network effects, (ii) satisfies all axioms, and (iii) gives a closed-form solution for beliefs and certainties.

Accuracy	Polblogs	Coauthor	Pokec
BP (Baseline)	91.38	76.26	73.78
NetConf	**92.40**	**81.89**	**75.02**

Table 5: Accuracy of BP vs NetConf (averaged over 5 runs): Underlined numbers indicate significant differences $p \leq 0.05$ according to a two-sided sign test.

	LP [28]	SOCNL [24]	BP [26]	Adsorption [3]	MADDL [23]	DGR [8]	TACO [17]	LinBP [10]	CAMLP [25]	NetConf
Obeys axioms						✓				✓
Homo-/hetero-phily			✓					✓	✓	✓
Scalability	✓	✓	✓	✓	✓		✓	✓	✓	✓
Closed-form	✓	✓						✓	✓	✓

Table 6: NetConf has all desirable properties

Transductive inference, a special case of semi-supervised learning, has attracted a lot of interest [4, 27]. Belief Propagation [26] is closely related, and we have described it in Section 2. BP has replaced label propagation [28] and it has been successful on node classification problems, due to its ability to handle both homophily and heterophily. However, its convergence can be guaranteed for some special graphs only [15]. Approximations to BP were able to prove convergence, for the 2-class case [14], the multi-class case [10], and heterogeneous graphs [7]. However, none of the methods can model uncertainty.

Efforts to incorporate uncertainty or confidence are recent [3, 8, 17, 23, 24, 25]. Except CAMLP [25], all are restricted to homophily effects only. Adsorption [3] and its extension MADDL [23], which propagate labels by performing a controlled random walk on the graph can only handle homophily. Dirichlet-based Graph Regularization (DRG) [8] assumes every node has a Dirichlet prior and propagates it along edges. However, it is slow, as it needs to solve an optimization problem numerically at every iteration. Transduction Algorithm with Confidence (TACO) [17] computes both belief and $k \times k$ uncertainty matrix for all nodes alternatively at every iteration. But, it penalizes high degree nodes for small differences in the beliefs of neighbors even if the neighbors indicate the same class. None of the above methods handles arbitrary network effects. SOCNL [24] and CAMLP both introduce uncertainty, but they both fail Axiom 3.1 (no network effects).

In summary, NetConf is the only method that satisfies all the specifications in Table 6.

7 Conclusions

We presented NetConf, a method to perform belief propagation along with uncertainties. The main idea was to model beliefs as Dirichlet distributions and messages as multinomial counts. Unlike existing works, NetConf follows proposed axioms, generalizes to arbitrary network effects and is highly scalable. NetConf has a closed-form solution and strong convergence guarantees. Our empirical analysis indicated the strong potential of using uncertainty in node classification tasks.

8 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. CNS-1314632 and IIS-1408924, by the German Research Foundation, Emmy Noether grant GU 1409/2-1, and by the Technical University of Munich - Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement no 291763, co-funded by the European Union. 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

[1] L. A. Adamic and N. Glance, *The political blogosphere and the 2004 us election: divided they blog*, in Workshop on Link discovery, 2005, pp. 36–43.

[2] L. Akoglu, R. Chandy, and C. Faloutsos, *Opinion fraud detection in online reviews by network effects*, in ICWSM, 2013.

[3] S. Baluja et al., *Video suggestion and discovery for youtube: Taking random walks through the view graph*, in WWW, 2008, pp. 895–904.

[4] O. Chapelle, B. Schölkopf, and A. Zien, *Transductive Inference and Semi-Supervised Learning*, 2006.

[5] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos, *Apolo: Making sense of large network data by combining rich user interaction and machine learning*, in ACM CHI, 2011.

[6] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, *Polonium: Tera-scale graph mining and inference for malware detection*, in SDM, 2011, pp. 131–142.

[7] D. Eswaran, S. Günnemann, C. Faloutsos, D. Makhija, and M. Kumar, *Zoobp: Belief propagation for heterogeneous networks*, PVLDB, 10 (2017), pp. 625–636.

[8] Y. Fang, B.-J. P. Hsu, and K. C.-C. Chang, *Confidence-aware graph regularization with heterogeneous pairwise features*, in SIGIR, 2012, pp. 951–960.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, *Efficient belief propagation for early vision*, IJCV, (2006), pp. 41–54.

[10] W. Gatterbauer, S. Günnemann, D. Koutra, and C. Faloutsos, *Linearized and single-pass belief propagation*, PVLDB, 8 (2015), pp. 581–592.

[11] S. Günnemann, N. Günnemann, and C. Faloutsos, *Detecting anomalies in dynamic rating data: a robust probabilistic model for rating evolution*, in KDD, 2014, pp. 841–850.

[12] H. V. Henderson and S. R. Searle, *The vec-permutation matrix, the vec operator and Kronecker products: a review*, Linear & Multilinear Algebra, 9 (1981), pp. 271–288.

[13] B. Hooi et al., *BIRDNEST: bayesian inference for ratings-fraud detection*, in SDM, 2016, pp. 495–503.

[14] D. Koutra, T.-Y. Ke, U. Kang, D. H. Chau, H.-K. K. Pao, and C. Faloutsos, *Unifying guilt-by-association approaches*, in ECML/PKDD, 2011.

[15] J. Mooij and H. Kappen, *Sufficient conditions for convergence of the sum-product algorithm*, IEEE Trans. on Information Theory, (2007).

[16] K. P. Murphy, Y. Weiss, and M. I. Jordan, *Loopy belief propagation for approximate inference: An empirical study*, in UAI, 1999.

[17] M. Orbach and K. Crammer, *Graph-based transduction with confidence*, in ECMLPKDD, Springer, 2012, pp. 323–338.

[18] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos, *Netprobe: A fast and scalable system for fraud detection in online auction networks*, in WWW, 2007.

[19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., 1988.

[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.

[21] Y. Sun, J. Han, J. Gao, and Y. Yu, *itopicmodel: Information network-integrated topic modeling*, in ICDM, IEEE, 2009, pp. 493–502.

[22] L. Takac and M. Zabovsky, *Data analysis in public social networks*, in International Scientific Conference and International Workshop Present Day Trends of Innovations, 2012, pp. 1–6.

[23] P. P. Talukdar and K. Crammer, *New regularized algorithms for transductive learning*, in ECML/PKDD, 2009, pp. 442–457.

[24] Y. Yamaguchi, C. Faloutsos, and H. Kitagawa, *Socnl: Bayesian label propagation with confidence*, in PAKDD, 2015, pp. 633–645.

[25] ———, *Camlp: Confidence-aware modulated label propagation*, in SDM, 2016.

[26] J. S. Yedidia, W. T. Freeman, and Y. Weiss, *Exploring artificial intelligence in the new millennium*, 2003, ch. Understanding Belief Propagation and Its Generalizations, pp. 239–269.

[27] X. Zhu, *Semi-supervised learning literature survey*, Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[28] X. Zhu, Z. Ghahramani, and J. Lafferty, *Semi-supervised learning using gaussian fields and harmonic functions*, in ICML, 2003, p. 912919.