HAL
open science

On the Troll-Trust Model for Edge Sign Prediction in Social Networks

Géraud Le Falher, Nicolò Cesa-Bianchi, Claudio Gentile, Fabio Vitale

To cite this version:

Géraud Le Falher, Nicolò Cesa-Bianchi, Claudio Gentile, Fabio Vitale. On the Troll-Trust Model for Edge Sign Prediction in Social Networks. [Research Report] INRIA Lille. 2016. hal-01425137

HAL Id: hal-01425137

https://inria.hal.science/hal-01425137v1

Submitted on 3 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Troll-Trust Model for Edge Sign Prediction in Social Networks

Géraud Le Falher[(1)] **Nicolò Cesa-Bianchi**[(2)] **Claudio Gentile**[(3)] **Fabio Vitale**[(1)]
[(1)] Inria, Univ. Lille, CNRS UMR 9189 – CRIStAL, France
[(2)] Università degli Studi di Milano, Italy
[(3)] University of Insubria, Italy

Abstract

In the problem of edge sign prediction, we are given a directed graph (representing a social network), and our task is to predict the binary labels of the edges (i.e., the positive or negative nature of the social relationships). Many successful heuristics for this problem are based on the troll-trust features, estimating at each node the fraction of outgoing and incoming positive/negative edges. We show that these heuristics can be understood, and rigorously analyzed, as approximators to the Bayes optimal classifier for a simple probabilistic model of the edge labels. We then show that the maximum likelihood estimator for this model approximately corresponds to the predictions of a Label Propagation algorithm run on a transformed version of the original social graph. Extensive experiments on a number of real-world datasets show that this algorithm is competitive against state-of-the-art classifiers in terms of both accuracy and scalability. Finally, we show that troll-trust features can also be used to derive online learning algorithms which have theoretical guarantees even when edges are adversarially labeled.

1 Introduction

Connections in social networks are mostly driven by the *homophily* assumption: linked individuals tend to be similar, sharing personality traits, attitudes, or interests. However, homophily alone is clearly not sufficient to explain the variety of social links. In fact, sociologists have long studied networks, hereafter called *signed* social networks, where also *negative* relationships —like dissimilarity, disapproval or distrust— are explicitly displayed. The presence of negative relationships is also a feature of many technology-mediated social networks. Known examples are Ebay, where users trust or distrust agents in the network based on their personal interactions, Slashdot, where each user can tag another user as friend or foe, and Epinion, where users can rate positively or negatively not only products, but also other users. Even in social networks where connections solely represent friendships, negative links can still emerge from the analysis of online debates among users.

When the social network is signed, specific challenges arise in both network analysis and learning. On the one hand, novel methods are required to tackle standard tasks (e.g., user clustering, link prediction, targeted advertising/recommendation, analysis of the spreading of diseases in epidemiological models). On the other hand, new problems such as edge sign prediction, which we consider here, naturally emerge. Edge sign prediction is the problem of classifying the positive or negative nature of the links based on the network topology. Prior knowledge of the network topology is often a realistic assumption, for in several situations the discovery of the link sign can be more costly than acquiring the topological information of the network. For instance, when two users of an online social network communicate on a public web page, we immediately detect a link. Yet, the classification of the link sign as positive or negative may require complex techniques.

From the modeling and algorithmic viewpoints, because of the huge amount of available networked data, a major concern in developing learning methods for edge sign prediction is algorithmic scalability. Many successful, yet simple heuristics for edge sign prediction are based on the troll-trust features, i.e., on the fraction of outgoing negative links (trollness) and incoming positive links (trustworthiness) at each node. We study such heuristics by defining a probabilistic generative model for the signs on the directed links of a given network, and show that these heuristics can be understood and analyzed as approximators to the Bayes optimal classifier for our generative model. We also gather empirical evidence supporting our probabilistic

Preliminary work under review: do not distribute.

model by observing that a logistic model trained on trollness and trustworthiness features alone is able to learn weights that, on all datasets considered in our experiments, consistently satisfy the properties predicted by our model.

We then introduce suitable graph transformations defining reductions from edge sign prediction to node sign prediction problems. This opens up the possibility of using the arsenal of known algorithmic techniques developed for node classification. In particular, we show that a Label Propagation algorithm, combined with our reduction, approximates the maximum likelihood estimator of our probabilistic generative model. Experiments on real-world data show the competitiveness of our approach in terms of both prediction performance (especially in the regime when training data are scarce) and scalability.

Finally, we point out that the notions of trollness and trustworthiness naturally define a measure of complexity, or learning bias, for the signed network that can also be used to design *online* (i.e., sequential) learning algorithms for the edge sign prediction problem. The learning bias encourages settings where the nodes in the network have polarized features (e.g., trollness/trustworthiness are either very high or very low). Our online analysis holds under adversarial conditions, namely, without any stochastic assumption on the assignment of signs to the network links.

1.1 Related work

Interest in signed networks can be traced back to the psychological theory of structural balance [4, 12] with its weak version [10]. The advent of online signed social networks has enabled a more thorough and quantitative understanding of that phenomenon. Among the several approaches related to our work, some extend the spectral properties of a graph to the signed case in order to find good embeddings for classification [18, 33]. However, the use of the adjacency matrix usually requires a quadratic running time in the number of nodes, which makes those methods hardly scalable to large graphs. Another approach is based on mining ego networks with SVM. Although this method seems to deliver good results [23], the running time makes it often impractical for large real-world datasets. An alternative approach, based on local features only and proposed in [19], relies on the so-called status theory for directed graphs [11]. Some works in active learning, using a more sophisticated bias based on the correlation clustering (CC) index [6, 5], provide strong theoretical guarantees. However, the bias used there is rather strong, since it assumes the existence of a 2-clustering of the nodes with a small CC index.

Whereas our focus will be on *binary* prediction, researchers have also considered a weighted version of the problem, where edges measure the amount of trust or distrust between two users (e.g., [11, 28, 25]). Other works have also considered versions of the problem where side information related to the network is available to the learning system. For instance, [24] uses the product purchased on Epinion in conjunction with a neural network, [8] identifies trolls by analysing the textual content of their post, and [32] uses SVM to perform transfer learning from one network to another. While many of these approaches have interesting performances, they often require extra information which is not always available (or reliable) and, in addition, may face severe scaling issues. The recent survey [29] contains pointers to many papers on edge sign prediction for signed networks, especially in the Data Mining area. Additional references, more closely related to our work, will be mentioned at the end of Section 4.1.

2 Notation and Preliminaries

In what follows, we let $G = (V, E)$ be a *directed* graph, whose edges $(i, j) \in E$ carry a binary label $y_{i,j} \in \{-1, +1\}$. The edge labeling will sometimes be collectively denoted by the $|V| \times |V|$ matrix $Y = [Y_{i,j}]$, where $Y_{i,j} = y_{i,j}$ if $(i, j) \in E$, and $Y_{i,j} = 0$, otherwise. The corresponding edge-labeled graph will be denoted by $G(Y) = (V, E(Y))$. We use $\mathcal{E}_{\text{in}}(i)$ and $\mathcal{E}_{\text{out}}(i)$ to denote, respectively, the set of edges incoming to and outgoing from node $i \in V$, with $d_{\text{in}}(i) = \left|\mathcal{E}_{\text{in}}(i)\right|$ and $d_{\text{out}}(i) = \left|\mathcal{E}_{\text{out}}(i)\right|$ being the in-degree and the out-degree of i. Moreover, $d^{+}_{\text{in}}(i)$ is the number of edges $(k, i) \in \mathcal{E}_{\text{in}}(i)$ such that $y_{k,i} = +1$. We define $d^{-}_{\text{in}}(i)$, $d^{+}_{\text{out}}(i)$, and $d^{-}_{\text{out}}(i)$ similarly, so that, for instance, $d^{-}_{\text{out}}(i)/d_{\text{out}}(i)$ is the fraction of outgoing edges from node i whose label in $G(Y)$ is -1. We call $tr(i) = d^{-}_{\text{out}}(i)/d_{\text{out}}(i)$ the *trollness* of node i, and $un(i) = d^{-}_{\text{in}}(i)/d_{\text{in}}(i)$ the *untrustworthiness* of node i. Finally, we also use the notation $\mathcal{N}_{\text{in}}(i)$ and $\mathcal{N}_{\text{out}}(i)$ to represent, respectively, the in-neighborhood and the out-neighborhood of node $i \in V$.

Given the directed graph $G = (V, E)$, we define two *edge-to-node reductions* transforming the original graph G into other graphs. As we see later, these reductions are useful in turning the edge sign prediction problem into a *node* sign prediction problem (often called node classification problem), for which many algorithms are indeed available —see, e.g., [3, 34, 13, 14, 7]. Although any node classification method could in principle be used, the reductions we describe next are essentially aimed at preparing the ground for quadratic energy-minimization approaches computed through a *Label Propagation* algorithm (e.g., [34, 2]).

The first reduction, called $G \to G'$, builds an *undirected* graph $G' = (V', E')$ as follows. Each node $i \in V$ has

two copies in V', call them i_{in} and i_{out}. Each directed edge (i,j) in E is associated with one node, call it $e_{i,j}$, in V', along with the two undirected edges $(i_{\text{out}}, e_{i,j})$ and $(e_{i,j}, j_{\text{in}})$. Hence $|V'| = 2|V| + |E|$ and $|E'| = 2|E|$. Moreover, if $G = G(Y)$ is edge labeled, then this labeling transfers to the subset of nodes $e_{i,j} \in V'$, so that G' is a graph $G'(Y) = (V'(Y), E')$ with partially-labeled nodes. The second reduction, called $G \to G''$, builds an *undirected and weighted* graph $G'' = (V'', E'')$. Specifically, we have $V'' \equiv V'$ and $E'' \supset E'$, where the set E'' also includes edges $(i_{\text{out}}, j_{\text{in}})$ for all i and j such that $(i,j) \in E$. The edges in E' have weight 2, whereas the edges in $E'' \setminus E'$ have weight -1. Finally, as in the $G \to G'$ reduction, if $G = G(Y)$ is edge labeled, then this labeling transfers to the subset of nodes $e_{i,j} \in V''$. Graph G', which will not be used in this paper, is an intermediate structure between G and G'' and provides a conceptual link to the standard cutsize measure in node sign classification. Figure 1 illustrates the two reductions.

These reductions are meaningful only if they are able to approximately preserve label *regularity* when moving from edges to nodes. That is, if the edge sign prediction problem is easy for a given $G(Y) = (V, E(Y))$, then the corresponding node sign prediction problems on $G'(Y) = (V'(Y), E')$ and $G''(Y) = (V''(Y), E)$ are also easy, and vice versa. While we could make this argument more quantitative, here we simply observe that if each node in G tends to be either troll or trustworthy, then few labels from the incoming and outgoing edges of each such node are sufficient to predict the labels on the remaining edges in G, and this translates to a small cutsize[1] of $G'(Y)$ over the nodes corresponding to the edges in G (the colored squares in Figure 1 (b)). Again, we would like to point out that these reductions serve two purposes: i. They allow us to use the many algorithms designed for the better studied problem of node sign prediction, and ii. the reduction $G \to G''$ specifically makes the Label Propagation solution approximate the maximum likelihood estimator associated with our generative model (see Section 4.1). Besides, efficient Label Propagation implementations exist that can leverage the sparsity of G''.

We consider two learning settings associated with the problem of edge sign prediction: a batch setting and an online setting. In the batch setting, we assume that a training set of edges E_0 has been drawn uniformly at random *without replacement* from E, we observe the labels in E_0, and we are interested in predicting the sign of the remaining edges $E \setminus E_0$ by making as few prediction mistakes as possible. The specific batch setting we study here assumes that labels are produced by a generative model which we describe in the next section, and our label regularity measure is a quadratic function (denoted by $\Psi^2_{G''}(Y)$ —see Section 6 for a definition), related to this model. $\Psi^2_{G''}(Y)$ is small just when all nodes in G tend to be either troll or trustworthy.

On the other hand, the *online* setting we consider is the standard mistake bound model of online learning [20] where all edge labels are assumed to be generated by an adversary and sequentially presented to the learner according to an arbitrary permutation. For an online learning algorithm A, we are interested in measuring the total number of mistakes $M_A(Y)$ the algorithm makes over $G(Y)$ when the worst possible presentation order of the edge labels in Y is selected by the adversary. Also in the online setting our label regularity measure, denoted here by $\Psi_G(Y)$, is small when nodes in G tend to be either troll or trustworthy. Formally, for fixed G and Y, let $\Psi_{\text{in}}(j, Y) = \min\big\{d^-_{\text{in}}(j), d^+_{\text{in}}(j)\big\}$ and $\Psi_{\text{out}}(i, Y) = \min\big\{d^-_{\text{out}}(i), d^+_{\text{out}}(i)\big\}$. Let also $\Psi_{\text{in}}(Y) = \sum_{j \in V} \Psi_{\text{in}}(j, Y)$ and $\Psi_{\text{out}}(Y) = \sum_{i \in V} \Psi_{\text{out}}(i, Y)$. Then we define $\Psi_G(Y) = \min\big\{\Psi_{\text{in}}(Y), \Psi_{\text{out}}(Y)\big\}$.

The two measures $\Psi^2_{G''}(Y)$ and $\Psi_G(Y)$ are conceptually related, as also empirically confirmed by our experiments (see Table 2 in Section 6).

3 Generative Model for Edge Labels

We now define the stochastic generative model for edge labels we use in the batch learning setting. Given the graph $G = (V, E)$, let the label $y_{i,j} \in \{-1, +1\}$ of directed edge $(i,j) \in E$ be generated as follows. Each node $i \in V$ is endowed with two latent parameters $p_i, q_i \in [0,1]$, which we assume to be generated, for each node i, by an independent draw from a fixed but unknown joint prior distribution $\mu(p,q)$ over $[0,1]^2$. Each label $y_{i,j} \in \{-1, +1\}$ is then generated by an independent draw from the mixture of p_i and q_j, $\mathbb{P}\big(y_{i,j} = 1\big) = \frac{p_i + q_j}{2}$. The basic intuition is that the nature $y_{i,j}$ of a relationship $i \to j$ is stochastically determined by a mixture between how much node i tends to like other people (p_i) and how much node j tends to be liked by other people (q_j). In a certain sense, $1 - tr(i)$ is the empirical counterpart to p_i, and $1 - un(j)$ is the empirical counterpart to q_j.[2] Notice that the Bayes optimal prediction for $y_{i,j}$ is $y^*(i,j) = \textsc{sgn}\big(\eta(i,j) - \frac{1}{2}\big)$, where $\eta(i,j) = \mathbb{P}\big(y_{i,j} = 1\big)$. Moreover, the probability of drawing at random a $+1$-labeled edge from $\mathcal{E}_{\text{out}}(i)$

[1] Recall that the cutsize of an undirected node-labeled graph $G'(Y)$ is the number of edges in G' connecting nodes having mismatching labels.

[2] One might view our model as reminiscent of standard models for link generation in social network analysis, like the classical p_1 model from [15]. Yet, the similarity falls short, for all these models aim at representing the likelihood of the network topology, rather than the probability of edge signs, once the topology is *given*.

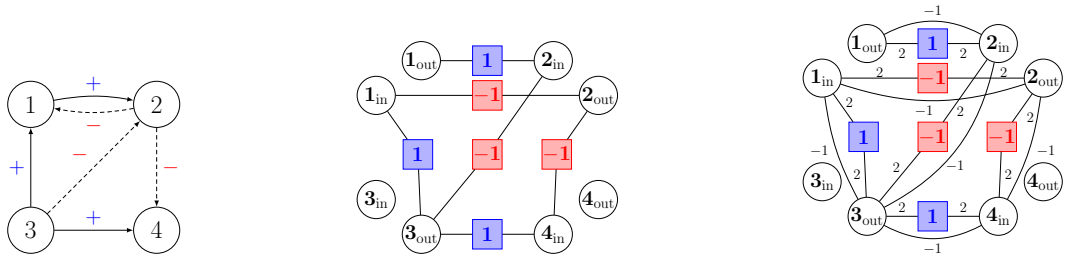

Figure 1: **(a)** A directed edge-labeled graph G. **(b)** Its corresponding graph G' resulting from the $G \to G'$ reduction. The square nodes in G' correspond to the edges in G, and carry the same labels as their corresponding edges. On the other hand, the $2|V|$ circle nodes in G' are unlabeled. Observe that some nodes in G' are isolated (and thus unimportant); these are exactly the nodes in G' corresponding to the nodes having in G no outgoing or no incoming edges —see, e.g., nodes 3 and 4 in G. **(c)** The weighted graph resulting from the $G \to G''$ reduction.

and the probability of drawing at random a +1-labeled edge from $\mathcal{E}_{\text{in}}(j)$ are respectively equal to

$$\frac{1}{2}\left(p_i + \frac{1}{d_{\text{out}}(i)} \sum_{j \in \mathcal{N}_{\text{out}}(i)} q_j\right) \quad \text{and} \quad \frac{1}{2}\left(q_j + \frac{1}{d_{\text{in}}(j)} \sum_{i \in \mathcal{N}_{\text{in}}(j)} p_i\right). \tag{1}$$

4 Algorithms in the Batch Setting

Given $G(Y) = (V, E(Y))$, we have at our disposal a training set E_0 of labeled edges from $E(Y)$, our goal being that of building a predictive model for the labels of the remaining edges.

Our first algorithm is an approximation to the Bayes optimal predictor $y^*(i,j)$. Let us denote by $\widehat{tr}(i)$ and $\widehat{un}(i)$ the trollness and the untrustworthiness of node i when both are computed on the subgraph induced by the training edges. We now design and analyze an edge classifier of the form

$$\textsc{sgn}\Big(\big(1 - \widehat{tr}(i)\big) + \big(1 - \widehat{un}(j)\big) - \tfrac{1}{2} - \tau\Big)\,, \tag{2}$$

where $\tau \geq 0$ is the only parameter to be trained. Despite its simplicity, this classifier works reasonably well in practice, as demonstrated by our experiments (see Section 6). Moreover, unlike previous edge sign prediction methods for directed graphs, our classifier comes with a rigorous theoretical motivation, since it approximates the Bayes optimal classifier $y^*(i,j)$ with respect to the generative model defined in Section 3. It is important to point out that when we use $1 - \widehat{tr}(i)$ and $1 - \widehat{un}(j)$ to estimate p_i and q_j, an additive bias shows up due to (1). This motivates the need of a threshold parameter τ to cancel this bias. Yet, the presence of a prior distribution $\mu(p,q)$ ensures that this bias is the same for all edges $(i,j) \in E$.

Our algorithm works under the assumption that for given parameters Q (a positive integer) and $\alpha \in (0,1)$ there exists a set[3] $E_L \subseteq E$ of size $\frac{2Q}{\alpha}$ where each vertex $i \in V$ appearing as an endpoint of some edge in E_L occurs at most once as origin —i.e., (i,j)— and at most once as destination —i.e., (j,i). Moreover, we assume E_0 has been drawn from E at random *without* replacement, with $m = |E_0| = \alpha\,|E|$. The algorithm performs the following steps:

1. For each $j \in V$, let $\widehat{un}(j) = \widehat{d}_{\text{in}}^{-}(j)/\widehat{d}_{\text{in}}(j)$, i.e., the fraction of negative edges found in $\mathcal{E}_{\text{in}}(j) \cap E_0$.

2. For each $i \in V$, let $\widehat{tr}(i) = \widehat{d}_{\text{out}}^{-}(i)/\widehat{d}_{\text{out}}(i)$, i.e., the fraction of negative edges found in $\mathcal{E}_{\text{out}}(i) \cap E_0$.

3. Let $\widehat{\tau}$ be the fraction of positive edges in $E_L \cap E_0$.

4. Any remaining edge $(i,j) \in E \setminus E_0$ is predicted as $\widehat{y}(i,j) = \textsc{sgn}\Big(\big(1 - \widehat{tr}(i)\big) + \big(1 - \widehat{un}(j)\big) - \frac{1}{2} - \widehat{\tau}\Big)$.

The next result[4] shows that if the graph is not too sparse, then the above algorithm can approximate the Bayes optimal predictor on nodes whose in-degree and out-degree is not too small.

Theorem 1. *Let $G(Y) = (V, E(Y))$ be a directed graph with labels on the edges generated according to the model in Section 3. If the algorithm is run with parameter $Q = \Omega(\ln|V|)$, and $\alpha \in (0,1)$ such that the above assumptions are satisfied, then $\widehat{y}(i,j) = y^*(i,j)$ holds with high probability simultaneously for all test edges $(i,j) \in E$ such that $d_{\text{out}}(i), d_{\text{in}}(j) = \Omega(\ln|V|)$, and $\eta(i,j) = \mathbb{P}(y_{i,j} = 1)$ is bounded away from $\frac{1}{2}$.*

From this algorithm we derive the BLC algorithm we use in our experiments of Section 6.

4.1 Approximation to Maximum Likelihood via Label Propagation

For simplicity, assume the joint prior distribution $\mu(p,q)$ is uniform over $[0,1]^2$ with independent

[3] This set is needed to find an estimate $\widehat{\tau}$ of τ in (2), see Step 3 of the algorithm. It can be any set of directed paths/cycles in G that are pairwise vertex-disjoint.

[4] Due to space limitations, all proofs are in the supplementary material.

marginals, and suppose that we draw at random without replacement the training set $E_0 = \big((i_1,j_1), y_{i_1,j_1}\big), \big((i_2,j_2), y_{i_2,j_2}\big), \ldots, \big((i_m,j_m), y_{i_m,j_m}\big)$, with $m = |E_0|$. Then a reasonable approach to approximate $y^*(i,j)$ would be to resort to a maximum likelihood estimator of the parameters $\{p_i, q_i\}_{i=1}^{|V|}$ based on E_0. Working out the math (details in the supplementary material), one can easily see that the gradient of the log-likelihood function w.r.t. $\{p_i, q_i\}_{i=1}^{|V|}$ satisfies

$$\frac{\partial \log \mathbb{P}\left(E_0 \,\middle|\, \{p_i,q_i\}_{i=1}^{|V|}\right)}{\partial p_\ell} = \sum_{k=1}^{m} \frac{\mathbb{I}\{i_k = \ell, y_{\ell,j_k} = +1\}}{p_\ell + q_{j_k}} - \sum_{k=1}^{m} \frac{\mathbb{I}\{i_k = \ell, y_{\ell,j_k} = -1\}}{2 - p_\ell - q_{j_k}}, \tag{3}$$

$$\frac{\partial \log \mathbb{P}\left(E_0 \,\middle|\, \{p_i,q_i\}_{i=1}^{|V|}\right)}{\partial q_\ell} = \sum_{k=1}^{m} \frac{\mathbb{I}\{j_k = \ell, y_{i_k,\ell} = +1\}}{p_{i_k} + q_\ell} - \sum_{k=1}^{m} \frac{\mathbb{I}\{j_k = \ell, y_{i_k,\ell} = -1\}}{2 - p_{i_k} - q_\ell}, \tag{4}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function of the event at argument. Unfortunately, equating (3) and (4) to zero, and solving for parameters $\{p_i, q_i\}_{i=1}^{|V|}$ gives rise to a hard set of nonlinear equations. Moreover, it may well be the case that some such parameters never occur in these equations. This will happen precisely whenever $\mathcal{E}_{\text{out}}(i)$ or $\mathcal{E}_{\text{in}}(j)$ are not represented in E_0 for some $i,j \in V$. A more viable and computation-friendly alternative is to approximate the nonlinear equations resulting from (3) and (4) by the following set of linear equations, one for each $\ell \in V$:

$$\sum_{k=1}^{m} \mathbb{I}\{i_k = \ell, y_{\ell,j_k} = +1\}\,(2 - p_\ell - q_{j_k}) = \sum_{k=1}^{m} \mathbb{I}\{i_k = \ell, y_{\ell,j_k} = -1\}\,(p_\ell + q_{j_k})$$

$$\sum_{k=1}^{m} \mathbb{I}\{j_k = \ell, y_{i_k,\ell} = +1\}\,(2 - p_{i_k} - q_\ell) = \sum_{k=1}^{m} \mathbb{I}\{j_k = \ell, y_{i_k,\ell} = -1\}\,(p_{i_k} + q_\ell)\ .$$

The solution to these equations are precisely the points where the gradient w.r.t. $(\boldsymbol{p},\boldsymbol{q}) = \{p_i, q_i\}_{i=1}^{|V|}$ of the quadratic function

$$f_{E_0}(\boldsymbol{p},\boldsymbol{q}) = \sum_{(i,j)\in E_0} \left(\frac{1 + y_{i,j}}{2} - \frac{p_i + q_j}{2}\right)^2$$

vanishes. We follow a label propagation approach by adding to f_{E_0} the corresponding test set function $f_{E\setminus E_0}$, and treat the sum of the two as the function to be minimized during training w.r.t. both $(\boldsymbol{p},\boldsymbol{q})$ and all $y_{i,j} \in [-1,+1]$ for $(i,j) \in E \setminus E_0$, i.e.,

$$\min_{(\boldsymbol{p},\boldsymbol{q}),\, y_{i,j}\in[-1,+1],\, (i,j)\in E\setminus E_0} \big(f_{E_0}(\boldsymbol{p},\boldsymbol{q}) + f_{E\setminus E_0}(\boldsymbol{p},\boldsymbol{q})\big)\ . \tag{5}$$

Binary ± 1 predictions on the test set $E \setminus E_0$ are then obtained by thresholding the obtained values $y_{i,j}$ at 0.

The above problem can be approximately solved by standard label propagation [34] on graph G'' obtained through the $G \to G''$ reduction of Section 2. However, because of the presence of negative edge weights in G'', we first have to symmetrize variables $p_i, q_i, y_{i,j}$ so as they all lie in the interval $[-1,+1]$. After this step, one can see that, once we get back to the original variables, label propagation computes the harmonic solution minimizing the function

$$\widehat{f}(\boldsymbol{p},\boldsymbol{q}, y_{i,j\,(i,j)\in E\setminus E_0}) = f_{E_0}(\boldsymbol{p},\boldsymbol{q}) + f_{E\setminus E_0}(\boldsymbol{p},\boldsymbol{q}) + \frac{1}{2}\sum_{i\in V}\left(d_{\text{out}}(i)\Big(p_i - \frac{1}{2}\Big)^2 + d_{\text{in}}(i)\Big(q_i - \frac{1}{2}\Big)^2\right).$$

The function $\widehat{f}$ is thus a regularized version of the target function $f_{E_0} + f_{E\setminus E_0}$ in (5), where the regularization term tries to enforce the extra constraint that whenever a node i has a high out-degree then the corresponding p_i should be close to $1/2$. Thus, on any edge (i,j) departing from i, the Bayes optimal predictor $y^*(i,j) = \textsc{sgn}(p_i + q_j - 1)$ will mainly depend on q_j being larger or smaller than $\frac{1}{2}$ (assuming j has small in-degree). Similarly, if i has a high in-degree, then the corresponding q_i should be close to $1/2$ implying that on any edge (j,i) arriving at i the Bayes optimal predictor $y^*(j,i)$ will mainly depend on p_j (assuming j has small out-degree). Put differently, a node having a huge out-neighborhood makes each outgoing edge "count less" than a node having only a small number of outgoing edges, and similarly for in-neighborhoods. The label propagation algorithm operating on G'' does so (see again Figure 1 (c)) by iteratively updating as follows:

$$p_i \leftarrow \frac{-\sum_{j\in\mathcal{N}_{\text{out}}(i)} q_j + \sum_{j\in\mathcal{N}_{\text{out}}(i)}(1 + y_{i,j})}{3\, d_{\text{out}}(i)} \qquad \forall i \in V$$

$$q_j \leftarrow \frac{-\sum_{i\in\mathcal{N}_{\text{in}}(j)} p_i + \sum_{i\in\mathcal{N}_{\text{in}}(j)}(1 + y_{i,j})}{3\, d_{\text{in}}(j)} \qquad \forall j \in V$$

$$y_{i,j} \leftarrow \frac{p_i + q_j}{2} \qquad \forall (i,j) \in E \setminus E_0\ .$$

The algorithm is guaranteed to converge [34] to the minimizer of $\widehat{f}$. Notice that the presence of negative weights on the edges of G'' does not prevent label propagation from converging. This is the algorithm we will be championing in our experiments of Section 6.

Further related work. The vast majority of existing edge sign prediction algorithms for directed graphs are based on the computation of local features of

the graph. These features are evaluated on the subgraph induced by the training edges, and the resulting values are used to train a supervised classification algorithm (e.g., logistic regression). The most basic set of local features used to classify a given edge (i, j) are defined by $d^+_{\text{in}}(j), d^-_{\text{in}}(j), d^+_{\text{out}}(i), d^-_{\text{out}}(i)$ computed over the training set E_0, and by the embeddedness coefficient $\left|\mathcal{E}_{\text{out}}(i) \cap \mathcal{E}_{\text{in}}(j)\right|$. In turn, these can be used to define more complicated features, such as $\frac{d^+_{\text{in}}(j)+|E^+|U_{\text{in}}(j)}{d_{\text{in}}(j)+U_{\text{in}}(j)}$ and $\frac{d^+_{\text{out}}(i)+p^+U_{\text{out}}(i)}{d_{\text{out}}(i)+U_{\text{out}}(i)}$ introduced in [27], together with their negative counterparts, where $|E^+|$ is the overall fraction of positive edges, and $U_{\text{in}}(j), U_{\text{out}}(i)$ are, respectively, the number of test edges outgoing from i and the number of test edges incoming to j. Other types of features are derived from social status theory (e.g., [19]), and involve the so-called triads; namely, the triangles formed by (i, j) together with (i, w) and (w, j) for any $w \in \mathcal{N}_{\text{out}}(i) \cap \mathcal{N}_{\text{in}}(j)$. A third group of features is based on node ranking scores. These scores are computed using a variery of methods, including Prestige [35], exponential ranking [30], PageTrust [16], Bias and Deserve [22], TrollTrust [31], and generalizations of PageRank and HITS to signed networks [26]. Examples of features using such scores are *reputation* and *optimism* [26], defined for a node i by $\frac{\sum_{j\in\mathcal{N}_{\text{in}}(i)} y_{j,i}\sigma(j)}{\sum_{j\in\mathcal{N}_{\text{in}}(i)} \sigma(j)}$ and $\frac{\sum_{j\in\mathcal{N}_{\text{out}}(i)} Y_{i,j}\sigma(j)}{\sum_{j\in\mathcal{N}_{\text{out}}(i)} \sigma(j)}$, where $\sigma(j)$ is the ranking score assigned to node j. Some of these algorithms will be used as representative competitors in our experimental study of Section 6.

5 Algorithms in the Online Setting

For the online scenario, we have the following result.

Theorem 2. *There exists a randomized online prediction algorithm A whose expected number of mistakes satisfies $\mathbb{E}M_A(Y) = \Psi_G(Y) + O\left(\sqrt{|V|\Psi_G(Y)} + |V|\right)$ on any edge-labeled graph $G(Y) = (V, E(Y))$.*

The algorithm used in Theorem 2 is a combination of randomized Weighted Majority instances. Details are reported in the supplementary material. We complement the above result by providing a mistake lower bound. Like Theorem 2, the following result holds for all graphs, and for all label irregularity levels $\Psi_G(Y)$.

Theorem 3. *Given any edge-labeled graph $G(Y) = (V, E(Y))$ and any integer $K \le \left\lfloor \frac{|E|}{2} \right\rfloor$, a randomized labeling $Y \in \{-1, +1\}^{|E|}$ exists such that $\Psi_G(Y) \le K$, and the expected number of mistakes that any online algorithm A can be forced to make satisfies $\mathbb{E}M_A(Y) \ge \frac{K}{2}$. Moreover, as $\frac{K}{|E|} \to 0$ then $\mathbb{E}M_A(Y) = K$.*

6 Experimental Analysis

In this section, we evaluate our edge sign classification methods on representative real-world datasets of varying density and label regularity, showing that our methods compete well against existing approaches in terms of both predictive and computational performance. We are especially interested in small training set regimes, and have restricted our comparison to the batch learning scenario since all competing methods we are aware of have been developed in that setting only.

Datasets. We considered five real-world classification datasets. The first three are directed signed social networks widely used as benchmarks for this task (e.g.,[19, 26, 31]): In Wikipedia, there is an edge from user i to user j if j applies for an admin position and i votes for or against that promotion. In Slashdot, a news sharing and commenting website, member i can tag other members j as friends or foes. Finally, in Epinion, an online shopping website, user j reviews products and, based on these reviews, another user i can display whether he considers j to be reliable or not. In addition to these three datasets, we considered two other signed social networks where the signs are inferred automatically. In Wik. Edits [21], an edge from Wikipedia user i to user j indicates whether they edited the same article in a constructive manner or not.[5] Finally, in the Citations [17] network, an author i cites another author j by either endorsing or criticizing j's work. The edge sign is derived by classifying the citation sentiment with a simple, yet powerful, keyword-based technique using a list of positive and negative words. See [17] for more details.[6]

Table 1 summarizes statistics for these datasets. We note that most edge labels are positive. Hence, test set accuracy is not an appropriate measure of prediction performance. We instead evaluated our performance using the so-called Matthews Correlation Coefficient (MCC) (e.g., [1]), defined as

$$\text{MCC} = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}\,.$$

MCC combines all the four quantities found in a binary confusion matrix (*t*rue *p*ositive, *t*rue *n*egative, *f*alse *p*ositive and *f*alse *n*egative) into a single metric which ranges from -1 (when all predictions are incorrect) to $+1$ (when all predictions are correct).

Although the semantics of the edge signs is not the same across these networks, we can see from Table 1 that our generative model essentially fits all of them. Specifically, the last two columns of the table report the rate of label (ir)regularity, as measured by $\Psi^2_{G''}(Y)/|E|$ (second-last column) and $\Psi_G(Y)/|E|$ (last column),

[5] This is the KONECT version of the "Wikisigned" dataset, from which we removed self-loops.

[6] We again removed self-loops and merged multi-edges which are all of the same sign.

Table 1: Dataset properties. The 5th column gives the fraction of positive labels. The last two columns provide two different measures of label regularity —see main text.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\frac{\lvert E\rvert}{\lvert V\rvert}$	$\frac{\lvert E^+\rvert}{\lvert E\rvert}$	$\frac{\Psi^2_{G''}(Y)}{\lvert E\rvert}$	$\frac{\Psi_G(Y)}{\lvert E\rvert}$
Citations	4,831	39,452	8.1	72.33%	.076	.191
Wikipedia	7,114	103,108	14.5	78.79%	.063	.142
Slashdot	82,140	549,202	6.7	77.40%	.059	.143
Wik. Edits	138,587	740,106	5.3	87.89%	.034	.086
Epinion	131,580	840,799	6.4	85.29%	.031	.074

where

$$\Psi^2_{G''}(Y) = \min_{(\boldsymbol{p},\boldsymbol{q})} \left(f_{E_0}(\boldsymbol{p},\boldsymbol{q}) + f_{E\setminus E_0}(\boldsymbol{p},\boldsymbol{q}) \right) ,$$

f_{E_0} and $f_{E\setminus E_0}$ being the quadratic criterions of Section 4.1, viewed as functions of both $(\boldsymbol{p},\boldsymbol{q})$, and $y_{i,j}$, and $\Psi_G(Y)$ is the label regularity measure adopted in the online setting, as defined in Section 2. It is reasonable to expect that higher label irregularity corresponds to lower prediction performance. This trend is in fact confirmed by our experimental findings: whereas Epinion tends to be easy, Citations tends to be hard, and this holds for all algorithms we tested, even if they do not explicitly comply with our inductive bias principles. Moreover, $\Psi^2_{G''}(Y)/|E|$ tends to be proportional to $\Psi_G(Y)/|E|$ across datasets, hence confirming the anticipated connection between the two regularity measures.

Algorithms and parameter tuning. We compared the following algorithms:

1. The label propagation algorithm of Section 4.1 (referred to as L. Prop.). The actual binarizing threshold was set by cross-validation on the training set.

2. The algorithm analyzed at the beginning of Section 4, which we call blc(tr, un) (Bayes Learning Classifier based on *tr*ollness and *un*trustworthiness). After computing $\widehat{tr}(i)$ and $\widehat{un}(i)$ on training set E_0 for all $i \in V$ (or setting those values to $\frac{1}{2}$ in case there is no outgoing or incoming edges for some node), we use Eq. (2) and estimate τ on E_0.

3. A logistic regression model where each edge (i,j) is associated with the features $[1-\widehat{tr}(i), 1-\widehat{un}(j)]$ computed again on E_0 (we call this method LogReg). Best binary thresholding is again computed on E_0. Experimenting with this logistic model serves to support the claim we made in the introduction that our generative model in Section 3 is a good fit for the data.

4. The solution obtained by directly solving the unregularized problem (5) through a fast constrained minimization algorithm (referred to as Unreg.). Again, the actual binarizing threshold was set by cross-validation on the training set.[7]

5. The matrix completion method from [9] based on LowRank matrix factorization. Since the authors showed their method to be robust to the choice of the rank parameter k, we picked $k = 7$ in our experiments.

6. A logistic regression model built on 16 Triads features derived from status theory [19].

7. The PageRank-inspired algorithm from [31], where a recursive notion of trollness is computed by solving a suitable set of nonlinear equations through an iterative method, and then used to assign ranking scores to nodes, from which (un)trustworthiness features are finally extracted for each edge. We call this method RankNodes. As for hyperparameter tuning (β and λ_1 in [31]), we closely followed the authors' suggestion of doing cross validation.

8. The last competitor is the logistic regression model whose features have been build according to [27]. We call this method Bayesian.

The above methods can be roughly divided into *local* and *global* methods. A local method hinges on building local predictive features, based on neighborhoods: blc(tr, un), LogReg, 16 Triads, and Bayesian essentially fall into this category. The remaining methods are global in that their features are designed to depend on global properties of the graph topology.

Results. Our main results are summarized in Table 2, where we report MCC test set performance after training on sets of varying size (from 3% to 25%). Results have been averaged over 12 repetitions. Because scalability is a major concern when training on sizeable datasets, we also give an idea of relative training times (in milliseconds) by reporting the time it took to train a single run of each algorithm on a training set of size[8] 15% of $|E|$, and then predict on the test set. Though our experiments are not conclusive, some trends can be readily spotted:

1. Global methods tend to outperform local methods in terms of prediction performance, but are also significantly (or even much) slower (running times can differ by as much as three orders of magnitude). This is not

[7] We have also tried to minimize (5) by removing the $[-1,+1]$ constraints, but got similar MCC results as the ones we report for Unreg.

[8] Comparison of training time performances is fair since all algorithms have been carefully implemented using the same stack of Python libraries, and run on the same machine (16 Xeon cores and 192Gb Ram).

Table 2: MCC with increasing training set size, with one standard deviation over 12 random sampling of E_0. The last four columns refer to the methods we took from the literature. For the sake of readability, we multiplied all MCC values by 100. The best number in each row is highlighted in **bold brown** and the second one in *italic red*. If the difference is statistically significant (p-value of a paired Student's t-test less than 0.005), the best score is underlined. The "time" rows contain the time taken to train on a 15% training set.

	$\frac{\lvert E_0\rvert}{\lvert E\rvert}$	L. Prop.	BLC(tr, un)	LogReg	Unreg.	LowRank	16 Triads	RankNodes	Bayesian
Citations	3%	**19.07** $\pm$ 1.38	14.97 $\pm$ 0.72	*15.05* $\pm$ 0.73	10.88 $\pm$ 0.83	8.89 $\pm$ 1.44	7.54 $\pm$ 1.42	12.48 $\pm$ 1.18	11.06 $\pm$ 1.64
	9%	**29.64** $\pm$ 0.74	*25.96* $\pm$ 0.61	25.90 $\pm$ 0.67	23.51 $\pm$ 0.72	17.05 $\pm$ 0.70	16.13 $\pm$ 0.64	23.84 $\pm$ 0.91	23.44 $\pm$ 0.81
	15%	**35.66** $\pm$ 0.68	*32.87* $\pm$ 0.58	32.79 $\pm$ 0.60	31.39 $\pm$ 0.75	22.58 $\pm$ 0.53	21.55 $\pm$ 0.91	30.60 $\pm$ 0.87	31.71 $\pm$ 0.99
	20%	**38.67** $\pm$ 0.48	*36.94* $\pm$ 0.51	36.86 $\pm$ 0.48	35.47 $\pm$ 0.41	25.80 $\pm$ 0.94	24.27 $\pm$ 0.56	35.01 $\pm$ 0.83	36.13 $\pm$ 0.75
	25%	**41.05** $\pm$ 0.73	39.83 $\pm$ 0.58	39.76 $\pm$ 0.59	38.48 $\pm$ 0.55	29.67 $\pm$ 0.78	26.85 $\pm$ 0.87	38.06 $\pm$ 0.86	*40.34* $\pm$ 0.94
	time	19.6	0.6	2.6	2835	3279	6.2	155	4813
Wikipedia	3%	**34.08** $\pm$ 1.03	31.65 $\pm$ 0.99	*32.08* $\pm$ 0.94	28.50 $\pm$ 0.92	19.73 $\pm$ 0.99	3.98 $\pm$ 0.84	27.43 $\pm$ 0.99	20.11 $\pm$ 1.14
	9%	**45.80** $\pm$ 0.43	44.92 $\pm$ 0.48	*45.72* $\pm$ 0.42	43.02 $\pm$ 0.52	30.24 $\pm$ 0.54	23.50 $\pm$ 1.66	41.36 $\pm$ 0.66	38.05 $\pm$ 0.93
	15%	*50.49* $\pm$ 0.33	49.89 $\pm$ 0.40	**50.87** $\pm$ 0.36	49.08 $\pm$ 0.33	35.77 $\pm$ 0.58	33.64 $\pm$ 0.83	48.50 $\pm$ 0.47	47.07 $\pm$ 0.38
	20%	*52.74* $\pm$ 0.31	52.24 $\pm$ 0.49	**53.13** $\pm$ 0.27	51.79 $\pm$ 0.35	37.90 $\pm$ 0.27	38.41 $\pm$ 0.53	51.49 $\pm$ 0.43	50.54 $\pm$ 0.39
	25%	*54.00* $\pm$ 0.63	53.42 $\pm$ 0.59	**54.26** $\pm$ 0.37	53.31 $\pm$ 0.37	40.16 $\pm$ 0.57	41.34 $\pm$ 1.07	53.30 $\pm$ 0.37	52.92 $\pm$ 0.48
	time	41.9	1.6	6.0	10629	8523	14.8	249	12507
Slashdot	3%	*36.58* $\pm$ 0.43	31.71 $\pm$ 0.59	32.29 $\pm$ 0.47	29.14 $\pm$ 0.56	34.39 $\pm$ 0.56	20.57 $\pm$ 0.93	**42.98** $\pm$ 0.56	24.96 $\pm$ 0.43
	9%	*45.71* $\pm$ 0.28	41.06 $\pm$ 0.38	42.19 $\pm$ 0.18	39.60 $\pm$ 0.25	39.39 $\pm$ 0.34	37.92 $\pm$ 1.57	**47.40** $\pm$ 0.42	36.83 $\pm$ 0.36
	15%	**49.62** $\pm$ 0.22	45.42 $\pm$ 0.36	46.42 $\pm$ 0.16	45.56 $\pm$ 0.19	41.21 $\pm$ 0.19	45.88 $\pm$ 1.01	*48.75* $\pm$ 0.71	43.47 $\pm$ 0.16
	20%	*51.88* $\pm$ 0.24	47.78 $\pm$ 0.25	48.66 $\pm$ 0.10	48.10 $\pm$ 0.30	42.74 $\pm$ 0.44	48.79 $\pm$ 0.57	**52.10** $\pm$ 0.33	46.89 $\pm$ 0.27
	25%	*53.12* $\pm$ 0.20	49.39 $\pm$ 0.24	50.22 $\pm$ 0.12	50.11 $\pm$ 0.20	44.24 $\pm$ 0.44	50.62 $\pm$ 0.53	**53.29** $\pm$ 0.22	49.42 $\pm$ 0.22
	time	677	8.3	32.8	78537	69988	131	2441	68085
Epinion	3%	*51.43* $\pm$ 0.29	41.42 $\pm$ 0.84	43.41 $\pm$ 0.34	36.56 $\pm$ 0.49	37.01 $\pm$ 0.57	31.71 $\pm$ 6.20	**52.33** $\pm$ 0.42	31.25 $\pm$ 0.80
	9%	*58.48* $\pm$ 0.26	53.20 $\pm$ 0.62	54.80 $\pm$ 0.12	51.58 $\pm$ 0.36	43.81 $\pm$ 0.50	50.43 $\pm$ 2.07	**60.06** $\pm$ 0.46	48.06 $\pm$ 0.41
	15%	*61.47* $\pm$ 0.21	57.63 $\pm$ 0.45	59.25 $\pm$ 0.17	58.80 $\pm$ 0.32	48.24 $\pm$ 0.58	54.64 $\pm$ 1.62	**62.69** $\pm$ 0.21	56.94 $\pm$ 0.65
	20%	*63.17* $\pm$ 0.13	60.15 $\pm$ 0.40	61.45 $\pm$ 0.17	61.86 $\pm$ 0.13	52.21 $\pm$ 0.37	57.27 $\pm$ 1.42	**64.10** $\pm$ 0.12	61.18 $\pm$ 0.45
	25%	64.05 $\pm$ 0.20	61.88 $\pm$ 0.38	62.89 $\pm$ 0.12	63.42 $\pm$ 0.14	54.68 $\pm$ 0.62	58.42 $\pm$ 1.59	**65.40** $\pm$ 0.85	*64.59* $\pm$ 0.30
	time	1329	10.1	54.0	143881	127654	209	3174	104305
Wik. Edits	3%	**33.88** $\pm$ 0.54	*26.02* $\pm$ 0.80	26.02 $\pm$ 0.42	15.13 $\pm$ 0.40	19.93 $\pm$ 0.56	0.86 $\pm$ 0.11	23.50 $\pm$ 0.54	20.10 $\pm$ 0.60
	9%	**38.62** $\pm$ 0.52	35.10 $\pm$ 0.17	*35.29* $\pm$ 0.11	28.23 $\pm$ 0.33	26.61 $\pm$ 0.37	9.99 $\pm$ 0.93	33.22 $\pm$ 0.29	33.86 $\pm$ 0.24
	15%	*39.08* $\pm$ 0.55	37.77 $\pm$ 0.22	38.27 $\pm$ 0.19	33.61 $\pm$ 0.11	30.05 $\pm$ 0.29	17.95 $\pm$ 0.92	36.88 $\pm$ 0.32	**40.00** $\pm$ 0.26
	20%	39.04 $\pm$ 0.69	38.88 $\pm$ 0.36	*39.55* $\pm$ 0.11	35.04 $\pm$ 0.17	32.17 $\pm$ 0.31	21.44 $\pm$ 0.67	38.60 $\pm$ 0.31	**43.32** $\pm$ 0.22
	25%	38.90 $\pm$ 0.45	39.41 $\pm$ 0.16	*40.44* $\pm$ 0.14	36.18 $\pm$ 0.20	33.94 $\pm$ 0.74	23.41 $\pm$ 0.41	39.75 $\pm$ 0.32	**45.76** $\pm$ 0.29
	time	927	9.6	46.8	219109	129460	177	3890	92719

surprising, and is in line with previous experimental findings (e.g., [26, 31]). Bayesian looks like an exception to this rule, but its running time is indeed in the same ballpark as global methods.

2. L. Prop. always ranks first or at least second in this comparison when MCC is considered. On top of it, L. Prop. is fastest among the global methods (one or even two orders of magnitude faster), thereby showing the benefit of our approach to edge sign prediction.

3. The regularized solution computed by L. Prop. is always better than the unregularized one computed by Unreg. in terms of both MCC and running time.

4. As claimed in the introduction, our Bayes approximator BLC(tr, un) closely mirrors in performance the more involved LogReg model. In fact, supporting our generative model of Section 3, the logistic regression weights for features $1 - \widehat{tr}(i)$ and $1 - \widehat{un}(j)$ are almost equal (see Table 2 in the supplementary material), thereby suggesting that predictor (2), derived from the theoretical results at the beginning of Section 4, is *also* the best logistic model based on trollness and untrustworthiness.

7 Conclusions and Ongoing Research

We have studied the edge sign prediction problem in directed graphs in both batch and online learning settings. In both cases, the underlying modeling assumption hinges on the trollness and (un)trustworthiness predictive features. We have introduced a simple generative model for the edge labels that serves both as a theoretical justification for many successful heuristic methods to edge sign prediction and as a way to craft this problem as a node sign prediction problem to be efficiently tackled by standard Label Propagation algorithms. We then observed that the notion of trollness and trustworthiness naturally lend themselves to define a notion of edge sign regularity that can also be analyzed in an (adversarial) online setting. In this setting, we have provided upper and (almost matching) lower bounds on the expected number of prediction mistakes.

We then complemented the above with experiments on five real-world datasets in the small training set regime. Two interesting conclusions from our experiments are: i. Our generative model is robust, for it produces Bayes optimal predictors which tend to be empirically best also within the larger set of models that includes all logistic regressors based on trollness and trustworthiness alone; ii. our methods are in practice either strictly better than their competitors in terms of prediction quality or, when they are not, they are faster.

We are currently engaged in gathering more experimental evidence, as well as in extending our approach so as to incorporate further predictive features (e.g., side information, when available).

References

[1] P. Baldi, S. Brunak, Y. Chauvin, C.A. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 5, 16, pp. 412–424, 2000.

[2] Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. In *Semi-Supervised Learning*, 193–216. MIT Press, 2006.

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *18th ICML*, pages 19–26. Morgan Kaufmann, 2001.

[4] D. Cartwright and F. Harary. Structural balance: a generalization of Heider's theory. *Psychological review*, 63(5):277, 1956.

[5] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. A correlation clustering approach to link classification in signed networks. In *25th COLT*, 2012.

[6] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. A linear time active learning algorithm for link classification. In *NIPS 25*, 2012.

[7] N. Cesa-Bianchi, C. Gentile, F. Vitale, G. Zappella. Random spanning trees and the prediction of weighted graphs. *JMLR*, 14, pp. 1251–1284.

[8] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec. Antisocial behavior in online discussion communities. In *Intl AAAI Conf. on Web and Social Media*, 2015.

[9] K. Chiang, C. Hsieh, N. Natarajan, I. Dhillon, and A. Tewari. Prediction and Clustering in Signed Networks: A Local to Global Perspective. *JMLR*, 15:1177–1213, 2014.

[10] J.A. Davis. Clustering and structural balance in graphs. *Human relations*, 1967.

[11] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *13th WWW*, pp. 403–412, 2004.

[12] F. Heider. *The psychology of interpersonal relations*. 1958.

[13] M. Herbster and M. Pontil. Prediction on a graph with the Perceptron. In *NIPS 21*, pp. 577–584. MIT Press, 2007.

[14] M. Herbster, G. Lever, and M. Pontil. Online prediction on large diameter graphs. In *NIPS 22*, pp. 649–656. MIT Press, 2009.

[15] P. W. Holland and S. Leinhardt. An Exponential Family of Probability Distributions for Directed Graphs, *JASA*, 76, pp. 33–65, 1981.

[16] C. De Kerchove and P. Van Dooren. The pagetrust algorithm: How to rank web pages when negative links are allowed? In *SDM*, pp. 346–352. SIAM, 2008.

[17] S. Kumar. Structure and Dynamics of Signed Citation Networks. In *25th WWW*, 2016.

[18] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The Slashdot Zoo: Mining a Social Network with Negative Edges. In *18th WWW*, pp. 741, 2009.

[19] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *19th WWW*, pp. 641–650, 2010.

[20] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[21] S. Maniu, T. Abdessalem, B. and Cautis. Casting a Web of Trust over Wikipedia: An Interaction-based Approach. In *20th WWW*, pp. 87–88, 2011.

[22] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *20th WWW*, pp. 567–576. ACM, 2011.

[23] A. Papaoikonomou, M. Kardara, K. Tserpes, and T.A. Varvarigou. Predicting Edge Signs in Social Networks Using Frequent Subgraph Discovery. *IEEE Internet Computing*, 18(5):36–43, 2014.

[24] A. Papaoikonomou, M. Kardara, and T.A. Varvarigou. Trust Inference in Online Social Networks. In *Proc. Intl. Conf. on Advances in Social Networks Analysis and Mining*, pp. 600–604, 2015.

[25] Y. Qian and S. and Adali. Foundations of Trust and Distrust in Networks: Extended Structural Balance Theory. *ACM Trans. Web*, 8(3):13:1–13:33, 2014.

[26] M. Shahriari and M. Jalili. Ranking nodes in signed social networks. *Social Network Analysis and Mining*, 4(1):1–12, 2014.

[27] D. Song and D.A. Meyer. Link sign prediction and ranking in signed directed social networks. *Social Network Analysis and Mining*, 5(1):1–14, 2015.

[28] J. Tang, H. Gao, X. Hu, and H. Liu. Exploiting homophily effect for trust prediction. In *6th WSDM*, pp. 53–62, 2013.

[29] J. Tang, Y. Chang, C. Aggarwal, H. Liu. A Survey of Signed Network Mining in Social Media. *arXiv preprint arXiv:1511.07569*, 2015

[30] V.A. Traag, Y.E. Nesterov, and P. Van Dooren. *Exponential Ranking: taking into account negative links*. Springer, 2010.

[31] Z. Wu, C. Aggarwal, and J. Sun. The troll-trust model for ranking in signed networks. In *9th WSDM*, pages 447–456. ACM, 2016.

[32] J. Ye, H. Cheng, Z. Zhu, and M. Chen. Predicting Positive and Negative Links in Signed Social Networks by Transfer Learning. In *22nd WWW*, pp. 1477–1488, 2013.

[33] Zheng, Q and Skillicorn, D.B. *Spectral Embedding of Signed Networks*, chapter 7, pp. 55–63. 2015.

[34] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

[35] K. Zolfaghar and A. Aghaie. Mining trust and distrust relationships in social web applications. In *IEEE ICCP*, pp. 73–80. IEEE, 2010.

A Proofs from Section 4

The following ancillary results will be useful.

Lemma 1 (Hoeffding's inequality for sampling without replacement)**.** *Let $\mathcal{X} = \{x_1, \dots, x_N\}$ be a finite subset of $[0,1]$ and let*

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \ .$$

If $X_1, \dots, X_n$ is a random sample drawn at random from $\mathcal{X}$ without replacement, then, for every $\varepsilon > 0$,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mu\right| \geq \varepsilon\right) \leq 2e^{-2n\varepsilon^2} \ .$$

Lemma 2. *Let $\mathcal{N}_1, \dots, \mathcal{N}_n$ be subsets of a finite set E. Let $E_0 \subseteq E$ be sampled uniformly at random* without *replacement from E, with $|E_0| = m$. Then, for $\delta \in (0,1)$, $Q > 0$, and $\theta \geq 2 \times \max\left\{Q, 4\ln\frac{n}{\delta}\right\}$, we have*

$$\mathbb{P}\Big(\exists i \,:\, |\mathcal{N}_i| \geq \theta, |\mathcal{N}_i \cap E_0| < Q\Big) \leq \delta$$

provided $|E| \geq m \geq \frac{2|E|}{\theta} \times \max\left\{Q, 4\ln\frac{n}{\delta}\right\}$.

Proof of Lemma 2. Set for brevity $p_i = |\mathcal{N}_i|/|E|$. Then, due to the sampling without replacement, each random variable $|\mathcal{N}_i \cap E_0|$ is the sum of m dependent Bernoulli random variables $X_{i,1}, \dots, X_{i,m}$ such that $\mathbb{P}(X_{i,t} = 1) = p_i$, for $t = 1, \dots, m$. Let i be such that $|\mathcal{N}_i| \geq \theta$. Then the condition $m \geq \frac{2|E|Q}{\theta}$ implies

$$Q \leq \frac{m\theta}{2|E|} \leq \frac{m\,p_i}{2} = \frac{\mathbb{E}\big[|\mathcal{N}_i \cap E_0|\big]}{2} \ .$$

Since the variables $X_{i,j}$ are negatively associated, we may apply a (multiplicative) Chernoff bound [3, Section 3.1]. This gives

$$\mathbb{P}\big(|\mathcal{N}_i \cap E_0| < Q\big) \leq e^{-\frac{m\,p_i}{8}} \leq e^{-\frac{m\,\theta}{8|E|}}$$

so that $\mathbb{P}\big(\exists i \,:\, |\mathcal{N}_i| \geq \theta, |\mathcal{N}_i \cap E_0| < Q\big) \leq n\,e^{-\frac{m\,\theta}{8|E|}}$, which is in turn upper bounded by δ whenever $m \geq \frac{8|E|}{\theta}\ln\frac{n}{\delta}$. □

Let now $E_\theta = \{(i,j) \in E \,:\, d_{\text{in}}(j) \geq \theta,\, d_{\text{out}}(i) \geq \theta\} \setminus E_0$, where $E_0 \subseteq E$ is the set of edges sampled by the learning algorithm of Section 4.1. Then Theorem 1 in the main paper is an immediate consequence of the following lemma.

Lemma 3. *Let $G(Y) = (V, E(Y))$ be a directed graph with labels on the edges generated according to the model in Section 3. For all $0 < \alpha, \delta < 1$ and $0 < \varepsilon < \frac{1}{16}$, if the learning algorithm of Section 4.1 is run with parameters $Q = \frac{1}{2\varepsilon^2}\ln\frac{4|V|}{\delta}$ and α, then with probability at least $1 - 11\delta$ the predictions $\widehat{y}(i,j)$ satisfy $\widehat{y}(i,j) = y^*(i,j)$ for all $(i,j) \in E_\theta$ such that $\left|\eta(i,j) - \frac{1}{2}\right| > 8\varepsilon$.*

Proof of Lemma 3. We apply Lemma 2 with $\theta = \frac{2Q}{\alpha} \geq 2 \times \max\left\{Q, 4\ln\frac{2|V|+1}{\delta}\right\}$ to the $2|V|+1$ subsets of E consisting of E_L and $\mathcal{E}_{\text{in}}(i), \mathcal{E}_{\text{out}}(i)$, for $i \in V$. We have that, with probability at least $1-\delta$, at least Q edges of E_L are sampled, at least Q edges of $\mathcal{E}_{\text{in}}(i)$ are sampled for each i such that $\mathcal{N}_{\text{in}}(i) \geq \theta$, and at least Q edges of $\mathcal{E}_{\text{out}}(j)$ are sampled for each j such that $\mathcal{N}_{\text{out}}(j) \geq \theta$. For all $(i,j) \in E_\theta$ let

$$\overline{p}_j = \frac{1}{d_{\text{in}}(j)}\sum_{i \in \mathcal{N}_{\text{in}}(j)} p_i \quad \text{and} \quad \overline{q}_i = \frac{1}{d_{\text{out}}(i)}\sum_{j \in \mathcal{N}_{\text{out}}(i)} q_j$$

and set for brevity $\widehat{\delta}_{\text{in}}(j) = 1 - \widehat{un}(j)$ and $\widehat{\delta}_{\text{out}}(i) = 1 - \widehat{tr}(i)$. We now prove that $\widehat{\delta}_{\text{in}}(j)$ and $\widehat{\delta}_{\text{out}}(i)$ are concentrated around their expectations for all $(i,j) \in E_\theta$. Consider $\widehat{\delta}_{\text{out}}(i)$ (the same argument works for $\widehat{\delta}_{\text{in}}(j)$). Let $J_1, \dots, J_Q$ be the first Q draws in $E_0 \cap \mathcal{N}_{\text{out}}(i)$ and define

$$\widehat{\mu}_p(i) = \frac{1}{Q}\sum_{t=1}^{Q}\frac{p_i + q_{J_t}}{2} \ .$$

Applying Lemma 1 to the set $\left\{\frac{p_i+q_j}{2} \,:\, j \in \mathcal{N}_{\text{out}}(i)\right\}$, and using our choice of Q, we get that $\left|\widehat{\mu}_p(i) - \mu_p(i)\right| \leq \varepsilon$ holds with probability at least $1 - \delta/(2|V|)$, where

$$\mu_p(i) = \frac{1}{d_{\text{out}}(i)}\sum_{j \in \mathcal{N}_{\text{out}}(i)}\frac{p_i + q_j}{2} = \frac{p_i + \overline{q}_i}{2} \ .$$

Now consider the random variables $Z_t = \mathbb{I}\{y_{i,J_t} = 1\}$, for $t = 1, \dots, Q$. Conditioned on $J_1, \dots, J_Q$, these are independent Bernoulli random variables with $\mathbb{E}[Z_t \mid J_t] = \frac{p_i + q_{J_t}}{2}$. Hence, applying a standard Hoeffding bound for independent variables and using our choice of Q, we get that

$$\left|\frac{1}{Q}\sum_{t=1}^{Q} Z_t - \widehat{\mu}_p(i)\right| \leq \varepsilon$$

with probability at least $1-\delta/(2|V|)$ for every realization of $J_1,\dots,J_Q$. Since $\widehat{\delta}_{\text{out}}(i) = (Z_1+\cdots+Z_Q)/Q$, we get that $\big|\widehat{\delta}_{\text{out}}(i) - \mu_p(i)\big| \le 2\varepsilon$ with probability at least $1-2\delta/(2|V|)$. Applying the same argument to $\widehat{\delta}_{\text{in}}(j)$, and the union bound[9] on the set $\Big\{\widehat{\delta}_{\text{in}}(j), \widehat{\delta}_{\text{out}}(i) \,:\, (i,j)\in E_\theta\Big\}$, we get that

$$\left|\widehat{\delta}_{\text{out}}(i) + \widehat{\delta}_{\text{in}}(j) - \frac{p_i+q_j}{2} - \frac{\overline{p}_j+\overline{q}_i}{2}\right| \le 4\varepsilon \qquad (6)$$

simultaneously holds for all $(i,j)\in E_\theta$ with probability at least $1-4\delta$. Now notice that $\overline{p}_j$ is a sample mean of Q i.i.d. $[0,1]$-valued random variables drawn from the prior marginal $\int_0^1 \mu(\cdot,q)dq$ with expectation μ_p. Similarly, $\overline{q}_i$ is a sample mean of Q i.i.d. $[0,1]$-valued random variables independently drawn from the prior marginal $\int_0^1 \mu(p,\cdot)dp$ with expectation μ_q. By applying Hoeffding bound for independent variables, together with the union bound to the set of pairs of random variables whose sample means are $\overline{p}_j$ and $\overline{q}_i$ for each $(i,j)\in E_\theta$ (there are at most $2|V|$ of them) we obtain that

$$\big|\overline{p}_j - \mu_p\big| \le \varepsilon \qquad \text{and} \qquad \big|\overline{q}_i - \mu_q\big| \le \varepsilon$$

hold simultaneously for all $(i,j)\in E_\theta$ with probability at least $1-2\delta$. Combining with (6) we obtain that

$$\left|\widehat{\delta}_{\text{out}}(i) + \widehat{\delta}_{\text{in}}(j) - \frac{p_i+q_j}{2} - \frac{\mu_p+\mu_q}{2}\right| \le 5\varepsilon \qquad (7)$$

simultaneously holds for each $(i,j)\in E_\theta$ with probability at least $1-6\delta$. Next, let E'_L be the set of the first Q edges drawn in $E_L\cap E_0$. Then

$$\mathbb{E}\big[\widehat{\tau}\big] = \frac{1}{Q}\sum_{(i,j)\in E'_L}\mathbb{P}\big(y_{i,j}=1\big) = \frac{1}{Q}\sum_{(i,j)\in E'_L}\frac{p_i+q_j}{2}\ ,$$

where the expectation is w.r.t. the independent draws of the labels $y_{i,j}$ for $(i,j)\in E'_L$. Hence, by applying again Hoeffding bound (this time without the union bound) to the $Q=\frac{1}{2\varepsilon^2}\ln\frac{2}{\delta}$ independent Bernoulli random variables $\mathbb{I}\{y_{i,j}=1\}$, $(i,j)\in E'_L$, the event $\big|\widehat{\tau}-\mathbb{E}\big[\widehat{\tau}\big]\big|\le\varepsilon$ holds with probability at least $1-\delta$. Now, introduce the function

$$F(\boldsymbol{p},\boldsymbol{q}) = \mathbb{E}\big[\widehat{\tau}\big] = \frac{1}{Q}\sum_{(i,j)\in E'_L}\frac{p_i+q_j}{2}\ .$$

For any realization $\boldsymbol{q}_0$ of $\boldsymbol{q}$, the function $F_1(\boldsymbol{p}) = F(\boldsymbol{p},\boldsymbol{q}_0)$ is a sample mean of $Q=\frac{1}{2\varepsilon^2}\ln\frac{4|V|}{\delta}$ i.i.d. $[0,1]$-valued random variables $\{p_i \,:\, (i,j)\in E'_L\}$ (recall that if $i\in V$ is the origin of an edge $(i,j)\in E'_L$, then it is not the origin of any other edge $(i,j')\in E'_L$). Using again the standard Hoeffding bound, we obtain that

$$\big|F(\boldsymbol{p},\boldsymbol{q}) - E_{\boldsymbol{p}}\big[F(\boldsymbol{p},\boldsymbol{q})\big]\big| \le \varepsilon$$

holds with probability at least $1-\delta$ for each $\boldsymbol{q}\in[0,1]^{|V|}$. With a similar argument, we obtain that

$$\big|E_{\boldsymbol{p}}\big[F(\boldsymbol{p},\boldsymbol{q})\big] - E_{\boldsymbol{p},\boldsymbol{q}}\big[F(\boldsymbol{p},\boldsymbol{q})\big]\big| \le \varepsilon$$

also holds with probability at least $1-\delta$. Since

$$E_{\boldsymbol{p},\boldsymbol{q}}\big[F(\boldsymbol{p},\boldsymbol{q})\big] = \frac{\mu_p+\mu_q}{2}$$

we obtain that

$$\left|\widehat{\tau} - \frac{\mu_p+\mu_q}{2}\right| \le 3\varepsilon \qquad (8)$$

with probability at least $1-3\delta$. Combining (7) with (8) we obtain

$$\left|\widehat{\delta}_{\text{out}}(i) + \widehat{\delta}_{\text{in}}(j) - \frac{p(i)+q(j)}{2}\right| \le 8\varepsilon$$

simultaneously holds for each $(i,j)\in E_\theta$ with probability at least $1-10\delta$. Putting together concludes the proof. □

Derivation of the maximum likelihood equations. Recall that the training set $E_0 = \big\{(i_t,j_t), y_{i_t,j_t}\big) \,:\, t=1,\dots,m\big\}$ is drawn uniformly at random from E without replacement. We can write

$$\begin{aligned}
&\mathbb{P}\left(E_0 \,\middle|\, \{p_i,q_i\}_{i=1}^{|V|}\right)\\
&= \frac{1}{\binom{|E|}{m}\,m!}\prod_{k=1}^m\left(\frac{p_{i_k}+q_{j_k}}{2}\right)^{\mathbb{I}\{y_{i_k,j_k}=+1\}}\\
&\quad\times\prod_{k=1}^m\left(1-\frac{p_{i_k}+q_{j_k}}{2}\right)^{\mathbb{I}\{y_{i_k,j_k}=-1\}}\\
&= \frac{1}{\binom{|E|}{m}\,m!}\prod_{\ell=1}^{|V|}\Bigg(\prod_{k=1}^m\left(\frac{p_\ell+q_{j_k}}{2}\right)^{\mathbb{I}\{i_k=\ell,\,y_{\ell,j_k}=+1\}}\\
&\quad\times\prod_{k=1}^m\left(1-\frac{p_\ell+q_{j_k}}{2}\right)^{\mathbb{I}\{i_k=\ell,\,y_{\ell,j_k}=-1\}}\Bigg)
\end{aligned}$$

so that $\log\mathbb{P}\left(E_0 \,\middle|\, \{p_i,q_i\}_{i=1}^{|V|}\right)$ is proportional to

$$\begin{aligned}
&\sum_{\ell=1}^{|V|}\sum_{k=1}^m\mathbb{I}\{i_k=\ell, y_{\ell,j_k}=+1\}\log\left(\frac{p_\ell+q_{j_k}}{2}\right)\\
&+\sum_{\ell=1}^{|V|}\sum_{k=1}^m\mathbb{I}\{i_k=\ell, y_{\ell,j_k}=+1\}\log\left(1-\frac{p_\ell+q_{j_k}}{2}\right)
\end{aligned}$$

[9] The sample spaces for the ingoing and outgoing edges of the vertices occurring as endpoints in E_θ overlap. Hence, in order to prove a uniform concentration result, we need to apply the union bound over the random variables defined over these sample spaces, which motivates the presence of the factor $\ln(2|V|)$ in the definition of Q.

and

$$\frac{\partial \log \mathbb{P}\left(E_0 \,\middle|\, \{p_i, q_i\}_{i=1}^{|V|}\right)}{\partial p_\ell} = \sum_{k=1}^{m} \frac{\mathbb{I}\{i_k = \ell, y_{\ell,j_k} = +1\}}{p_\ell + q_{j_k}} - \sum_{k=1}^{m} \frac{\mathbb{I}\{i_k = \ell, y_{\ell,j_k} = -1\}}{2 - p_\ell - q_{j_k}} .$$

By a similar argument,

$$\mathbb{P}\left(E_0 \,\middle|\, \{p_i, q_i\}_{i=1}^{|V|}\right) = \frac{1}{\binom{|E|}{m} m!} \prod_{\ell=1}^{|V|} \left(\prod_{k=1}^{m} \left(\frac{p_{i_k} + q_\ell}{2}\right)^{\mathbb{I}\{j_k=\ell,\, y_{i_k,\ell}=+1\}} \times \prod_{k=1}^{m} \left(1 - \frac{p_{i_k} + q_\ell}{2}\right)^{\mathbb{I}\{j_k=\ell,\, y_{i_k,\ell}=-1\}} \right)$$

so that

$$\frac{\partial \log \mathbb{P}\left(E_0 \,\middle|\, \{p_i, q_i\}_{i=1}^{|V|}\right)}{\partial q_\ell} = \sum_{k=1}^{m} \frac{\mathbb{I}\{j_k = \ell, y_{i_k,\ell} = +1\}}{p_{i_k} + q_\ell} - \sum_{k=1}^{m} \frac{\mathbb{I}\{j_k = \ell, y_{i_k,\ell} = -1\}}{2 - p_{i_k} - q_\ell} .$$

B Proofs from Section 5

Proof of Theorem 2. Let each node $i \in V$ host two instances of the randomized Weighted Majority (RWM) algorithm [4] with an online tuning of their learning rate [2, 1]: one instance for predicting the sign of outgoing edges (i, j), and one instance for predicting the sign of incoming edges (j, i). Both instances simply compete against the two constant experts, predicting always $+1$ or always -1. Denote by $M(i, j)$ the indicator function (zero-one loss) of a mistake on edge (i, j). Then the expected number of mistakes of each RWM instance satisfy [2, 1]:

$$\sum_{j \in \mathcal{N}_{\text{out}}(i)} \mathbb{E}\, M(i,j) = \Psi_{\text{out}}(i, Y) + O\left(\sqrt{\Psi_{\text{out}}(i, Y)} + 1\right)$$

and

$$\sum_{i \in \mathcal{N}_{\text{in}}(j)} \mathbb{E}\, M(i,j) = \Psi_{\text{in}}(j, Y) + O\left(\sqrt{\Psi_{\text{in}}(j, Y)} + 1\right) .$$

We then define two meta-experts: an ingoing expert, which predicts $y_{i,j}$ using the prediction of the ingoing RWM instance for node j, and the outgoing expert, which predicts $y_{i,j}$ using the prediction of the outgoing RWM instance for node i. The number of mistakes of these two experts satisfy

$$\sum_{i \in V} \sum_{j \in \mathcal{N}_{\text{out}}(i)} \mathbb{E}\, M(i,j) = \Psi_{\text{out}}(Y) + O\left(\sqrt{|V| \Psi_{\text{out}}(Y)} + |V|\right)$$

$$\sum_{j \in V} \sum_{i \in \mathcal{N}_{\text{in}}(j)} \mathbb{E}\, M(i,j) = \Psi_{\text{in}}(Y) + O\left(\sqrt{|V| \Psi_{\text{in}}(Y)} + |V|\right) ,$$

where we used $\sum_{j \in V} \sqrt{\Psi_{\text{in}}(j, Y)} \le \sqrt{|V| \Psi_{\text{in}}(Y)}$, and similarly for $\Psi_{\text{out}}(Y)$. Finally, let the overall prediction of our algorithm be a RWM instance run on top of the ingoing and the outgoing experts. Then the expected number of mistakes of this predictor satisfies

$$\sum_{(i,j) \in E} \mathbb{E}\, M(i,j) = \Psi_G(Y) + O\left(\sqrt{|V| \Psi_G(Y)} + |V| + \sqrt{\left(\Psi_G(Y) + |V| + \sqrt{|V| \Psi_G(Y)}\right)} \right) = \Psi_G(Y) + O\left(\sqrt{|V| \Psi_G(Y)} + |V|\right) ,$$

as claimed. □

Proof sketch of Theorem 3. Let $\mathcal{Y}_K$ be the set of all labelings Y such that the total number of negative and positive edges are K and $|E| - K$, respectively (without loss of generality we will focus on negative edges). Consider the randomized strategy that draws a labeling $Y \in \{-1, +1\}^{|E|}$ uniformly at random from $\mathcal{Y}_K$. For each node $i \in V$, we have $\Psi_{\text{in}}(i, Y) \le d^-_{\text{in}}(i)$, which implies $\Psi_{\text{in}}(Y) \le K$. A very similar argument applies to the outgoing edges, leading to $\Psi_{\text{out}}(Y) \le K$. The constraint $\Psi_G(Y) \le K$ is therefore always satisfied.

The adversary will force on average $1/2$ mistakes in each one of the first K rounds of the online protocol by repeating K times the following: (i) A label value $\ell \in \{-1, +1\}$ is selected uniformly at random. (ii) An edge (i, j) is sampled uniformly at random from the set of all edges that were not previously revealed and whose labels are equal to ℓ.

The learner is required to predict $y_{i,j}$ and, in doing so, $1/2$ mistakes will be clearly made on average because of the randomized labeling procedure. Observe that this holds even when A knows the value of K and $\Psi_G(Y)$. Hence, we can conclude that the expected number of mistakes that A can be forced to make is always at least $K/2$, as claimed.

We now show that, as $\frac{K}{|E|} \to 0$, the lower bound gets arbitrarily close to K for any $G(Y)$ and any constant

K. Let $\mathcal{E}$ be the following event: There is at least one unrevealed negative label. The randomized iterative strategy used to achieve this result is identical to the one described above, except for the stopping criterion. Instead of repeating step (i) and (ii) only for the first K rounds, these steps are repeated until $\mathcal{E}$ is true. Let $m_{r,c}$ be defined as follows: For $c = 1$ it is equal to the expected number of mistakes forced in round r when $K = 1$. For $c > 1$ it is equal to the difference between the expected number of mistakes forced in round r when $K = c$ and $K = c - 1$. One can see that $m_{r,c}$ is null when $r < c$. When $K = 1$, the probability that $\mathcal{E}$ is true in round r is clearly equal to $\frac{1}{2^{r-1}}$. Hence, the expected number of mistakes made by A when $K = 1$ in any round r is equal to $\frac{1}{2}\frac{1}{2^{r-1}} = \frac{1}{2^r}$. We can therefore conclude that $m_{r,1} = \frac{1}{2^r}$ for all r.

A simple calculation shows that if $r = c$ then $m_{r,c} = \frac{1}{2^r}$. Furthermore, when $r > 1$ and $c > 1$, we have the following recurrence:

$$m_{r,c} = \frac{m_{r-1,c} + m_{r-1,c-1}}{2} \ .$$

In order to calculate $m_{r,c}$ for all r and c, we will rest on the ancillary quantity $s_j(i)$, recursively defined as specified next.

Given any integer variable i, we have $s_0(i) = 1$ and, for any positive integer j,

$$s_j(i) = \sum_{k=1}^{i} s_{j-1}(k) \ .$$

It is not difficult to verify that

$$m_{r,c} = \frac{s_{c-1}(r-c+1)}{2^r}.$$

Since $s_j(i) = \frac{\langle i\rangle_j}{j!}$, where $\langle i\rangle_j$ is the rising factorial $i(i+1)(i+2)\dots(i+j-1)$, we have

$$m_{r,c} = \frac{\langle r-c+1\rangle_{c-1}}{(c-1)!2^r}.$$

When $\frac{K}{|E|} \to 0$, given any integer $K' > 1$, the difference between the expected number of mistakes forced when $K = K'$ and $K = K' - 1$ is equal to

$$\begin{aligned}\sum_{r=K'}^{\infty} m_{r,K'} &= \frac{1}{(K'-1)!}\sum_{r=K'}^{\infty}\frac{\langle r-K'+1\rangle_{K'-1}}{2^r} \\ &= \frac{1}{(K'-1)!2^{K'-1}}\sum_{r'=1}^{\infty}\frac{\langle r'\rangle_{K'-1}}{2^{r'}} \ ,\end{aligned}$$

where we set $r' = r - K' + 1$. Setting $i' = i - 1$ and recalling that

$$\langle i\rangle_j = j!\binom{i+j-1}{i-1} \ ,$$

we have

$$\frac{1}{j!}\sum_{i=1}^{\infty}\frac{\langle i\rangle_j}{2^i} = \sum_{i=1}^{\infty}\frac{\binom{i+j-1}{i-1}}{2^i} = \sum_{i'=0}^{\infty}\frac{\binom{i'+j}{i'}}{2^{i'+1}} \ .$$

Now, using the identity

$$\binom{i'+j+1}{i'} = \binom{i'+j}{i'} + \binom{i'+j}{i'-1} \ ,$$

we can easily prove by induction on j that

$$\sum_{i'=0}^{\infty}\frac{\binom{i'+j}{i'}}{2^{i'+1}} = 2^j \ .$$

Hence, we have

$$\sum_{r=K'}^{\infty} m_{r,K'} = 1.$$

Moreover, as shown earlier, $m_{r,1} = \frac{1}{2^r}$ for all r. Hence we can conclude that when $\frac{K}{|E|} \to 0$

$$\mathbb{E}M_A(Y) \geq \sum_{r=1}^{\infty}\frac{1}{2^r} + \sum_{K'=2}^{K}\sum_{r=K'}^{\infty} m_{r,K'} = K$$

for any edge-labeled graph $G(Y)$ and any constant K, as claimed. □

C Further Experimental Results

This section contains more evidence related to the experiments in Section 6 of the main paper. In particular, we experimentally demonstrate the alignment between BLC(tr, un) and LogReg.

After training on the two features $1-\widehat{tr}(i)$ and $1-\widehat{un}(j)$, LogReg has learned three weights w_0, w_1 and w_2, which allow to predict $y_{i,j}$ according to

$$\text{SGN}\Big(\big(w_1(1-\widehat{tr}(i)\big) + w_2\big(1-\widehat{un}(j)\big) + w_0\Big) \ .$$

This can be rewritten as

$$\text{SGN}\Big(\big(1-\widehat{tr}(i)\big) + w_2'\big(1-\widehat{un}(j)\big) - \tfrac{1}{2} - \tau'\Big) \ ,$$

with $w_2' = \frac{w_2}{w_1}$ and $\tau' = -\left(\frac{1}{2} + \frac{w_0}{w_1}\right)$.

As shown in Table 3, and in accordance with the predictor built out of Equation (2) from the main paper, w_2' is almost 1 on *all datasets*, while τ' tends to be always close the fraction of positive edges in the dataset.

Table 3: Normalized logistic regression coefficients averaged over 12 runs (with one standard deviation)

	$\frac{\lvert E_0 \rvert}{\lvert E \rvert}$	w_2'	τ'
Citations	3%	0.909 ± 0.07	0.592 ± 0.04
	9%	0.985 ± 0.04	0.707 ± 0.03
	15%	1.001 ± 0.03	0.729 ± 0.03
	20%	1.013 ± 0.02	0.747 ± 0.02
	25%	1.011 ± 0.02	0.746 ± 0.01
Wikipedia	3%	0.919 ± 0.04	0.666 ± 0.02
	9%	0.940 ± 0.03	0.726 ± 0.01
	15%	0.947 ± 0.01	0.741 ± 0.01
	20%	0.963 ± 0.01	0.760 ± 0.01
	25%	0.962 ± 0.02	0.764 ± 0.01
Slashdot	3%	1.012 ± 0.02	0.669 ± 0.01
	9%	1.014 ± 0.01	0.703 ± 0.01
	15%	1.007 ± 0.01	0.707 ± 0.01
	20%	1.002 ± 0.00	0.710 ± 0.00
	25%	0.995 ± 0.01	0.712 ± 0.00
Epinion	3%	1.124 ± 0.02	0.786 ± 0.02
	9%	1.066 ± 0.01	0.783 ± 0.01
	15%	1.037 ± 0.01	0.774 ± 0.01
	20%	1.018 ± 0.01	0.765 ± 0.01
	25%	1.010 ± 0.01	0.763 ± 0.01
Wik. Edits	3%	1.075 ± 0.02	0.851 ± 0.02
	9%	1.041 ± 0.01	0.872 ± 0.01
	15%	1.025 ± 0.01	0.876 ± 0.01
	20%	1.012 ± 0.01	0.874 ± 0.01
	25%	1.007 ± 0.01	0.874 ± 0.01

References

[1] P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *J. Comput. Syst. Sci.*, 64(1):48–75, 2002.

[2] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.

[3] D. P. Dubhashi, A. Panconesi. Concentration of Measure for the Analysis of Randomized Algorithms Cambridge University Press, 2009.

[4] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.