Avoiding Biases due to Similarity Assumptions in Node Embeddings

Deepayan Chakrabarti
deepay@utexas.edu
McCombs School of Business, University of Texas at Austin
USA

ABSTRACT

Node embeddings are vectors, one per node, that capture a graph's structure. The basic structure is the adjacency matrix of the graph. Recent methods also make assumptions about the similarity of unlinked nodes. However, such assumptions can lead to unintentional but systematic biases against groups of nodes. Calculating similarities between far-off nodes is also difficult under privacy constraints and in dynamic graphs. Our proposed embedding, called NEWS, makes no similarity assumptions, avoiding potential risks to privacy and fairness. NEWS is parameter-free, enables fast link prediction, and has linear complexity. These gains from avoiding assumptions do not significantly affect accuracy, as we show via comparisons against several existing methods on 21 real-world networks. Code is available at `https://github.com/deepayan12/news`.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**;
• **Information systems** → **Data mining**.

KEYWORDS

node embedding, fairness, robustness

ACM Reference Format:
Deepayan Chakrabarti. 2022. Avoiding Biases due to Similarity Assumptions in Node Embeddings. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539287

1 INTRODUCTION

A node embedding is a low-dimensional vector representation of each node in a graph, that captures the graph's link structure. The embeddings can be used as feature vectors in other tasks. Thus, any method designed for vector inputs can be applied to graph structured data.

Recent work on node embeddings has focused on "second-order" and "higher-order" proximity. These methods choose a similarity measure between nodes, and then find vectors to mimic (or, "embed") the similarity measure. The graph already provides a "first-order" similarity: two nodes are linked, or not. Second-order methods add similarity relations between nodes that are not linked but share friends. Higher-order methods also consider nodes that are farther apart. The use of such similarity measures is justified on the grounds that real-world networks are too sparse, with too few links. Higher-order methods can provide more fine-grained data for the embedding. A variety of such similarity measures have been studied, based on common neighbors, random walks, and personalized pagerank, among others [17, 43, 44, 51, 52, 60].

While second-order and higher-order proximity methods are widely popular, they also have weaknesses. One is that **it is difficult to ensure fairness.** Every similarity measure encodes assumptions: it says that node i is closer to node j than node k even though neither j nor k is linked to i. Such assumptions can lead to hidden biases. For example, the hitting-time similarity between two nodes i and j is the expected time for a random walk from i to reach j. But, it turns out that asymptotically, this only depends on the degree of j; the higher the degree, the higher the similarity [35]. Similar results also hold for commute-time similarity. Suppose we use such a similarity to build node embeddings in a social network. If we use these embeddings in a friend recommendation system, it would only recommend celebrities. It would unintentionally bias against people with few friends, such as introverts or non-native language speakers. Such effects have been observed in community detection too [40]. It is difficult to rule out such biases for any chosen similarity measure.

Furthermore, **different graphs may need different assumptions.** For instance, in social networks, two people with many shared friends are often assumed to be close, even if they are not linked. But in an airport network, if there are many two-hop flights connecting two airports, there is less economic reason to add a direct flight. Similarly, in a peer-to-peer lending network, two people who borrow from the same set of lenders are unlikely to borrow from each other. Thus, an assumption that helps link prediction on one network may hurt it on other networks.

In some networks, **privacy constraints may prohibit higher-order similarity computations.** For example, companies may prohibit crawls of their internal knowledge networks. In private networks, everyone knows their neighbors, but no one sees the entire network. So, to build a node's embedding, we can only use the embeddings of its neighbors and a sample of non-neighbors. This rules out higher-order similarity methods.

Finally, **higher-order similarity matrices can be expensive to compute and maintain.** A matrix storing pairwise similarities requires $O(n^2)$ space for n nodes. Approximations via random

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539287

walks [17, 52] need access to the entire network, which may be difficult under privacy restrictions. Furthermore, we must keep recomputing the matrix in dynamic networks such as computer networks, location-based social networks, and financial networks. The disruption of even a few links can affect higher-order similarity metrics for a large subset of nodes.

Our goal in this paper is *an embedding method for plain graphs, without node or edge features or any side information.* We want to minimize the risk of unfairness, without knowing the sensitive attributes for the nodes. The above discussion suggests two constraints on the desired method. First, **the algorithm should make no assumptions beyond what the data says,** that is, whether two nodes are linked or not. In other words, it can only use first-order proximity. Second, **each node's embedding should be personalized.** That is, the embedding for node i should be the best possible from i's point of view, given everyone else's embeddings. No node is sacrificed to optimize an overall quality metric. Note that this disallows hyper-parameter tuning via an overall loss measure. Thus, a first-order personalized method reduces the chances of systematic biases against any node or group of nodes. Such a method is also amenable to privacy restrictions, and can be easily updated in dynamic networks. Note that our problem setting has no node features; the use of sensitive features in embedding or post-processing is an orthogonal problem [1, 6, 12, 20, 47].

Now, the popularity of higher-order proximity methods stems from their accuracy. By avoiding their assumptions, we reduce the risk of biases and gain in terms of privacy and computational efficiency. But the gains are appealing only if the loss in accuracy is minor. So, the question is: *Can we still achieve accurate embeddings under the no-similarity-assumptions constraint?*

Our contributions. On 21 different networks, including social, communication, citation, product co-purchase, and transportation networks, we show that **our first-order method is comparable to the best higher-order methods** in terms of accuracy.

We achieve this with a new node embedding algorithm called NEWS (**N**ode **E**mbeddings **W**ithout **S**imilarity assumptions)[1]. Under NEWS, the embedding of a node i is the feature vector of a personalized classifier for i. Given the embedding of another node j, this classifier predicts if j is a neighbor of i, or not. Since many nodes have few neighbors, and hence limited training data, NEWS uses a robust training algorithm. This algorithm only uses statistics that can be reliably estimated even from limited data. NEWS's time and space complexities are **linear in the number of edges and nodes** respectively. Furthermore, NEWS is **parameter-free**, and needs no cross-validation. It also enables **fast link prediction** via simple matrix operations.

The rest of the paper is organized as follows. We present NEWS in Section 2, and analyze it in Section 3. Empirical results are shown in Section 4. We discuss related work in Section 5, and conclude in Section 6.

2 PROPOSED WORK

We are given an undirected graph of n nodes with adjacency matrix A, where $A_{ij} = A_{ji} = 1$ if nodes i and j are linked by an edge, and 0 otherwise. We want to find vectors $\boldsymbol{u}_i \in \mathbb{R}^d (i \in [n])$ that captures the information in A. In other words, we should be able to infer A_{ij} from the value of $g(\boldsymbol{u}_i, \boldsymbol{u}_j)$, for some fixed function $g(.,.)$. Thus, the vectors $\boldsymbol{u}_i$ "embed" the network A.

The algorithm to infer $\{\boldsymbol{u}_i\}$ should have three properties:

- **(P1)** It should only use **first-order proximity.** Either two nodes are linked, or not; the algorithm should make no extra assumptions about node similarity.
- **(P2)** The algorithm should be **parameter-free.** Hyperparameter-tuning can tilt results in favor of the majority while negatively affecting a hidden minority. Furthermore, embeddings are often used for tasks that are not known beforehand. So, we cannot rely on tuning hyperparameters.
- **(P3)** Given $\{\boldsymbol{u}_i\}$, **link prediction should be fast and simple.** In other words, the function $g(\boldsymbol{u}_i, \boldsymbol{u}_j)$ should be easy to compute and fixed a priori.

We will first present the formulation and main idea of our proposed method. Then, we will discuss its details, its computational complexity, and the extension to directed graphs.

2.1 Formulation

Consider the following problem:

(Local Problem) *Given* $\{A_{ij}; j \neq i\}$ *and* $\{\boldsymbol{u}_j \in \mathbb{R}^d; j \neq i\}$, *find* $\boldsymbol{u}_i$.

The local problem focuses on inferring $\boldsymbol{u}_i$ from only first-order proximity. The global embedding of all nodes is just the fixed-point solution of local problems for all $i \in [n]$.

For the local problem, we can split the set of nodes $[n] \setminus \{i\}$ into the neighbors of i ($S_{i+} = \{j; j \neq i, A_{ij} = 1\}$) and everyone else (S_{i-}). Now, we can think of $\boldsymbol{u}_i$ as the parameter vector of a classifier C_i. The training data for C_i has the set S_{i+} as the positive class and S_{i-} as the negative class. Each "training point" $j \in S_{i+} \cup S_{i-}$ has a d-dimensional "feature vector" $\boldsymbol{u}_j$. In the local problem, these feature vectors are known. Since $|S_{i+}| \ll |S_{i-}|$ typically, the classification problem is imbalanced. Further, many real-world networks have skewed degree distributions, with most nodes having low degrees [9]. In other words, $|S_{i+}|$ is very small for a large fraction of the nodes. For example, in the benchmark Flickr network, 41% of the nodes have fewer than 32 neighbors, and 73% have fewer than 128 neighbors. So, if we seek $\boldsymbol{u}_i \in \mathbb{R}^d$ with $d = 32$ or 128, we have fewer positive points than features for many nodes: $|S_{i+}| < d$. Thus, **inferring $\boldsymbol{u}_i$ corresponds to imbalanced classification from very limited data.**

Interpretation of existing methods. Existing imbalanced classifiers are ill-suited for such extreme data scarcity. For example, when $|S_{i+}| < d$, the positive points lie in a low-dimensional subspace of the feature space. Sampling-based or cost-sensitive methods may not account for this artificially low dimensionality [10, 27]. Complex ensemble-based and neural classifiers have many parameters, and hence may overfit [26] .

Existing embedding methods counter imbalance by using a fixed ratio of negative to positive samples. But does not fix the scarcity of positive samples. Second-order and higher-order proximity augments the positive set S_{i+} with nodes that are not directly linked to i. This reduces data scarcity, but requires extra assumptions. As discussed in Section 1, such assumptions may be unfair and have other weaknesses, which we wish to avoid.

[1] Code is available at https://github.com/deepayan12/news.

2.2 Main Idea

When $|S_{i+}|$ is small, the average loss on S_{i+} is a poor proxy for the expected test loss for the positive class. So, if we optimize C_i over the average loss, it can overfit. Our approach is to construct a robust smoothed distribution $\mathcal{D}^{\star}_{i+}$ for the positive class. Then, instead of the average loss on S_{i+}, we use the expected loss on $\mathcal{D}^{\star}_{i+}$. Furthermore, we ensure that this expected loss has a closed-form formula, via an appropriate choice of the loss function.

Robust smoothed distribution $\mathcal{D}^{\star}_{i+}$. We use a robust kernel density estimate for the positive class as $\mathcal{D}^{\star}_{i+}$. Each node i has a "personalized" kernel, built from statistics that can be reliably estimated even when $|S_{i+}|$ is small. By relying only on such robust statistics, NEWS avoids overfitting to the idiosyncrasies in the data. The personalization of $\mathcal{D}^{\star}_{i+}$ also contrasts with alternative approaches such as using a generic regularization term for all nodes.

Choice of classifier C_i. The desired embedding $\boldsymbol{u}_i$ is the parameter vector that minimizes the expected test loss of C_i. For the negative class, the expected test loss is close to the average loss on S_{i-}, since $|S_{i-}|$ is large enough. For the positive class, we use the expected loss over $\mathcal{D}^{\star}_{i+}$, as discussed above. Now, in general, this expected loss over $\mathcal{D}^{\star}_{i+}$ will not have a closed form. We can approximate it by sampling, but this increases the variability of results and the computational effort. Instead, we choose a loss function for which the expected loss under $\mathcal{D}^{\star}_{i+}$ has a closed form. This simplifies and speeds up the optimization of $\boldsymbol{u}_i$.

Thus, we can solve the local problem (find $\boldsymbol{u}_i$ for node $i \in [n]$) by training C_i using the above approach. The global problem of finding all embeddings is the fixed-point solution of all n local problems. Our proposed method, called NEWS, trains all classifiers $\cup_i C_i$ jointly to solve the global problem.

Matching properties (P1)-(P3). NEWS uses the network only to construct the subsets S_{i+} and S_{i-} for each node i. It makes no further assumptions about node similarities. Hence, NEWS is a first-order proximity method, matching property (**P1**). The entire method has no hyperparameters, so no cross-validation is necessary. The only parameters are those for the optimizer, which are fixed for all our experiments and standard for all algorithms. So NEWS satisfies property (**P2**). Finally, with our chosen classification model, link prediction only needs simple matrix operations, matching property (**P3**). Finally, our focus on the local problem makes NEWS personalized by default.

2.3 Details of NEWS

The chances of two people being friends depends on (a) how much their interests match, and also (b) their ability to attract friends irrespective of interests ("celebrity" status). To model this, NEWS splits every node vector $\boldsymbol{u}_i \in \mathbb{R}^d$ into a "bias" term $\alpha_i \in \mathbb{R}$ and a vector of "interests" $\boldsymbol{\beta}_i \in \mathbb{R}^{d-1}$, i.e., $\boldsymbol{u}_i = (\alpha_i, \boldsymbol{\beta}_i)$. So, our goal in the local problem is to infer $\boldsymbol{u}_i = (\alpha_i, \boldsymbol{\beta}_i)$ given all $\{\boldsymbol{u}_j = (\alpha_j, \boldsymbol{\beta}_j); j \neq i\}$.

To infer the interests $\boldsymbol{\beta}_i$ of node i, we need to know the interest distribution among i's neighbors (S_{i+}) and non-neighbors (S_{i-}). Since there are many non-neighbors (large $|S_{i-}|$), we can use the empirical distribution. But many nodes have few neighbors (small $|S_{i+}|$). So, NEWS constructs a robust distribution $\mathcal{D}^{\star}_{i+}$ from the neighbors' interests. We will now discuss the construction of $\mathcal{D}^{\star}_{i+}$ and the optimization of $\boldsymbol{u}_i = (\alpha_i, \boldsymbol{\beta}_i)$. Then, we will present the complexity analysis, and the extension to directed graphs.

Robust smoothed distribution. For each node i, we want a smooth density $\mathcal{D}^{\star}_{i+}$ for the positive class that is personalized to i. Such personalization must be based on the statistics of i's neighbors $\{\boldsymbol{u}_j; j \in S_{i+}\}$. When there are few neighbors, only low-order moments can be reliably estimated. Higher-order moments are more sensitive to the tail of a distribution, and hence are harder to estimate accurately from a few samples. So, the personalized density can use robust estimates of the mean and covariance, but otherwise should be as flexible as possible.

A common measure of the flexibility of a distribution is its entropy. The maximum-entropy distribution with a given mean and covariance is the Gaussian distribution [11]. So, we use a Gaussian kernel density as i's personalized density $\mathcal{D}^{\star}_{i+}$ for the positive class. Specifically, we set the probability density at $\boldsymbol{x} \in \mathbb{R}^{d-1}$ to be

$$p_{\mathcal{D}^{\star}_{i+}}(\boldsymbol{x}) = \frac{1}{|S_{i+}|} \sum_{j \in S_{i+}} \phi\left((\Sigma^{\star}_{i+})^{-1/2}(\boldsymbol{x} - \boldsymbol{\beta}_j)\right), \tag{1}$$

$$\Sigma^{\star}_{i+} = \eta_i \cdot \hat{\Sigma}_{i+} + \nu_i \cdot I, \tag{2}$$

where $\phi(.)$ is the standard Normal density, $\boldsymbol{\beta}_j$ the interest vector for node j, and $\hat{\Sigma}_{i+}$ the sample covariance of $\{\boldsymbol{\beta}_j; j \in S_{i+}\}$. Here, η_i and ν_i are shrinkage parameters for the robust covariance estimator $\Sigma^{\star}_{i+}$. **We choose the optimal shrinkage** (η_i, ν_i) to minimize the expected mean-squared error of $\Sigma^{\star}_{i+}$. These optimal values can be computed via simple matrix operations [33].

Thus, the density $\mathcal{D}^{\star}_{i+}$ uses a maximum-entropy kernel based on robust estimates of the low-order moments. This allows for personalization without being sensitive to the noise in S_{i+}. We note that our kernel density does not have a bandwidth parameter. The optimal bandwidth varies as $n^{-1/(d+4)}$, where n is the number of points and d the dimensionality [50]. In our case, $n = |S_{i+}|$ is often small, while the embedding dimension d is much larger ($d = 32$ or $d = 128$ are common choices). Thus, the scaling of the bandwidth can be ignored.

Choice of classifier. Next, we formalize the classifier C_i with parameters $\boldsymbol{u}_i = (\alpha_i, \boldsymbol{\beta}_i)$. Let $\ell(y, (a, \boldsymbol{\beta}); (\alpha_i, \boldsymbol{\beta}_i))$ denote the loss on a data point with bias $a \in \mathbb{R}$ and interest vector $\boldsymbol{\beta} \in \mathbb{R}^{d-1}$ belonging to class $y \in \{+1, -1\}$. Then, we seek $\boldsymbol{u}_i = (\alpha_i, \boldsymbol{\beta}_i)$ that minimizes

$$\frac{1}{|S_{i-}|} \sum_{j \in S_{i-}} \ell(y = -1, (\alpha_j, \boldsymbol{\beta}_j); (\alpha_i, \boldsymbol{\beta}_i)) + \frac{1}{|S_{i+}|} \sum_{j \in S_{i+}} E_{\boldsymbol{\beta} \sim \mathcal{D}^{\star}_{i+}} \ell(y = +1, (\alpha_j, \boldsymbol{\beta}); (\alpha_i, \boldsymbol{\beta}_i)). \tag{3}$$

The second term is the expected loss over $\mathcal{D}^{\star}_{i+}$, and will generally not have a closed form. Sampling-based approximations of the expected loss can be slow. Instead, we will choose a loss function $\ell(.)$ for which the expected loss over $\mathcal{D}^{\star}_{i+}$ has a closed-form formula.

Specifically, we set

$$\ell(y, (a, \boldsymbol{\beta}); (\alpha_i, \boldsymbol{\beta}_i)) = \max(0, 1 - y \cdot (a + \alpha_i + \boldsymbol{\beta}_i^T \boldsymbol{\beta})).$$

This is a unit-margin hinge loss where $s = a + \alpha_i + \boldsymbol{\beta}_i^T \boldsymbol{\beta}$ measures the similarity between two nodes with embeddings $(a, \boldsymbol{\beta})$ and $(\alpha_i, \boldsymbol{\beta}_i)$. The term $\boldsymbol{\beta}_i^T \boldsymbol{\beta}$ in the score measures the overlap of interests between the two nodes. The term $a + \alpha_i$ is the total propensity to

attract friends irrespective of interests. Higher the score s, greater the similarity between the nodes. We predict a link iff $s > 0$, and the prediction is incorrect if $y \cdot s < 0$ (since $y \in \{+1, -1\}$). Thus, this choice of loss enables a simple and fast link prediction system.

THEOREM 2.1. *The expected loss on the positive class is given by*

$$E_{\boldsymbol{\beta} \sim \mathcal{D}^{\star}_{i+}} \ell(y = +1, (\alpha_j, \boldsymbol{\beta}); (\alpha_i, \boldsymbol{\beta}_i))$$
$$= \frac{1}{|S_{i+}|} \sum_{j \in S_{i+}} \left[(1 - s_{ij}) \cdot \Phi\left(\frac{1 - s_{ij}}{t_i}\right) + t_i \cdot \phi\left(\frac{1 - s_{ij}}{t_i}\right) \right], \quad (4)$$

$$s_{ij} = \alpha_j + \alpha_i + \boldsymbol{\beta}_i^T \boldsymbol{\beta}_j, \quad (5)$$

$$t_i = \sqrt{\boldsymbol{\beta}_i^T \Sigma^{\star}_{i+} \boldsymbol{\beta}_i} \quad (6)$$
$$= \sqrt{\eta_i \cdot \left(\frac{\sum_{j \in S_{i+}} \left(\boldsymbol{\beta}_i^T \boldsymbol{\beta}_j\right)^2}{|S_{i+}|} - \left(\frac{\sum_{j \in S_{i+}} \boldsymbol{\beta}_i^T \boldsymbol{\beta}_j}{|S_{i+}|} \right)^2 \right) + \nu_i \cdot \|\boldsymbol{\beta}_i\|^2},$$

where $\Phi(.)$ and $\phi(.)$ are the cdf and pdf of the standard Normal distribution, and $\Sigma^{\star}_{i+}$ was defined in Eq. 2.

COROLLARY 2.2. *Under the setting of Theorem 2.1, the overall loss in Eq. 3 increases monotonically with t_i.*

Both are proved in the appendix.

For intuition, suppose we fix $\boldsymbol{\beta}_i$ and let $\|\Sigma^{\star}_{i+}\| \to 0$. Then, the density $\mathcal{D}^{\star}_{i+}$ tends to the empirical distribution of the positive class. So the expected loss on point j should reduce to the empirical hinge loss $\max(0, 1 - s_{ij})$. Plugging $\|\Sigma^{\star}_{i+}\| \to 0$ into Eq. 4, we find that $t_i \to 0$, so $\phi((1-s_{ij})/t_i) \to 0$ and $\Phi((1-s_{ij})/t_i) \to \mathbb{1}(1 - s_{ij} > 0)$. Hence, the expected loss becomes $(1 - s_{ij}) \cdot \mathbb{1}(1 - s_{ij} > 0) = \max(0, 1 - s_{ij})$, as desired.

Now, suppose we fix $\Sigma^{\star}_{i+}$ and vary $\boldsymbol{\beta}_i$ such that the interest match $\boldsymbol{\beta}_i^T \boldsymbol{\beta}_j$ is fixed. So s_{ij} remains fixed, and only t_i changes. By Corollary 2.2, among all possible $\boldsymbol{\beta}_i$ with the same interest match, we prefer the one with the lowest t_i. Note that $t_i = \|{\Sigma^{\star}_{i+}}^{1/2} \boldsymbol{\beta}_i\|$. So, the t_i term acts as a regularizer for $\boldsymbol{\beta}_i$, but instead of a norm, it uses a Mahalanobis metric that is specific to i.

Implementation. The above steps were aimed at the local problem. To solve the global problem of finding all node embeddings, NEWS trains all the classifiers C_i jointly. In particular, we seek $\{\boldsymbol{u}_i; i \in [n]\}$ to minimize the sum of losses (Eq. 3) over all nodes i. We use the ADAM optimizer in all our experiments.

NEWS uses two optimizations to speed up processing. First, we never calculate $\Sigma^{\star}_{i+}$ explicitly, since we only need it for t_i (Theorem 2.1). We can compute t_i via simple matrix operations. Second, for the negative class loss (the first term of Eq. 3), we average over a sample of nodes instead of all nodes in S_{i-}. Following [41], we sample node $j \in S_{i-}$ with probability proportional to $d_j^{3/4}$, where d_j is its degree. Note that the choice of sampling scheme is orthogonal to our method, and other schemes can be used. In each mini-batch, we choose one set of samples which we use as the negative class for all nodes in that mini-batch. Then, we only need one matrix multiplication to calculate all negative loss terms. This speeds up the loss computation considerably.

Complexity. To calculate (η_i, ν_i), we need $O(\min(|S_{i+}|^2 \cdot d, |S_{i+}| \cdot d^2))$ time, where d is the embedding dimension. For the expected loss on the positive class (Eq. 4), we need all s_{ij} and t_i, which takes $O(|S_{i+}| \cdot d)$ time. For the negative class, we average the loss over a fixed-size sample of nodes from S_{i-}, and this take $O(d)$ time. Hence, the time taken for every epoch of the optimizer is $\sum_i O(\min(|S_{i+}| \cdot d^2, |S_{i+}|^2 \cdot d)) = O(md^2)$, where $m = \sum_i |S_{i+}|$ is the number of edges in the network.

The embedding requires $O(d)$ space per node, and the calculation of (η_i, ν_i) takes $O(\min(|S_{i+}|^2, d^2))$ space. Hence, the overall space complexity is $O(nd^2)$, where n is the number of nodes in the network. Thus, **NEWS's complexity is linear in the number of nodes and edges.**

Extensions. For directed graph, we can have separate bias and interest vectors for incoming and outgoing edges: $\boldsymbol{u}_i = \left(\alpha_i^{(in)}, \boldsymbol{\beta}_i^{(in)}, \alpha_i^{(out)}, \boldsymbol{\beta}_i^{(out)}\right)$. The necessary modifications to NEWS are straightforward. The positive set S_{i+} becomes the out-edges of i, and the robust distribution $\mathcal{D}^{\star}_{i+}$ is now built from $\{\boldsymbol{\beta}_j^{(in)}; j \in S_{i+}\}$. The score for a directed edge $i \to j$ becomes $s_{ij} = \alpha_i^{(out)} + \alpha_j^{(in)} + \left(\boldsymbol{\beta}_i^{(out)}\right)^T \boldsymbol{\beta}_j^{(in)}$.

For undirected graphs, by symmetry, the minimum loss is achieved when the in- and out-parameters are identical for each node. So, we recover the node embeddings of the undirected NEWS algorithm, but with half the embedding dimension.

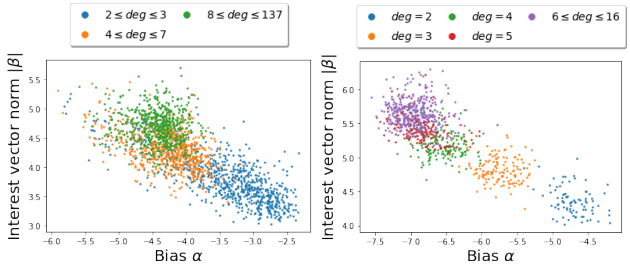
3 ANALYSIS AND SIMULATIONS

NEWS's embedding includes the bias terms α_i alongside the interest vectors $\boldsymbol{\beta}_i$. In contrast, most existing methods do not have bias terms. Here, we show the need for bias terms by exploring their interaction with interest vectors. Further evidence for the importance of bias terms will be shown in Section 4.

Figure 1a plots the norm of $\boldsymbol{\beta}_i$ against α_i for the Deezer social network. As the degree increases, $\|\boldsymbol{\beta}_i\|$ increases and α_i decreases. We see similar patterns for communication and protein interaction networks too. To understand why, we simulated a random graph with $n = 10,000$ nodes and an expected degree of 5 (Fig. 1b).

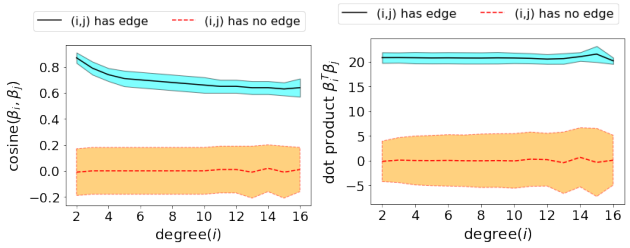
Why $\|\boldsymbol{\beta}\|$ increases with degree. The explanation lies in the correlations between the interest vectors of the nodes. Consider two nodes i and j connected by an edge. We find that the cosine between $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$ decreases with degree (Figure 2a). This is intuitive; as the degree of a node increases, it is harder to have high cosine similarity with all its neighbors. Now, to minimize loss, we should have $\alpha_i + \alpha_j + \boldsymbol{\beta}_i^T \boldsymbol{\beta}_j \gg 0$. When the degree of i increases, the cosine decreases, so we must either increase α_i or $\|\boldsymbol{\beta}_i\|$. The choice depends on the fluctuations in the cosine. In this instance, linked nodes have small cosine fluctuations (Figure 2a). So NEWS chooses to increase $\|\boldsymbol{\beta}_i\|$ such that $\boldsymbol{\beta}_i^T \boldsymbol{\beta}_j$ is nearly constant for all degrees (Fig. 2b). Hence, $\|\boldsymbol{\beta}\|$ increases with degree.

Why α decreases as $\|\boldsymbol{\beta}\|$ increases. Consider nodes i and j that are *not* linked by an edge. Ideally, we should have $s_{ij} := \alpha_i + \alpha_j + \boldsymbol{\beta}_i^T \boldsymbol{\beta}_j \ll 0$. But $\cos(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) \approx 0$ for unlinked node pairs (Figure 2a). This is because for a fixed $\boldsymbol{\beta}_i$, the volume of the cone $\{\boldsymbol{\beta} \in \mathbb{R}^{d-1}; \|\boldsymbol{\beta}\| = 1, \cos(\boldsymbol{\beta}, \boldsymbol{\beta}_i) < -(1 - \epsilon)\}$ decays exponentially with the embedding dimension d. So it is difficult to push the vectors for unlinked node pairs to have a negative cosine. Instead, they behave like random vectors, which are nearly orthogonal in high dimensions [54].



(a) Deezer social network **(b) Random graph**

Figure 1: An inverse relation between $\|\boldsymbol{\beta}_i\|$ and α_i.



(a) Cosine **(b) Dot-product**

Figure 2: Correlations between interest vectors.

Now, as discussed above, $\|\boldsymbol{\beta}_i\|$ grows with the degree of i. So, as the cosine fluctuates around 0, the dot product $\boldsymbol{\beta}_i^T \boldsymbol{\beta}_j$ shows larger fluctuations for high-degree nodes (Figure 2b). Hence, some unlinked node pairs will have a dot-product that is relatively large and positive. To ensure $s_{ij} < 0$, the bias terms for such pairs must be correspondingly large and negative. Hence, α_i becomes more negative as the degree increases (and $\|\boldsymbol{\beta}\|$ increases).

In summary, Figure 1 reflects the relationship between cosines and degrees. The norms of the interest vectors account for this variation in cosines. But without the bias terms, the interest vectors would not have this flexibility. This shows the importance of the bias terms. However, note that t_i also increases with $\|\boldsymbol{\beta}\|$ (Eq. 6), which in turn increases the loss on the positive class. Hence, in general, the relationship between $\|\boldsymbol{\beta}\|$ and α can be complex.

4 EXPERIMENTS

We compared NEWS against competing methods on the accuracy of link prediction, and how it varies with the embedding dimension. We also show results for node classification, and test the importance of the robust distribution and the bias term in NEWS.

Baselines. Our focus is on plain embedding methods, without node or edge features. Since there is a large literature on such methods, we chose methods that worked well in a recent benchmark study [38] and added a few other recent methods. These methods are GraRep [7], HOPE [43], LINE (second order) [51], Node2Vec [17], ProNE [60], VERSE [52], SDNE [55], and Graph2Gauss (G2G) [5]. These cover matrix-based methods, auto-encoders, random-walk methods, and energy-based methods. The unsupervised version of GraphSage [19] performed no better than Graph2Gauss, so it is not shown. We do not consider other convolution-based methods since they are meant for supervised or semi-supervised settings, and need features or side information for training. We used default settings for all methods since we may not know beforehand the tasks for which the embeddings will be used.

Datasets. We ran experiments on **21 real-world datasets.** These include networks based on social interactions (Deezer, Flickr, Blog Catalog, and Youtube), citations (Cora and DBLP), location-based connections (Gowalla), product co-purchases (Amazon), collaborations (four Arxiv networks, and Youtube group memberships), biology (Protein interactions, and Reactome), financial relations (Prosper lending), transportation (US airports and Texas roads), communications (Enron and EU Emails), and other networks (Wordnet) [16, 24, 29, 30, 34, 42, 59]. We made all networks undirected and removed self-loops.

4.1 Link Prediction

Since the goal of node embeddings is to capture the network structure, link prediction accuracy is the natural metric for comparing algorithms.

Experimental setup. For each network, we used 80% of the edges as the training set. We used the remaining 20% of the edges as positive test examples, and added random node pairs as negative test examples. Specifically, for each node i with a positive example (i, j), we created 100 node pairs (i, j') with nodes j' chosen randomly.

For each algorithm, we computed embeddings from the training set, and used these to rank the test node pairs. For NEWS, we used Eq. 5 to score the test pairs (Node2Vec has a similar formula). For the other methods, we trained a neural network with two hidden layers to score the test pairs. For Graph2Gauss, the neural network outperformed the energy-based score proposed by the authors. We do not compare against non-embedding link prediction heuristics, since they underperform our baselines [38].

For each node and each algorithm, we ranked all test pairs with that node, and calculated the area under the precision-recall curve (AUPRC). The AUPRC is a standard measure for imbalanced settings [13]. Note that the AUPRC measures the embedding's accuracy *for each node*. Thus, better the personalization, higher the AUPRC.

Results. Table 1 reports the trimmed mean of the AUPRC scores for nodes grouped by degree. For each row, we circle the methods that are within 0.05 of the best AUPRC, and underline those that are worse at the $p < 0.01$ level. Methods that did not finish are shown by crosses. We make two observations:

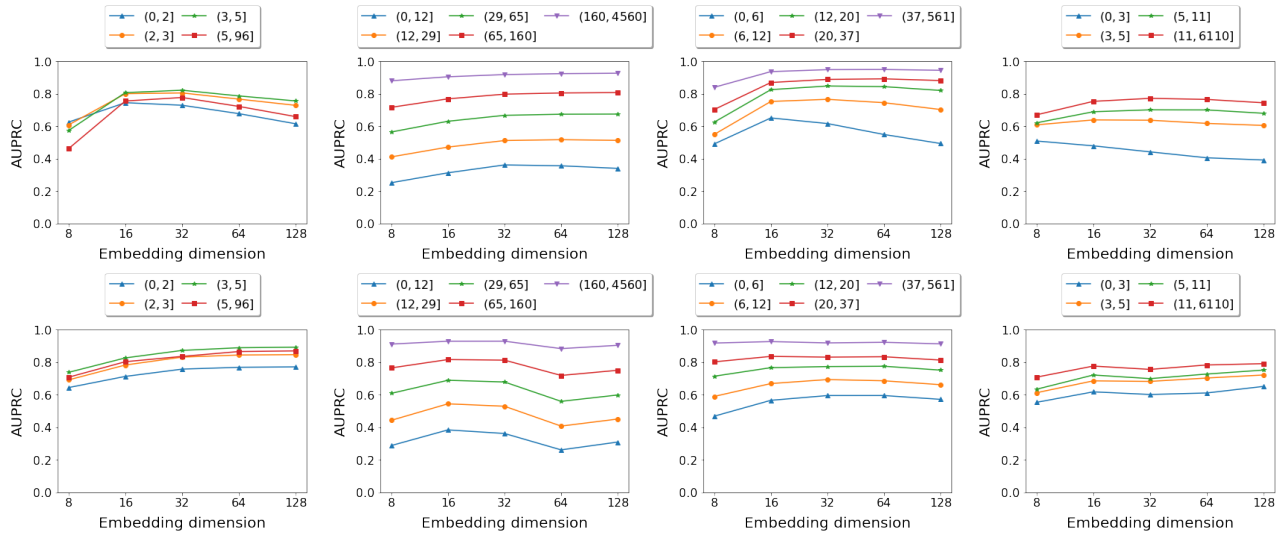
- **NEWS is among the best performing methods in almost all cases.** Note that NEWS only uses first-order proximity, while the baselines use second and higher-order proximity. Even with this severe constraint, NEWS is better than most baselines and comparable to the best method on any dataset.
- **NEWS performs well even for low-degree nodes.** These are the nodes for which extra assumptions of higher-order proximity can have the most impact. NEWS works well even without such assumptions. This points to the importance of NEWS's robust approach.

Varying the embedding dimension. Figure 3 shows the accuracy of NEWS and VERSE as the embedding dimension varies from

Degree	G2G	GraRep	HOPE	LINE	Node2Vec	ProNE	SDNE	VERSE	**NEWS**
Cora (1, 434 **nodes**, 4, 256 **edges**)									
(0, 2]	0.04	0.48	0.10	0.17	0.69	0.48	0.25	0.70	0.76
(2, 3]	0.04	0.47	0.31	0.34	0.76	0.52	0.31	0.67	0.79
(3, 5]	0.04	0.44	0.26	0.23	0.66	0.46	0.28	0.63	0.70
(5, 10]	0.05	0.36	0.28	0.23	0.59	0.37	0.30	0.54	0.69
(10, 140]	0.09	0.37	0.36	0.27	0.55	0.29	0.25	0.49	0.54
Amazon (334, 863 **nodes**, 925, 872 **edges**)									
(0, 2]	0.02	0.38	0.02	0.46	0.61	0.73	0.02	0.90	0.87
(2, 3]	0.02	0.46	0.02	0.66	0.70	0.86	0.02	0.99	0.98
(3, 4]	0.02	0.51	0.02	0.74	0.69	0.91	0.02	1.00	1.00
(4, 7]	0.02	0.51	0.02	0.73	0.56	0.92	0.02	1.00	0.99
(7, 428]	0.04	0.56	0.04	0.73	0.50	0.92	0.04	0.99	0.98
Deezer (28, 281 **nodes**, 92, 752 **edges**)									
(0, 2]	0.02	0.19	0.05	0.14	0.33	0.33	0.05	0.42	0.33
(2, 4]	0.02	0.16	0.09	0.19	0.42	0.40	0.08	0.48	0.41
(4, 7]	0.02	0.18	0.14	0.25	0.48	0.46	0.13	0.51	0.46
(7, 12]	0.03	0.21	0.22	0.32	0.54	0.50	0.21	0.56	0.52
(12, 137]	0.05	0.37	0.43	0.48	0.65	0.59	0.40	0.66	0.65
Arxiv (Gen. Rel.) (5, 241 **nodes**, 14, 484 **edges**)									
(0, 3]	0.03	0.80	0.05	0.55	0.85	0.81	0.36	0.96	0.95
(3, 5]	0.03	0.71	0.15	0.53	0.89	0.76	0.37	0.96	0.94
(5, 10]	0.03	0.68	0.32	0.52	0.83	0.72	0.46	0.84	0.91
(10, 21]	0.05	0.74	0.47	0.60	0.80	0.70	0.52	0.80	0.86
(21, 78]	0.15	0.93	0.82	0.94	0.92	0.90	0.89	0.97	0.99
US Airports (1, 574 **nodes**, 17, 215 **edges**)									
(0, 5]	0.07	0.31	0.18	0.28	0.26	0.19	0.18	0.31	0.27
(5, 12]	0.10	0.38	0.27	0.30	0.33	0.35	0.30	0.45	0.44
(12, 21]	0.14	0.51	0.44	0.48	0.35	0.47	0.55	0.53	0.54
(21, 57]	0.26	0.53	0.51	0.57	0.39	0.55	0.55	0.58	0.58
(57, 295]	0.57	0.72	0.75	0.71	0.56	0.71	0.76	0.70	0.73
Prosper Lending Network (89, 269 **nodes**, 3, 330, 022 **edges**)									
(0, 11]	0.03	0.18	0.08	0.38	0.14	0.19	0.14	0.43	0.51
(11, 25]	0.07	0.31	0.19	0.50	0.15	0.44	0.31	0.56	0.58
(25, 48]	0.11	0.44	0.33	0.61	0.17	0.60	0.49	0.67	0.65
(48, 99]	0.18	0.56	0.48	0.72	0.21	0.72	0.66	0.76	0.72
(99, 5503]	0.35	0.73	0.68	0.84	0.32	0.85	0.82	0.85	0.81
DBLP (317, 080 **nodes**, 1, 049, 866 **edges**)									
(0, 2]	0.02	0.63	0.03	0.68	0.79	0.84	0.03	0.98	0.97
(2, 3]	0.02	0.69	0.03	0.81	0.87	0.90	0.04	1.00	1.00
(3, 6]	0.03	0.69	0.04	0.81	0.89	0.89	0.06	0.99	1.00
(6, 10]	0.03	0.67	0.06	0.78	0.90	0.85	0.11	0.98	0.99
(10, 266]	0.06	0.71	0.17	0.75	0.90	0.81	0.25	0.95	0.98
Enron (36, 692 **nodes**, 183, 831 **edges**)									
(0, 3]	0.06	0.56	0.20	0.74	0.74	0.58	0.32	0.91	0.93
(3, 7]	0.06	0.63	0.17	0.80	0.91	0.63	0.41	0.92	0.99
(7, 17]	0.06	0.53	0.22	0.75	0.90	0.57	0.48	0.83	0.92
(17, 44]	0.08	0.56	0.32	0.74	0.89	0.57	0.61	0.77	0.88
(44, 1317]	0.19	0.60	0.53	0.74	0.89	0.63	0.70	0.77	0.89
Flickr (80, 513 **nodes**, 5, 899, 882 **edges**)									
(0, 12]	×	0.16	0.15	0.25	0.34	0.16	0.14	0.37	0.36
(12, 29]	×	0.31	0.26	0.41	0.50	0.37	0.30	0.54	0.52
(29, 65]	×	0.48	0.41	0.60	0.65	0.57	0.51	0.69	0.67
(65, 160]	×	0.66	0.60	0.77	0.78	0.75	0.73	0.81	0.80
(160, 4560]	×	0.87	0.84	0.92	0.91	0.91	0.91	0.93	0.92
Blog Catalog (10, 312 **nodes**, 333, 983 **edges**)									
(0, 7]	0.15	0.48	0.51	0.56	0.09	0.32	0.53	0.55	0.45
(7, 15]	0.16	0.52	0.54	0.59	0.11	0.41	0.56	0.57	0.52
(15, 30]	0.21	0.60	0.61	0.65	0.19	0.53	0.62	0.62	0.61
(30, 66]	0.27	0.66	0.69	0.71	0.31	0.63	0.70	0.70	0.69
(66, 3162]	0.44	0.78	0.81	0.81	0.57	0.78	0.81	0.80	0.80

Degree	G2G	GraRep	HOPE	LINE	Node2Vec	ProNE	SDNE	VERSE	**NEWS**
Gowalla (196, 591 **nodes**, 950, 327 **edges**)									
(0, 4]	0.03	0.12	0.09	0.42	0.56	0.36	0.09	0.69	0.65
(4, 9]	0.03	0.20	0.10	0.54	0.72	0.48	0.13	0.77	0.79
(9, 17]	0.03	0.26	0.13	0.59	0.77	0.53	0.20	0.78	0.84
(17, 35]	0.04	0.34	0.19	0.64	0.80	0.59	0.34	0.79	0.86
(35, 14118]	0.11	0.54	0.41	0.74	0.88	0.68	0.58	0.81	0.90
Youtube (1, 134, 890 **nodes**, 2, 987, 624 **edges**)									
(0, 3]	×	×	×	0.30	0.37	0.18	0.30	0.56	0.41
(3, 5]	×	×	×	0.41	0.62	0.25	0.34	0.66	0.59
(5, 12]	×	×	×	0.48	0.70	0.31	0.40	0.70	0.66
(12, 22971]	×	×	×	0.70	0.84	0.56	0.61	0.82	0.78
Arxiv (Astrophysics) (18, 771 **nodes**, 198, 050 **edges**)									
(0, 9]	0.02	0.58	0.03	0.75	0.94	0.54	0.17	0.90	0.99
(9, 20]	0.03	0.59	0.06	0.77	0.97	0.57	0.47	0.86	0.97
(20, 35]	0.03	0.59	0.14	0.76	0.95	0.60	0.61	0.83	0.94
(35, 57]	0.04	0.64	0.39	0.78	0.95	0.66	0.69	0.86	0.95
(57, 489]	0.06	0.63	0.51	0.76	0.93	0.67	0.70	0.80	0.92
Arxiv (Cond. Mat.) (38, 741 **nodes**, 58, 595 **edges**)									
(0, 2]	0.02	0.06	0.05	0.06	0.59	0.64	0.03	0.76	0.73
(2, 3]	0.02	0.06	0.05	0.05	0.53	0.71	0.03	0.84	0.80
(3, 5]	0.02	0.05	0.05	0.04	0.39	0.74	0.03	0.88	0.82
(5, 96]	0.03	0.07	0.06	0.05	0.22	0.71	0.03	0.86	0.78
Protein Interactions (56, 688 **nodes**, 793, 632 **edges**)									
(0, 6]	0.17	0.44	0.07	0.52	0.67	0.38	0.20	0.60	0.61
(6, 12]	0.29	0.52	0.18	0.62	0.73	0.52	0.41	0.71	0.76
(12, 20]	0.37	0.62	0.34	0.71	0.78	0.62	0.56	0.78	0.85
(20, 37]	0.47	0.72	0.52	0.79	0.83	0.73	0.70	0.85	0.89
(37, 561]	0.68	0.87	0.74	0.90	0.90	0.87	0.87	0.93	0.95
Reactome (6, 229 **nodes**, 146, 160 **edges**)									
(0, 6]	0.03	0.52	0.07	0.63	0.92	0.42	0.29	0.82	0.93
(6, 16]	0.04	0.67	0.12	0.82	0.96	0.53	0.51	0.87	0.96
(16, 35]	0.07	0.81	0.38	0.92	0.98	0.76	0.82	0.94	0.98
(35, 88]	0.18	0.92	0.86	0.97	0.98	0.91	0.95	0.97	0.99
(88, 700]	0.35	0.98	0.98	1.00	0.97	0.99	0.99	0.99	0.99
Wordnet (146, 005 **nodes**, 656, 999 **edges**)									
(0, 3]	0.10	0.37	0.06	0.75	0.78	0.76	0.08	0.97	0.98
(3, 4]	0.10	0.46	0.09	0.86	0.90	0.82	0.12	0.99	1.00
(4, 6]	0.10	0.46	0.12	0.87	0.91	0.82	0.16	0.98	1.00
(6, 11]	0.09	0.43	0.15	0.84	0.94	0.79	0.19	0.97	1.00
(11, 821]	0.12	0.43	0.28	0.80	0.92	0.75	0.32	0.94	0.98
Email-EU (265, 009 **nodes**, 364, 481 **edges**)									
(0, 3]	0.31	0.78	0.87	0.90	0.31	0.28	0.88	0.94	0.86
(3, 8]	0.35	0.79	0.88	0.91	0.78	0.23	0.87	0.95	0.95
(8, 5030]	0.33	0.72	0.78	0.82	0.82	0.44	0.75	0.89	0.88
HepTh (9, 875 **nodes**, 25, 973 **edges**)									
(0, 2]	0.02	0.74	0.03	0.61	0.83	0.83	0.30	0.94	0.94
(2, 5]	0.03	0.70	0.05	0.61	0.83	0.77	0.35	0.88	0.90
(5, 9]	0.03	0.61	0.09	0.52	0.82	0.68	0.44	0.83	0.86
(9, 17]	0.03	0.57	0.15	0.44	0.76	0.54	0.40	0.74	0.72
(17, 63]	0.05	0.63	0.45	0.56	0.77	0.61	0.56	0.73	0.75
Roads (TX) (1, 379, 917 **nodes**, 1, 921, 660 **edges**)									
(0, 2]	×	0.01	0.01	0.02	0.48	0.52	0.50	0.96	0.79
(2, 3]	×	0.01	0.01	0.03	0.69	0.67	0.50	1.00	0.94
(3, 11]	×	0.01	0.02	0.05	0.70	0.78	0.50	1.00	0.98
Groups (Youtube) (124, 325 **nodes**, 293, 360 **edges**)									
(0, 3]	0.04	0.24	0.39	0.40	0.37	0.22	0.39	0.60	0.44
(3, 5]	0.04	0.31	0.38	0.48	0.61	0.30	0.38	0.67	0.64
(5, 11]	0.04	0.35	0.38	0.54	0.65	0.36	0.41	0.69	0.70
(11, 6110]	0.09	0.49	0.43	0.66	0.72	0.50	0.49	0.76	0.77

Table 1: ***Link prediction accuracy:*** **We calculate the area under the precision-recall curve (AUPRC) for link prediction for each node. We then split the nodes into equal-sized bins based on degree, and report the trimmed mean of AUPRC scores for nodes in each bin. We circle the methods that are within 0.05 of the best AUPRC, and underline the ones that are statistically significantly worse by at least 0.05 (at the $p < 0.01$ level). NEWS is seen to be comparable to the best higher-order proximity method for almost all datasets and degree ranges, even though NEWS uses only first-order proximity.**



(a) Arxiv (Cond. Mat.) (b) Flickr (c) Protein Interactions (d) Groups (Youtube)

Figure 3: Varying the embedding dimension: The top panel shows NEWS, while the bottom panel shows VERSE. Both show similar patterns and plateau beyond an embedding dimension of $d = 32$.

	Lift
$deg \leq 2$	14.6%
$2 < deg \leq 3$	15.5%
$3 < deg \leq 5$	17.3%
$5 < deg$	27.9%

(a) Arxiv (Cond. Mat.)

	Lift
$deg \leq 3$	−2.5%
$3 < deg \leq 5$	22.1%
$5 < deg \leq 11$	23.3%
$11 < deg$	21.3%

(b) Groups (Youtube)

	Lift
$deg \leq 12$	27.7%
$12 < deg \leq 29$	19.9%
$29 < deg \leq 65$	10.2%
$65 < deg \leq 160$	3.1%
$160 < deg$	0.4%

(c) Flickr

	Lift
$deg \leq 6$	22.5%
$6 < deg \leq 12$	11.7%
$12 < deg \leq 20$	7.2%
$20 < deg \leq 37$	5.0%
$37 < deg$	2.1%

(d) Protein Interactions

Table 2: Lift of NEWS over not using a robust density.

	Lift
(0, 2]	50.7%
(2, 3]	67.9%
(3, 5]	85.0%
(5, 96]	108.4%

(a) Arxiv (Cond. Mat.)

	Lift
(0, 3]	68.8%
(3, 5]	47.2%
(5, 11]	45.0%
(11, 6110]	41.6%

(b) Groups (Youtube)

	Lift
(0, 12]	572.6%
(12, 29]	143.8%
(29, 65]	39.4%
(65, 160]	10.8%
(160, 4560]	2.0%

(c) Flickr

	Lift
(0, 6]	196.4%
(6, 12]	103.8%
(12, 20]	59.5%
(20, 37]	29.2%
(37, 561]	6.3%

(d) Protein Interactions

Table 3: Lift of NEWS over not using a bias term.

$d = 8$ to $d = 128$. In both cases, the accuracy plateaus for $d \geq 32$, so we chose $d = 32$ for our experiments. For NEWS, the accuracy on low-degree nodes can dip as d increases. This is because limited data in higher dimensions increases the chances of overfitting. Higher-order proximity methods converge to their assumed similarity matrix as d increases, so their accuracy depends on the quality of that assumption.

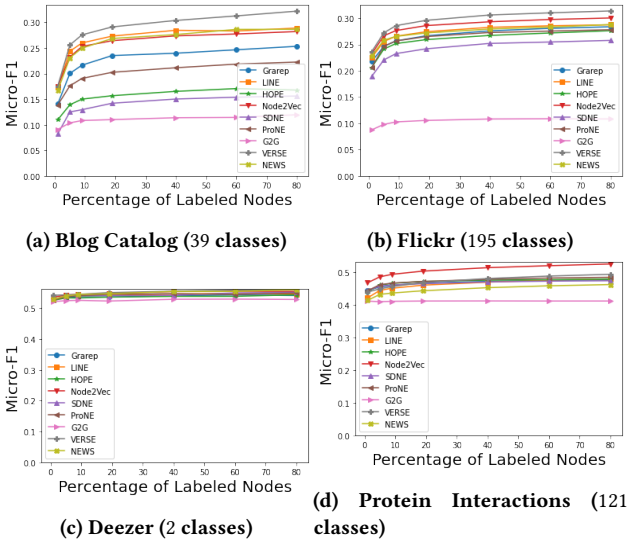
4.2 Ablation study

Next, we show the importance of the robust smoothed distribution and the bias terms in NEWS.

Importance of robust smoothing. Recall that the main difficulty with first-order proximity stems from low-degree nodes, for which we have little data. NEWS creates a robust distribution (Eq. 2) to account for the lack of data. We ran an experiment replacing it with the empirical distribution. This is the same as setting $t_i \to 0$ in Eq. 4. For both the robust and empirical distributions, we calculate the AUPRC trimmed mean for each degree range. Table 2 shows the percentage lift achieved by the robust distribution.

For each of the four datasets, **the robust distribution yields > 20% lift.** Also, we see improvements for nodes of all degrees, and not only the low-degree nodes. The reason is that low-degree nodes predominate in networks and often connect to high-degree nodes. So, better embeddings for low-degree nodes lead to better embeddings for other nodes too.

Importance of bias terms. Recall that NEWS's embedding is of the form $\boldsymbol{u}_i = (\alpha_i \in \mathbb{R}, \boldsymbol{\beta}_i \in \mathbb{R}^{d-1})$, where α_i is the bias term for node i. In this experiment, we find the best embedding without bias terms: $\boldsymbol{u}_i = (\boldsymbol{\beta}_i \in \mathbb{R}^d)$. Figure 3 shows the lift of NEWS over the version without bias terms. Across all four datasets, the bias terms



(a) Blog Catalog (39 classes) (b) Flickr (195 classes)

(c) Deezer (2 classes) (d) Protein Interactions (121 classes)

Figure 4: Classification accuracy versus % labeled nodes.

provide a significant lift in accuracy. The reason is that the bias terms ensure a negative score for node pairs that are not linked by an edge, as discussed in Section 3.

4.3 Node Classification

Here, we used the node embeddings as features in a random forest for node classification. We do not use any extra features because we only want to compare embedding methods. Figure 4 shows that NEWS is comparable to higher-order proximity methods.

5 RELATED WORK

We will review the literature on embeddings and fairness.

Embeddings. Our focus is on graphs with no extra features. Existing embedding methods for this problem can be split into three groups. *First-order proximity* methods use only the link structure of the graph. These include Locally Linear Maps [48], Laplacian Eigenmaps [3], Graph Factorization [2], and Probabilistic Matrix Factorization [49]. *Second-order proximity* methods measure similarity between nodes based on their neighborhoods. Examples include SDNE [55], which encodes the neighborhood via a deep autoencoder, and LINE [51] and ProNE [60], which encode it via a context vector. *Higher-order proximity* methods consider similarities between nodes that are farther apart in the network. DeepWalk [44] and Node2Vec [17] do this via random walks. GraRep [7] also considers a random-walk transition matrix, but then transforms it and factorizes it. NetMF [46] uses a similar procedure on a different matrix. HOPE [43] factorizes similarity matrices constructed from common link prediction heuristics such as the Katz measure. VERSE[52] uses a personalized pagerank similarity matrix.

When node or edge features are available, one can use Graph Neural Networks and its many variants [19, 53, 58]. Recent surveys [18, 57] cover these aspects in detail. Some methods construct embeddings to jointly preserve proximity in terms of network topology as well as attribute similarity [22]. SIGNet [23] is an embedding for networks where edges have signs (e.g., trust versus no-trust relationships). We focus on plain embeddings without node/edge attributes, so these works are orthogonal to ours. Finally, there is rich literature on latent variable inference under graph generative models (see [28, 37] for a survey and recent results). However, the assumption of a known generative model may not hold in practice.

Fairness. Fair algorithms trade off overall accuracy against a fairness metric defined over groups of individuals [4, 20], pairs of similar individuals [15], or for individuals under counterfactual conditions [31]. Group memberships are often encoded as individual attributes (called the "sensitive" attributes). FairGNN [12] estimates the sensitive attributes when they are missing. GNNs combining fairness and stability are explored in [1] and generalization bounds analyzed in [36]. Fairwalk [47] modifies the random walks of Node2Vec [17] to capture diverse neighborhoods. Other work aims to ensure zero mutual information between node embeddings and sensitive attributes [6], or to establish individual fairness given a similarity measure [14, 45]. However, unlike our setting, these works assume the availability of the sensitive attribute or a similarity metric between sensitive nodes.

There is also work on fairness when the sensitive attribute is noisy [8, 56] or unknown [21, 32, 39]. These works typically apply a worst-case robust optimization over the unknown value of the sensitive attribute or their underlying distribution (though [25] use posterior sampling). These sensitive attribute is also unknown in our setting. However, our problem is different; we aim to remove one source of systematic bias that comes from the assumptions made by higher-order proximity methods.

6 CONCLUSIONS

There is significant interest in algorithms that are both accurate and fair. One potential source of unfairness lies in the algorithm's assumptions. For node embedding methods, the assumptions are about the similarity of unlinked nodes. Such similarity assumptions govern many popular "higher-order" embedding methods. But in seeking the highest overall accuracy, they may unintentionally bias against a minority of nodes with atypical linkage patterns. We present a method, called NEWS, that avoids making any similarity assumptions without sacrificing much accuracy.

NEWS's embedding for each node represents the parameter vector of a *robust and personalized* classifier for that node. Each node's classifier is trained to differentiate between that node's neighbors and the rest of the network. We make no assumptions about the similarity of unlinked nodes. The robustness ensures stable embeddings for low-degree nodes, for which the classifier has limited training data. The personalization guarantees that each node's embedding is the best possible, given the embeddings of all other nodes. Together, they remove potential sources of bias while still achieving accuracy comparable to the best higher-order methods.

NEWS can be extended in several directions. One is to incorporate node or edge features within the framework of the personalized classifiers. One possibility is to learn weights for these features alongside the node embedding features we currently use. A second extension is to try more complex classifiers for high-degree nodes. For such nodes, the greater data availability makes it possible to fit such classifiers without overfitting.

REFERENCES

[1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a Unified Framework for Fair and Stable Graph Representation Learning. In *UAI*.

[2] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. 2013. Distributed Large-scale Natural Graph Factorization. In *WWW '13*.

[3] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* (2003).

[4] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research* (2018).

[5] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *ICLR*.

[6] Avishek Joey Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *ICML*.

[7] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *CIKM*.

[8] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. 2021. Fair Classification with Noisy Protected Attributes: A Framework with Provable Guarantees. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 1349–1361.

[9] Deepayan Chakrabarti and Christos Faloutsos. 2006. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys (CSUR)* 38, 1 (2006), 2. http://dl.acm.org/citation.cfm?id=1132954

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (June 2002), 321–357.

[11] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA.

[12] Enyan Dai and Suhang Wang. 2021. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. In *WSDM*.

[13] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *ICML*.

[14] Yushun Dong, Jian Kang, Hanghang Tong, and Jundong Li. 2021. Individual Fairness for Graph Neural Networks: A Ranking based Approach. In *KDD*. 300–310. https://doi.org/10.1145/3447548.3467266

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. 2011. Fairness Through Awareness. In *ITCS*.

[16] J. Leskovec E. Cho, S. A. Myers. 2011. Friendship and Mobility: Friendship and Mobility: User Movement in Location-Based Social Networks. In *KDD*.

[17] Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In *KDD*.

[18] William L. Hamilton. 2020. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14, 3 (2020), 1–159.

[19] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

[20] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NeurIPS*.

[21] Tatsunori B Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness Without Demographics in Repeated Loss Minimization. In *ICML*. 10.

[22] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label Informed Attributed Network Embedding. In *WSDM*.

[23] Mohammad Raihanul Islam, B. Aditya Prakash, and Naren Ramakrishnan. 2018. SIGNet: Scalable Embeddings for Signed Networks. In *PAKDD*.

[24] J. Kleinberg J. Leskovec and C. Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM TKDD* 1, 1 (2007).

[25] Ajil Jalal, Sushrut Karmalkar, Jessica Hoffmann, Alexandros G Dimakis, and Eric Price. 2021. Fairness for Image Generation with Uncertain Sensitive Attributes. In *ICML*. 12.

[26] Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6, 1 (March 2019), 27.

[27] Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. 2018. Cost Sensitive Learning of Deep Feature Representations from Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems* 29, 9 (2018).

[28] Bomin Kim, Kevin H. Lee, Lingzhou Xue, and Xiaoyue Niu. 2018. A review of dynamic network models with latent variables. *Statistics surveys* 12 (2018), 105–135. https://doi.org/10.1214/18-SS121

[29] B. Klimt and Y. Yang. 2004. Introducing the Enron corpus. In *CEAS conference*.

[30] Jérôme Kunegis. 2013. KONECT: The Koblenz Network Collection. In *WWW*. 1343–1350.

[31] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2018. Counterfactual Fairness. In *NeurIPS*.

[32] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. 2020. Fairness without Demographics through Adversarially Reweighted Learning. In *NeurIPS*. 13.

[33] Olivier Ledoit and Michael Wolf. 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis* 88, 2 (Feb. 2004), 365–411.

[34] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. 2009. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics* 6, 1 (2009), 29–123.

[35] Ulrike von Luxburg, Agnes Radl, and Matthias Hein. 2014. Hitting and Commute Times in Large Random Neighborhood Graphs. *JMLR* 15, 52 (2014), 1751–1798.

[36] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2021. Subgroup Generalization and Fairness of Graph Neural Networks. In *NeurIPS*. http://arxiv.org/abs/2106.15535 arXiv: 2106.15535.

[37] Xueyu Mao, Purnamrita Sarkar, and Deepayan Chakrabarti. 2021. Estimating Mixed Memberships with Sharp Eigenvector Deviations. *J. Amer. Statist. Assoc.* 116, 536 (2021), 1928–1940.

[38] Alexandru Mara, Jefrey Lijffijt, and Tijl De Bie. 2020. Benchmarking Network Embedding Models for Link Prediction: Are We Making Progress?. In *DSAA*. 138–147.

[39] Natalia L. Martinez, Martin A. Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. 2021. Blind Pareto Fairness and Subgroup Robustness. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 7492–7501. https://proceedings.mlr.press/v139/martinez21a.html ISSN: 2640-3498.

[40] Ninareh Mehrabi, Fred Morstatter, Nanyun Peng, and Aram Galstyan. 2019. Debiasing community detection: the importance of lowly connected nodes. In *ASONAM*. 509–512.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

[42] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*. San Diego, CA.

[43] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric Transitivity Preserving Graph Embedding. In *KDD*.

[44] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *KDD*.

[45] Felix Petersen, Debarghya Mukherjee, Yuekai Sun, and Mikhail Yurochkin. 2021. Post-processing for Individual Fairness. In *NeurIPS*. https://arxiv.org/abs/2110.13796v1

[46] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2017. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *WSDM*.

[47] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *IJCAI*.

[48] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* (2000).

[49] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*.

[50] B. W. Silverman. 1986. *Density estimation for statistics and data analysis*. Number 26 in Monographs on statistics and applied probability. Chapman & Hall/CRC, London.

[51] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*.

[52] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. 2018. VERSE: Versatile Graph Embeddings from Similarity Measures. In *WWW*. 539–548.

[53] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[54] Roman Vershynin. 2018. *High-Dimensional Probability*. Cambridge University Press.

[55] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural Deep Network Embedding. In *KDD*.

[56] Serena Wang, Harikrishna Narasimhan, Maya Gupta, Wenshuo Guo, Andrew Cotter, and Michael I Jordan. 2020. Robust Optimization for Fairness with Noisy Protected Groups. In *NeurIPS*. 14.

[57] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32 (Jan. 2021), 4–24.

[58] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.

[59] J. Yang and J. Leskovec. 2012. Defining and Evaluating Network Communities based on Ground-truth. In *ICDM*.

[60] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. ProNE: Fast and Scalable Network Representation Learning. In *IJCAI*. Macao, China.

APPENDIX

Proof of Theorem 2.1. The expected loss on the positive class $E_{\boldsymbol{\beta}\sim\mathcal{D}^{\star}_{i+}}\ell(y=+1,(\alpha_j,\boldsymbol{\beta});(\alpha_i,\boldsymbol{\beta}_i))$ equals

$$\begin{aligned}
&\frac{1}{|S_{i+}|}\sum_{j\in S_{i+}} E_{\boldsymbol{\beta}\sim\mathcal{N}(\boldsymbol{\beta}_j,\Sigma^{\star}_{i+})}\max(0,1-\alpha_j-\alpha_i-\boldsymbol{\beta}_i^T\boldsymbol{\beta})\\
=&\frac{1}{|S_{i+}|}\sum_{j\in S_{i+}} E_{z\sim\mathcal{N}(1-\alpha_j-\alpha_i-\boldsymbol{\beta}_i^T\boldsymbol{\beta}_j,\boldsymbol{\beta}_i^T\Sigma^{\star}_{i+}\boldsymbol{\beta}_i)}\max(0,z)\\
=&\frac{1}{|S_{i+}|}\sum_{j\in S_{i+}}\left[(1-s_{ij})\cdot\Phi\left(\frac{1-s_{ij}}{t_i}\right)+t_i\cdot\phi\left(\frac{1-s_{ij}}{t_i}\right)\right],
\end{aligned}$$

where s_{ij} and t_i are defined in the theorem statement. Furthermore,

$$\begin{aligned}
t_i &= \sqrt{\boldsymbol{\beta}_i^T\Sigma^{\star}_{i+}\boldsymbol{\beta}_i} = \sqrt{\eta_i\cdot\boldsymbol{\beta}_i^T\hat{\Sigma}_{i+}\boldsymbol{\beta}_i+\nu_i\cdot\|\boldsymbol{\beta}_i\|^2}\\
&= \sqrt{\eta_i\cdot\left(\frac{\sum_{j\in S_{i+}}\left(\boldsymbol{\beta}_i^T\boldsymbol{\beta}_j\right)^2}{|S_{i+}|}-\left(\frac{\sum_{j\in S_{i+}}\boldsymbol{\beta}_i^T\boldsymbol{\beta}_j}{|S_{i+}|}\right)^2\right)+\nu_i\cdot\|\boldsymbol{\beta}_i\|^2}.
\end{aligned}$$

□

Proof of Corollary 2.2. The first partial derivative of the positive class loss (Eq. 4) with respect to t_i is

$$\frac{1}{\|S_{i+}\|}\sum_{j\in S_{i+}}\phi((1-s_{ij})/t_i)>0,$$

where we use the fact that $\Phi'(x)=\phi(x)$ and $\phi'(x)=-x\phi(x)$. The negative class loss does not depend on t_i. □