Asymptotics of ℓ_2 Regularized Network Embeddings

Anonymous Author(s)
Affiliation
Address
email

Abstract

A common approach to solving prediction tasks on large networks, such as node classification or link prediction, begin by learning a Euclidean embedding of the nodes of the network, from which traditional machine learning methods can then be applied. This includes methods such as DeepWalk and node2vec, which learn embeddings by optimizing stochastic losses formed over subsamples of the graph at each iteration of stochastic gradient descent. In this paper, we study the effects of adding an ℓ_2 penalty of the embedding vectors to the training loss of these types of methods. We prove that, under some exchangeability assumptions on the graph, this asymptotically leads to learning a graphon with a nuclear-norm-type penalty, and give guarantees for the asymptotic distribution of the learned embedding vectors. In particular, the exact form of the penalty depends on the choice of subsampling method used as part of stochastic gradient descent. We also illustrate empirically that concatenating node covariates to ℓ_2 regularized node2vec embeddings leads to comparable, when not superior, performance to methods which incorporate node covariates and the network structure in a non-linear manner.

1 Introduction

Network embedding methods [see e.g 6, 23–26, 44, 54, 60] aim to find a latent representation of the nodes of the network within Euclidean space, in order to facilitate the solution of tasks such as node classification and link prediction, by using the produced embeddings as features for machine learning algorithms designed for Euclidean data. For example, they can be used for recommender systems in social networks, or to predict whether two proteins should be linked in a protein-protein interaction graph. Generally, such methods obtain state-of-the-art performance on these type of tasks.

A classical approach to learning a network embedding is to use the principal components of the Laplacian of the network [6]; however, this approach is computationally prohibitive for large datasets. In order to scale to large datasets, methods such as DeepWalk [44] and node2vec [24] learn embeddings via optimizing a loss formed over stochastic subsamples of the graph. Letting $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected graph, and writing $\omega_u \in \mathbb{R}^d$ for the embedding of a vertex $u \in \mathcal{V}$ with embedding dimension d such that $d \ll |\mathcal{V}|$, these methods learn embeddings by iterating over the following process: we take a random walk $(u_i)_{i \leq k}$ over the graph, form a loss

$$\mathcal{L} = -\sum_{i=1}^{k} \sum_{j\,:\,|j-i| \leq W} \log(\sigma\langle \omega_{u_i}, \omega_{u_j}\rangle) - \sum_{i=1}^{k} \sum_{l=1}^{L} \mathbb{E}_{v_l \sim \mathcal{N}(\cdot|u_i)} \Big[\log(1 - \sigma(\langle \omega_{u_i}, \omega_{v_l}\rangle)) \Big], \quad (1)$$

and then perform gradient updates of the form $\omega_u \leftarrow \omega_u - \eta \nabla_{\omega_u} \mathcal{L}$ for some step size $\eta > 0$. Here W is a window size, $N(\cdot|v)$ a negative sampling distribution for vertex v and $\sigma(y) := (1 + e^{-y})^{-1}$ the sigmoid function. LINE [54] uses a similar approach, except that the random walk is replaced by sampling edges uniformly from the edge set of the graph.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

One can generalize (1) to consider embeddings learned via stochastic updates

$$\omega_u \leftarrow \omega_u - \eta \nabla_{\omega_u} \mathcal{L} \quad \text{where} \quad \mathcal{L} = \sum_{(i,j)\in\mathcal{P}} \ell_{\mathcal{P}}(\langle \omega_i, \omega_j \rangle) + \sum_{(i,j)\in\mathcal{N}} \ell_{\mathcal{N}}(\langle \omega_i, \omega_j \rangle), \tag{2}$$

$\mathcal{P}, \mathcal{N} \subseteq \mathcal{V} \times \mathcal{V}$ are random subsamples of the graph called positive and negative samples respectively [see e.g 29], and $\ell_P(\cdot)$ and $\ell_N(\cdot)$ are chosen to force the embeddings of vertex pairs appearing in $\mathcal{P}$ close together, and those in $\mathcal{N}$ far away. Usually one takes $\ell_{\mathcal{P}}(y) = -\log \sigma(y)$ and $\ell_{\mathcal{N}}(y) = -\log \sigma(-y)$. We note that in node2vec, $\mathcal{P}$ is formed by taking concurrent edges appearing in a random walk, and in LINE, $\mathcal{P}$ is formed by uniform edge sampling; in both methods $\mathcal{N}$ is taken to be the same negative sampling distribution $N(u|v) \propto \deg(v)^{3/4}$ [42]. The scheme in (2) attempts to minimize the function

$$\mathcal{R} := \sum_{i,j} \left\{ \mathbb{P}\big((i,j) \in \mathcal{P}\big) \ell_P(\langle \omega_i, \omega_j \rangle) + \mathbb{P}\big((i,j) \in \mathcal{N}\big) \ell_{\mathcal{N}}(\langle \omega_i, \omega_j \rangle) \right\} \tag{3}$$

obtained by averaging over the random process used to create $\mathcal{P}$ and $\mathcal{N}$ at each iteration of stochastic gradient descent [51, 59]; see Appendix A.1 for a detailed derivation.

In the case where $\ell_{\mathcal{P}}(y) = -\log \sigma(y)$ and $\ell_{\mathcal{N}}(y) = -\log \sigma(-y)$, with $\mathcal{P}$ and $\mathcal{N}$ being random subsets of $\mathcal{E}$ and $(\mathcal{V} \times \mathcal{V}) \setminus \mathcal{E}$ (such as in LINE, or node2vec with a window length of 1), then we can view (3) as the function obtained via trying to minimize the negative log-likelihood (equivalent to maximizing the log-likelihood) of a probabilistic model

$$a_{uv} \,|\, \omega_u, \omega_v \sim \text{Bernoulli}(\sigma(\langle \omega_u, \omega_v \rangle)) \text{ independently for } u < v, \tag{4}$$

using the stochastic gradient descent scheme specified in (2) (see Appendix A.2). If it is assumed that the embedding vectors are drawn i.i.d from some latent distribution, then this corresponds to implicitly fitting an exchangeable model to the graph [40], where the distribution of the adjacency matrices are invariant to joint permutations of the vertices of the network, or equivalently (by the Aldous-Hoover theorem [1]) arise from a probabilistic model

$$\lambda_u \sim \text{Unif}([0,1]) \text{ independently}, \qquad a_{uv} \,|\, \lambda_u, \lambda_v \sim \text{Bernoulli}(W(\lambda_u, \lambda_v)) \text{ independently}, \tag{5}$$

for some symmetric measurable function $W : [0,1]^2 \to [0,1]$ called a *graphon*. We highlight that the model in (4) is implicitly fitting a dense graph model to the data, *even when the observed graph data are sparse*, as is the case with real world networks. One way of partially addressing this issue is to consider a *sparsified exchangeable graph* with $W \to \rho_n W$ for some sparsifying sequence $\rho_n \to 0$.

A natural example of a prior distribution on embedding vectors is $\omega_i \sim \text{Normal}(0, (2\xi)^{-1/2} I_d)$ for some constant $\xi > 0$, so that the contribution to the negative log-likelihood of each embedding is $\xi \|\omega_i\|_2^2$. The full negative log-likelihood of (4) with such a prior distribution is then given by

$$-\sum_{i,j} \{ a_{ij} \log \sigma(\langle \omega_i, \omega_j \rangle) + (1 - a_{ij}) \log \sigma(-\langle \omega_i, \omega_j \rangle) \} + \xi \sum_i \|\omega_i\|_2^2, \tag{6}$$

which depends only on the matrix $G_{ij} = \langle \omega_i, \omega_j \rangle$; this loss can also arise by considering a weight decay optimization scheme [see e.g 34]. In either case, letting $\Omega \in \mathbb{R}^{n \times d}$ be the matrix whose rows are the ω_i, so $G = \Omega\Omega^T$, we get that

$$\sum_{i=1}^n \|\omega_i\|_2^2 = \sum_{i=1}^n \sum_{j=1}^d \omega_{ij}^2 = \sum_{i=1}^n \sum_{j=1}^d \omega_{ij}\omega_{ji} = \sum_{i=1}^n (\Omega\Omega^T)_{ii} = \text{tr}(\Omega\Omega^T) = \text{tr}(G) = \|G\|_* \tag{7}$$

where $\|G\|_*$ is the nuclear norm, which equals the sum of the singular values of G. As a result, the prior distribution on the ω_i induces a nuclear-norm penalty on the matrix G, which is well known to shrink the singular values of G exactly towards zero, and to make G low rank [see e.g 4, 18, 33, 50].

In this paper, our interest is in studying the effects of such a regularizer in the scenario where embeddings are learned via subsampling, in which case the corresponding version of (3) becomes

$$\sum_{i,j} \left\{ \mathbb{P}\big((i,j) \in \mathcal{P}\big) \ell_P(\langle \omega_i, \omega_j \rangle) + \mathbb{P}\big((i,j) \in \mathcal{N}\big) \ell_{\mathcal{N}}(\langle \omega_i, \omega_j \rangle) \right\} + \sum_i \mathbb{P}\big(i \in \mathcal{V}(\mathcal{P} \cup \mathcal{N})\big) \|\omega_i\|_2^2, \tag{8}$$

where $\mathcal{V}(\mathcal{P} \cup \mathcal{N})$ is the set of vertices which appear either in $\mathcal{P}$ or $\mathcal{N}$ (see Appendix A.3). In particular, we study minimizers $(\widehat{\omega}_1, \ldots, \widehat{\omega}_n)$ of (8) where the graph is generated from a sparsified

exchangeable graph, by associating (8) with a population version of the objective, which allows us to give guarantees of the form

$$\frac{1}{n^2}\sum_{i,j}\left(\langle\widehat{\omega}_i,\widehat{\omega}_j\rangle - K(\lambda_i,\lambda_j)\right)^2 = o_p(1) \;\text{ and }\; \min_{Q\in O(d)}\frac{1}{n}\sum_i\left\|\widehat{\omega}_i - \psi(\lambda_i)Q\right\|_2^2 = o_p(1) \quad (9)$$

for some functions $K:[0,1]^2\to\mathbb{R}$ and $\psi:[0,1]\to\mathbb{R}^d$ obtained through the process of minimizing the population objective (Section 3). We highlight that our theoretical results apply *in the regime* $d \ll n$, as what occurs with chosen embedding dimensions in practice. This means that (8) is non-convex even in the matrix $G_{ij} = \langle\omega_i,\omega_j\rangle$ as a consequence of rank constraints, which complicates the theoretical analysis. Our results will recover the motivation above that the regularization acts to effectively reduce the rank of the learned embedding vectors. We also illustrate experimentally that using such regularization, when chosen optimally, can give performance competitive to architectures such as GraphSAGE (Section 4).

1.1 Related works

Guarantees for embedding methods: We highlight that there is an extensive literature on the embeddings formed by the eigenvectors of the adjacency or Laplacian matrices of a network [e.g 3, 36–38, 52, 55]. Under various latent space models for the network, these works give guarantees on quantities of the form $\max_i\|\widehat{\omega}_i - Q\psi_i\|_2^2$ for some orthogonal matrix Q and vectors ψ_i, to discuss recovery of latent variables and/or obtain exact recovery in a community detection task. Stronger bounds are obtained in this setting as they are able to directly apply matrix eigenvector perturbation methods to study the embeddings, which we cannot with our approach; as a tradeoff, our approach allows us to study embeddings learned via a variety of subsampling schemes, which these works do not. We highlight that the second bound in (9) can still be used to give guarantees for weak recovery of community detection [37]. There are a few works discussing random walk methods like node2vec, albeit circumventing the non-convexity in the problem; [46] discusses the unconstrained minima of the loss (1) when $d = n$, with [64] then examining the best rank r approximation to this matrix when the generating model is a stochastic block model with k communities and $r \le k$. [20] covers the non-convex regime $d \ll n$ in the case where $\xi_n = 0$, i.e without regularization.

Nuclear norm penalties and ℓ_2 regularized embeddings: In the context of matrix factorization (so the matrix factors are embedding matrices), the effects of Frobenius norm penalties inducing nuclear norm penalties are well known [e.g 49, 57]; generally, there is an extensive literature on the effects of nuclear norm penalization in the finite-dimensional setting [e.g 4, 18, 33, 50]. In [63], it is shown that ℓ_2 regularized node2vec gives an improvement in performance on downstream tasks, mirroring our experiments.

Graphon estimation: We mention that our guarantees on the gram matrix formed by the learned embeddings are similar to those obtained in the graphon estimation literature [see e.g 8, 9, 17, 21, 31, 35, 61, 62]. Depending on the choice of sampling scheme and loss, it is possible for the limiting matrix $K(\lambda_i,\lambda_j)$ to be an invertible transformation of $W(\lambda_i,\lambda_j)$, and so we compare our rates of convergence in such a scenario (see Remark 6 in Appendix C).

2 Framework and assumptions of analysis

Given a sequence of graphs $\mathcal{G}_n = (\mathcal{V}_n,\mathcal{E}_n)$, and writing $\boldsymbol{\omega}_n = (\omega_1,\dots,\omega_n)$ with $\omega_u\in\mathbb{R}^d$ denoting the d-dimensional embedding of vertex u, we study the regularized empirical risk function

$$\begin{aligned}\mathcal{R}_n(\boldsymbol{\omega}_n) + \xi_n\mathcal{R}_n^{\text{reg}}(\boldsymbol{\omega}_n) \;\text{ where }\; \mathcal{R}_n(\boldsymbol{\omega}_n) &:= \sum_{\substack{i,j\in[n]\\ i\neq j}}\mathbb{P}\big((i,j)\in S(\mathcal{G}_n)\,|\,\mathcal{G}_n\big)\ell(\langle\omega_i,\omega_j\rangle,a_{ij}),\\ \mathcal{R}_n^{\text{reg}}(\boldsymbol{\omega}_n) &:= \sum_{i\in[n]}\mathbb{P}\big(i\in\mathcal{V}(S(\mathcal{G}_n))\,|\,\mathcal{G}_n\big)\|\omega_i\|_2^2,\end{aligned} \quad (10)$$

for $\xi_n \ge 0$. Here, we define a subsample $S(\mathcal{G})$ of a graph $\mathcal{G}$ as a collection of vertices $\mathcal{V}(S(\mathcal{G}))$ and a symmetric subset of $\mathcal{V}(S(\mathcal{G}))\times\mathcal{V}(S(\mathcal{G}))$. The sampling probabilities are conditional on $\mathcal{G}_n$ as we will soon assume that the $\mathcal{G}_n$ also arise from a probabilistic model. We note (10) arises from (8) whenever $\mathcal{P}$ and $\mathcal{N}$ are random subsets of $\mathcal{E}_n$ and $\mathcal{V}\times\mathcal{V}\setminus\mathcal{E}_n$ respectively (see Appendix A.4).

Throughout, $\ell(y,x)$ is either the cross-entropy loss $\ell_\sigma(y,x) := -x\log\sigma(y) - (1-x)\log\sigma(-y)$ or the squared loss $\ell_2(y,x) := (y-x)^2$. We now discuss our assumptions on the generative model of the graph, which are motivated by the discussion in the introduction that embedding methods are implicitly fitting an exchangeable graph model.

Assumption 1. *We assume that the sequence of graphs $(\mathcal{G}_n)_{n\geq 1}$ have vertex sets $\mathcal{V}_n = [n]$ and arise from a graphon process with generating graphon $W_n = \rho_n W$ with $\rho_n \gg \log(n)/n$, so that*

$$\lambda_i \overset{i.i.d}{\sim} \mathrm{Unif}[0,1] \textit{ for } i \in [n], \quad a_{ij}|\lambda_i,\lambda_j \overset{indep.}{\sim} \mathrm{Bernoulli}(\rho_n W(\lambda_i,\lambda_j)) \textit{ for } i<j.$$

We moreover suppose that $W \in [c, 1-c]$ for some $c > 0$, and either a) W is piecewise constant on a partition $\mathcal{Q}\times\mathcal{Q}$ where $\mathcal{Q}$ is a partition of $[0,1]$ of size κ (so the model is a stochastic block model), or b) W is Hölder$([0,1]^2, \beta_W, L_W)$ for some exponents and constants $\beta_W \in (0,1]$ and $L_W < \infty$.

Remark 1. *Here we use the canonical choice of uniform latent variables for the graphon as guaranteed by the Aldous-Hoover theorem for vertex exchangeable graphs [1]; in principle, our results can extend to graphons on higher dimensional latent spaces by the same style of arguments [see e.g 20]. We highlight that our assumptions are somewhat restrictive with regards to the boundedness and and sparsity conditions; while it is common to allow $\rho_n \gg \log(n)/n$ [e.g 43, 61], it is also common to work in the regime where $\rho_n = \Theta(\log(n)/n)$ or smaller [e.g 9, 62]. We highlight that in general graphons, regardless of the sparsity factor, tend to not give rise to graphs with power-law or heavy-tailed type degree structures, which is frequent with real world networks [15, 65].*

For subsampling schemes used in practice (such as random walk and uniform edge samplers), the sampling probability of vertices and edges depends only on local features of the graph. Following [20], we can formalize this as follows:

Assumption 2. *There exist sequences of measurable functions $(f_n)_{n\geq 1}$ and $(\tilde{g}_n)_{n\geq 1}$ and a sequence $s_n = o(1)$, such that*

$$\max_{\substack{i,j\in[n]\\ i\neq j}} \Big|\frac{n^2\mathbb{P}\big((i,j)\in S(\mathcal{G}_n)\,|\,\mathcal{G}_n\big)}{f_n(\lambda_i,\lambda_j,a_{ij})} - 1\Big| = O_p(s_n), \ \max_{i\in[n]}\Big|\frac{n\mathbb{P}\big(i\in\mathcal{V}(S(\mathcal{G}_n))\,|\,\mathcal{G}_n\big)}{\tilde{g}_n(\lambda_i)} - 1\Big| = O_p(s_n),$$

and moreover $\mathbb{E}[f_n(\lambda_1,\lambda_2,a_{12})] = O(1)$, $\mathbb{E}[f_n(\lambda_1,\lambda_2,a_{12})^2] = O(\rho_n^{-1})$, $\mathbb{E}[\tilde{g}_n(\lambda_1)] = O_p(1)$.

Examples of sampling schemes satisfying this condition are given in Section 3.2. We additionally impose some regularity conditions on the "averaged" versions of the above functions defined by

$$\tilde{f}_n(l,l',1) := f_n(l,l',1)W_n(l,l'), \qquad \tilde{f}_n(l,l',0) := f_n(l,l',0)(1-W_n(l,l')). \tag{11}$$

Assumption 3. *We assume the the functions $\tilde{f}_n(l,l',1)$, $\tilde{f}_n(l,l',0)$ and $\tilde{g}_n(l)$ are uniformly bounded above by M and away from zero by M^{-1} for some constant $M \in (0,\infty)$. We also suppose that either a) there exists a partition $\mathcal{Q}$ of $[0,1]$ into κ parts such that $\tilde{f}_n(l,l',1)$ and $\tilde{f}_n(l,l',0)$ are piecewise constant on $\mathcal{Q}\times\mathcal{Q}$, and $\tilde{g}_n(l)$ is piecewise constant on $\mathcal{Q}$; or b) the $\tilde{f}_n(l,l',1)$ and $\tilde{f}_n(l,l',0)$ are Hölder$([0,1]^2, \beta, L)$, and that the $\tilde{g}_n(l)$ are Hölder$([0,1], \beta, L)$.*

We note that Assumption 3 will follow as a direct consequence of Assumptions 1 and 2 for the sampling schemes discussed in Section 3.2; in particular, β will depend on β_W and the hyper-parameters of the sampling scheme used.

3 Theoretical results

Our theoretical results take the following flavour: we identify the correct population versions of $\mathcal{R}_n(\boldsymbol{\omega}_n)$ and $\mathcal{R}_n^{\mathrm{reg}}(\boldsymbol{\omega}_n)$ in the large sample limit $n\to\infty$ to give a regularized population risk, and then use this to give guarantees about any minimizers of $\mathcal{R}_n(\boldsymbol{\omega}_n) + \xi_n\mathcal{R}_n^{\mathrm{reg}}(\boldsymbol{\omega}_n)$ being close (in some sense) to the unique minimizer of the regularized population risk. As network embedding methods are used on very large networks, such a large sample statistical analysis is appropriate.

We introduce the population versions of $\mathcal{R}_n(\boldsymbol{\omega}_n)$ and $\mathcal{R}_n^{\mathrm{reg}}(\boldsymbol{\omega}_n)$ respectively as

$$\mathcal{I}_n[K] := \int_{[0,1]^2}\sum_{x\in\{0,1\}} \tilde{f}_n(l,l',x)\ell(K(l,l'),x)\,dl\,dl', \qquad \mathcal{I}_n^{\mathrm{reg}}[K] := \int_0^1 K(l,l)\tilde{g}_n(l)\,dl, \tag{12}$$

defined over functions $K : [0,1]^2 \to \mathbb{R}$. $\mathcal{I}_n[K]$ was first introduced in [20]. The formula given for $\mathcal{I}_n^{\text{reg}}[K]$ holds only for K whose diagonal is well defined; in general, if K admits a decomposition

$$K(l,l') = \sum_{i=1}^{\infty} \mu_i(K)\psi_i(l)\psi_i(l') \quad \text{where} \quad \int_0^1 \psi_i(l)\psi_j(l)\tilde{g}_n(l)\,dl = \begin{cases} 1 & \text{if } i=j, \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

and $\mu_i(K) \geq 0$ for all i (understood as a limit in $L^2([0,1]^2)$), then we can extend the definition of the regularizer to be $\mathcal{I}_n^{\text{reg}}[K] := \sum_{i=1}^{\infty} \mu_i(K)$. Consequently, the penalty corresponds to the trace of K, when viewed as the kernel of a Hilbert-Schmidt operator. This means that $\mathcal{I}_n^{\text{reg}}[K]$ should be viewed as a nuclear-norm penalty on the kernel K, which encourages the $\mu_i(K)$ to be shrunk exactly towards zero, similar to the finite dimensional scenario; also see e.g Theorem 5.

3.1 Guarantees on the learned embedding vectors

We begin with a result which guarantees that $\mathcal{I}_n[K] + \xi_n \mathcal{I}_n^{\text{reg}}[K]$, once restricted to an appropriate domain, is the correct population version of $\mathcal{R}_n(\boldsymbol{\omega}_n) + \xi_n \mathcal{R}_n^{\text{reg}}(\boldsymbol{\omega}_n)$.

Theorem 1. *Suppose that Assumptions 2 and 3 hold, and that $\xi_n = O(1)$. Define*

$$\mathcal{Z}_d^{\geq 0}(A) := \left\{ K \,:\, K(l,l') = \langle \eta(l), \eta(l') \rangle, \eta : [0,1] \to [-A,A]^d \right\} \text{ for } d \in \mathbb{N}, A > 0.$$

Then we have that

$$\Big| \min_{\boldsymbol{\omega}_n \in ([-A,A]^d)^n} \left\{ \mathcal{R}_n(\boldsymbol{\omega}_n) + \xi_n \mathcal{R}_n^{reg}(\boldsymbol{\omega}_n) \right\} - \min_{K \in \mathcal{Z}_d^{\geq 0}(A)} \left\{ \mathcal{I}_n[K] + \xi_n \mathcal{I}_n^{reg}[K] \right\} \Big| = O_p(r_n)$$

where $r_n = s_n + (d^p/n\rho_n)^{1/2} + t_n$, with $t_n = (\log \kappa / n)^{1/2}$ under Assumption 3a), $t_n = (\log(n)/n^{2\beta/(1+2\beta)})^{1/2}$ under Assumption 3b), $p = 3$ for the cross-entropy loss and $p = 5$ for the squared loss.

See Appendix B for the proof and a discussion of the rates give; we note that it is necessary that $d \ll n$ in order for $r_n \to 0$. The rates can be improved to give $p = 1$, under additional restrictions on the parameter space (Remark 3). The interpretation of the set $\mathcal{Z}_d^{\geq 0}(A)$ is as follows: if $\omega_u \in [-A,A]^d$ is the embedding of vertex u, in the population limit $\eta(\lambda_u) \in [-A,A]^d$ should give the embedding of a vertex with latent feature λ_u. As (10) is parameterized through terms of the form $\langle \omega_u, \omega_v \rangle$, the population version of (10) should be parameterized through functions $K(\lambda_u, \lambda_v) = \langle \eta(\lambda_u), \eta(\lambda_v) \rangle$.

Remark 2. *We note that the assumption that the embedding vectors belong to a hypercube $[-A,A]^d$ is not restrictive; for example, in practice embedding vectors are usually initialized randomly and uniformly over $[-1,1]^d$. Moreover, our results allow for A to grow logarithmically with n, and only change the bound by a poly-log factor. We highlight that if $\xi_n \to \infty$ as $n \to \infty$, then the embedding vectors will shrink towards 0 as $n \to \infty$ (as seen in Figure 1). This is because the proof of Theorem 1 shows that any minimizer must satisfy $n^{-1}\sum_{i=1}^n \|\omega_i\|_2^2 = O_p(\xi_n^{-1}) = o_p(1)$ in such a regime.*

We now give convergence guarantees for any sequence of embedding vectors minimizing (10).

Theorem 2. *Suppose that Assumptions 2 and 3 hold and that $\xi_n = O(1)$. Then for each n, there exists a unique minimizer to the optimization problem*

$$K_n^* = \operatorname*{arg\,min}_{K \in \mathcal{Z}^{\geq 0}} \left\{ \mathcal{I}_n[K] + \xi_n \mathcal{I}_n^{reg}[K] \right\} \quad \textit{where} \quad \mathcal{Z}^{\geq 0} := \text{cl}\Big(\bigcup_{d \geq 1} \mathcal{Z}_d^{\geq 0}(A) \Big)$$

is free of $A > 0$ (see Proposition 1 for details). Moreover, under some regularity conditions on the K_n^* (see Theorem 6 in Appendix C), there exists $A' < \infty$ free of n and a sequence of embedding dimensions $d = d(n) \ll n$ such that, for any sequence of minimizers

$$\widehat{\boldsymbol{\omega}}_n \in \operatorname*{arg\,min}_{\boldsymbol{\omega}_n \in ([-A_1,A_1]^d)^n} \left\{ \mathcal{R}_n(\boldsymbol{\omega}_n) + \xi_n \mathcal{R}_n^{reg}(\boldsymbol{\omega}_n) \right\} \textit{ satisfying } \max_{i,j} |\langle \widehat{\omega}_i, \widehat{\omega}_j \rangle| \leq A_2$$

with $A_1, A_2 \geq A'$, we have that

$$\frac{1}{n^2} \sum_{i,j \in [n]} \left(\langle \widehat{\omega}_i, \widehat{\omega}_j \rangle - K_n^*(\lambda_i, \lambda_j) \right)^2 = o_p(1).$$

Moreover, when Assumption 3a) holds, K_n^* can be computed via a finite dimensional convex program, and is of rank $r \leq \kappa$, in that an expansion of the form (13) holds with $\mu_i(K_n^*) = 0$ for $i > r$.

The case where $\xi_n = 0$ is proven in [20], which also verifies the convergence on simulated data. A proof is given in Appendix C. Under certain circumstances, this allows us to give guarantees about the distribution of the embedding vectors themselves.

Theorem 3. *Suppose that K_n^* is regular in the sense of Theorem 2, with the conclusions of the theorem holding. Moreover suppose that K_n^* is a kernel of rank $r < \infty$, has a decomposition of the form $K_n^*(l, l') = \sum_{i=1}^r \phi_{n,i}(l)\phi_{n,i}(l')$ for some functions $\phi_{n,i} : [0, 1] \to \mathbb{R}$, and the dimension d of the embedding vectors is chosen to be equal to r. Writing $\phi_n(l) = (\phi_{n,i}(l))_{i=1}^r$, we have*

$$\min_{Q \in O(r)} \frac{1}{n} \sum_{i=1}^n \left\| \widehat{\omega}_i - Q\phi_n(\lambda_i) \right\|_2^2 = o_p(1). \tag{14}$$

The assumption that $d = r$ in Theorem 3 is a restrictive one, given that embedding dimensions in practice are usually chosen to be one of either 128, 256 or 512. This can be alleviated by instead giving a guarantee for the optimal r dimensional projection of the embedding vectors.

Theorem 4. *If instead $d > r$ in Theorem 3, let $\widetilde{G} \in \mathbb{R}^{n \times n}$ denote the best rank r approximation to the matrix $G_{ij} := (\langle \widehat{\omega}_i, \widehat{\omega}_j \rangle)_{ij}$, and write $\widetilde{G} = \tilde{\Omega}\tilde{\Omega}^T$ for some $\tilde{\Omega} \in \mathbb{R}^{n \times r}$. Then $n^{-2}\|\widetilde{G} - G\|_F^2 = o_p(1)$, and writing $\widetilde{\omega}_i$ for the rows of $\widetilde{\Omega}$, the guarantee in* (14) *holds with $\widetilde{\omega}_i$ replacing the $\widehat{\omega}_i$.*

Informally, this says that the embedding vectors approximately lie on a r-dimensional subspace which contains some latent information about the network, depending on the minimizing kernel K_n^*. We now highlight that the assumption that K_n^* is of finite rank $r < \infty$ is not restrictive even when W is not a SBM; this is a consequence of the effect of the regularization penalty $\mathcal{I}_n^{\text{reg}}[K]$.

Theorem 5. *Let $\ell(y, x)$ be the squared loss, and suppose that $\rho_n = 1$, $\tilde{f}_n(l, l', 1) = \tilde{f}_n(l, l', 0) = c_1$ and $\tilde{g}_n(l) = c_2$ for some constants $c_1, c_2 > 0$ (see e.g Algorithm 1 in Section 3.2). Then if W is Hölder$([0, 1]^2, \beta, L)$, the minima of $\mathcal{I}_n[K] + \xi_n \mathcal{I}_n^{reg}[K]$ over $\mathcal{Z}^{\geq 0}$ is of finite rank for any $\xi_n > 0$, and is also Hölder continuous of exponent β.*

We highlight that this result also shows that the regularizer acts to shrink the singular values of a minimizer of $\mathcal{I}_n[K] + \xi_n \mathcal{I}_n^{\text{reg}}[K]$ exactly towards zero. See Appendix D for proofs of Theorems 3, 4 and 5.

3.2 Sampling schemes satisfying Assumption 2

We now discuss some examples of frequently used sampling schemes which satisfy Assumption 2. Proofs of the results in this section can be found in Appendix E. We introduce the notation

$$W(\lambda, \cdot) := \int_0^1 W(\lambda, y)\, dy, \quad \mathcal{E}_W(\alpha) := \int_0^1 W(\lambda, \cdot)^\alpha\, d\lambda, \quad \mathcal{E}_W := \mathcal{E}_W(1). \tag{15}$$

Algorithm 1 (Uniform vertex sampling)**.** *Given a graph $\mathcal{G}_n$ and number of samples k, we select k vertices from $\mathcal{G}_n$ uniformly and without replacement, and then return $S(\mathcal{G}_n)$ as the induced subgraph using these sampled vertices.*

Lemma 1. *For Algorithm 1, Assumption 2 holds with $f_n(\lambda_i, \lambda_j, a_{ij}) = k(k-1)$, $\tilde{g}_n(\lambda_i) = k$ and $s_n = 1/n$.*

Algorithm 2 (Uniform edge sampling [54])**.** *Given a graph $\mathcal{G}_n$, number of edges to sample k, and number of negative samples l per positive sample,*

i) We form a sample $S_0(\mathcal{G}_n)$ by sampling k edges from $\mathcal{G}_n$ uniformly and without replacement;

ii) We form a sample set of negative samples $S_{ns}(\mathcal{G}_n)$ by drawing, for each $u \in \mathcal{V}(S_0(\mathcal{G}_n))$, l vertices $v_1, \ldots, v_l$ i.i.d according to the unigram distribution

$$\mathrm{Ug}_\alpha\big(v \,|\, \mathcal{G}_n\big) \propto \mathbb{P}\big(v \in \mathcal{V}(S_0(\mathcal{G}_n)) \,|\, \mathcal{G}_n\big)^\alpha$$

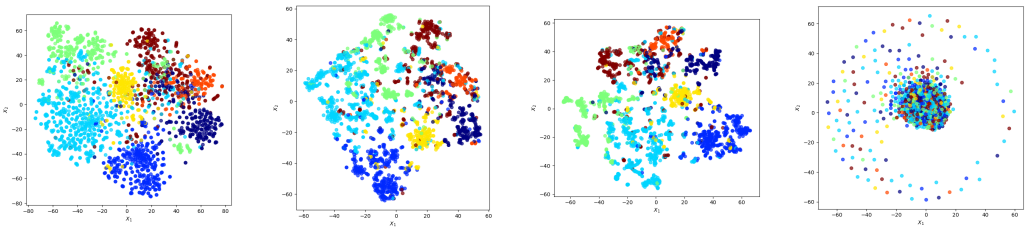
and then adjoining $(u, v_i) \to S_{ns}(\mathcal{G}_n)$ if $a_{uv_i} = 0$.

We then return $S(\mathcal{G}_n)$ as the union of $S_0(\mathcal{G}_n)$ and $S_{ns}(\mathcal{G}_n)$.

Lemma 2. *For Algorithm 2, Assumption 2 holds with $s_n = (\log(n)/n\rho_n)^{-1/2}$,*

$$f_n(\lambda_i, \lambda_j, 1) = \frac{2k}{\mathcal{E}_W \rho_n}, \; f_n(\lambda_i, \lambda_j, 0) = \frac{2kl}{\mathcal{E}_W \mathcal{E}_W(\alpha)} \big\{ W(\lambda_i, \cdot)W(\lambda_j, \cdot)^\alpha + W(\lambda_i, \cdot)^\alpha W(\lambda_j, \cdot) \big\},$$

$$\tilde{g}_n(\lambda_i) = \frac{2kW(\lambda_i, \cdot)}{\mathcal{E}_W} + \frac{2klW(\lambda_i, \cdot)^\alpha}{\mathcal{E}_W \mathcal{E}_W(\alpha)} \cdot \int_0^1 (1 - \rho_n W(\lambda_i, y))W(y, \cdot)\, dy.$$



(a) $\xi = 0$ (b) $\xi = 5 \times 10^{-7}$ (c) $\xi = 10^{-6}$ (d) $\xi = 10^{-3}$

Figure 1: TSNE visualizations of Cora network embeddings, learnt using node2vec for different regularization penalties ξ, with different colors representing different classes. As ξ increases, the classes begin to cluster together and separate, and then eventually collapse towards the origin.

Algorithm 3 (Random walk sampling [24, 44])**.** *Given a graph $\mathcal{G}_n$, a walk length k, number of negative samples l per positively sampled vertex and unigram parameter α, we*

i) Perform a simple random walk on $\mathcal{G}_n$ of length k, beginning from its stationary distribution, to form a path $(\tilde{v}_i)_{i \leq k+1}$, and report $(\tilde{v}_i, \tilde{v}_{i+1})$ for $i \leq k$ as part of $S_0(\mathcal{G}_n)$;

ii) For each vertex $\tilde{v}_i$, we select l vertices $(\eta_j)_{j \leq l}$ independently and identically according to the unigram distribution

$$\mathrm{Ug}_\alpha(v \,|\, \mathcal{G}_n) \propto \mathbb{P}\big(\tilde{v}_i = v \text{ for some } i \leq k \,|\, \mathcal{G}_n\big)^\alpha$$

and then form $S_{ns}(\mathcal{G}_n)$ as the collection of vertex pairs $(\tilde{v}_i, \eta_j)$ which are not an edge in $\mathcal{G}_n$.

We then return $S(\mathcal{G}_n)$ as the union of $S_0(\mathcal{G}_n)$ and $S_{ns}(\mathcal{G}_n)$.

Lemma 3. *For Algorithm 3, Assumption 2 holds with $s_n = (\log(n)/n\rho_n)^{1/2}$,*

$$f_n(\lambda_i, \lambda_j, 1) = \frac{2k}{\mathcal{E}_W \rho_n}, \; f_n(\lambda_i, \lambda_j, 0) = \frac{l(k+1)}{\mathcal{E}_W \mathcal{E}_W(\alpha)} \big\{ W(\lambda_i, \cdot) W(\lambda_j, \cdot)^\alpha + W(\lambda_i, \cdot)^\alpha W(\lambda_j, \cdot) \big\},$$

$$\tilde{g}_n(\lambda_i) = \frac{kW(\lambda_i, \cdot)}{\mathcal{E}_W} + \frac{(k+1)lW(\lambda_i, \cdot)^\alpha}{\mathcal{E}_W(\alpha)\mathcal{E}_W} \cdot \int_0^1 (1 - \rho_n W(\lambda_i, y)) W(y, \cdot)\, dy.$$

Examining the formula for $\tilde{g}_n(\lambda)$ above, we see that for random walk samplers the shrinkage provided to the learned kernel will be greater for vertices with larger degrees. Indeed, as e.g $K(\lambda, \lambda) = \|\eta(\lambda)\|_2^2$ for $K \in \mathcal{Z}_d^{\geq 0}(A)$, the larger $\tilde{g}_n(\lambda)$ is, $\|\eta(\lambda)\|_2^2$ will be forced closer towards zero.

4 Experiments

We now examine the performance in using regularized node2vec embeddings for link prediction and node classification tasks, and illustrate comparable, when not superior, performance to more complicated encoders for network embeddings. We perform experiments on the Cora, CiteSeer and PubMedDiabetes citation network datasets (see Appendix F for more details).

4.1 Methods used for comparison

For our experiments, we consider 128 dimensional node2vec embeddings learned with either no regularization, or a ℓ_2 penalty on the embedding vectors with weight $\xi \in \{1, 5\} \times 10^{-\{3,4,5,6,7,8\}}$; and with or without the node covariates concatenated. We compare these against methods which incorporate the network and covariate structure together in a non-linear fashion. We consider three methods - a two layer GCN [30] with 256 dimensional output embeddings trained in an unsupervised fashion through the node2vec loss, a two layer GraphSAGE architecture [25] with 256 dimensional output embeddings trained through the node2vec loss, and a single layer 256 dimensional GCN trained using Deep Graph Infomax (DGI) [60]. We highlight that the GCN, GraphSAGE and DGI always have access to the nodal covariates during training. All of our experiments used the Stellargraph [19] implementations for each method. Further experimental details are given in Appendix F.

Table 1: PR AUC scores for link prediction experiments, and macro F1 scores for node classification experiments. "n2v" stands for node2vec without regularization; "rn2v" stands for regularized node2vec with the best score over the specified grid of penalty values; "NF" indicates that node features were concatenated to the learned node embeddings.

Methods	PR AUC (link prediction)			Macro F1 (node classification)		
	Cora	CiteSeer	PubMed	Cora	CiteSeer	PubMed
n2v	0.84 ± 0.01	0.80 ± 0.02	0.86 ± 0.01	0.67 ± 0.04	0.48 ± 0.03	0.76 ± 0.00
rn2v	0.90 ± 0.01	0.88 ± 0.02	0.91 ± 0.00	0.73 ± 0.03	0.55 ± 0.04	0.77 ± 0.00
n2v+NF	0.88 ± 0.01	0.90 ± 0.01	0.92 ± 0.00	0.70 ± 0.03	0.54 ± 0.03	0.79 ± 0.01
rn2v+NF	0.92 ± 0.01	0.93 ± 0.01	0.95 ± 0.00	0.74 ± 0.03	0.58 ± 0.04	0.84 ± 0.00
GCN	0.90 ± 0.01	0.87 ± 0.02	0.94 ± 0.00	0.67 ± 0.04	0.48 ± 0.04	0.80 ± 0.01
GSAGE	0.90 ± 0.01	0.90 ± 0.01	0.88 ± 0.01	0.74 ± 0.04	0.56 ± 0.03	0.79 ± 0.00
DGI	0.91 ± 0.01	0.93 ± 0.01	0.95 ± 0.00	0.76 ± 0.03	0.60 ± 0.02	0.84 ± 0.00

4.2 Link prediction experiments

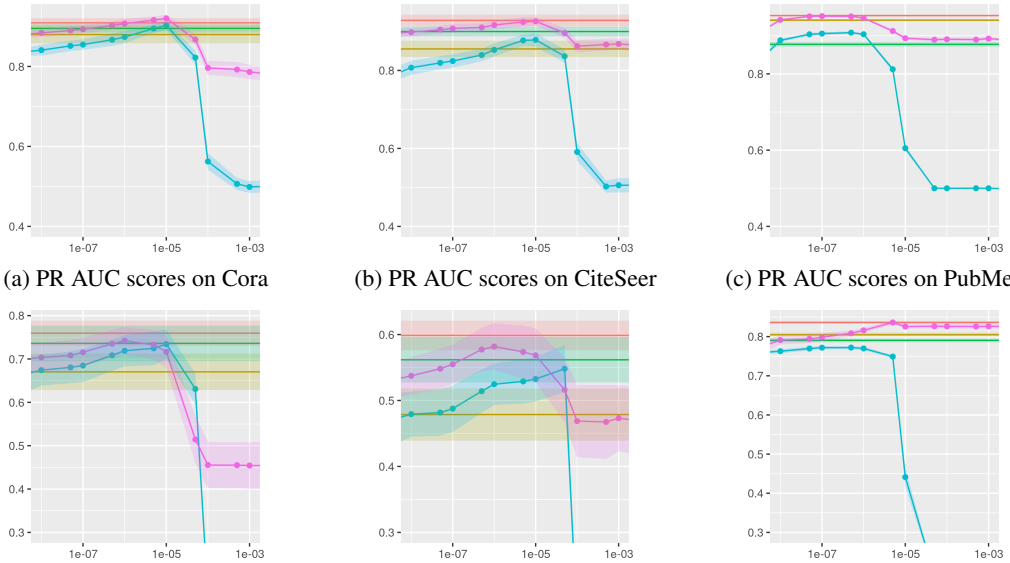
For the link prediction experiments, we create a training graph by removing 10% of both the edges and non-edges within the network, and use this to learn an embedding of the network. We then form link embeddings by taking the entry-wise product of the corresponding node embeddings, use 10% of the held-out edges to build a logistic classifier for the link categories, and then evaluate the performance on the remaining edges, repeating this process 50 times.

The PR AUC scores are given in Table 1 and visualized in Figure 2. Provided the regularization parameter is chosen optimally, we see an improvement in performance compared to using no regularization, with the jump in performance slightly greater when nodal covariates are not incorporated into the embedding. The optimally regularized version of node2vec, even without including the node covariates, is competitive with the GCN and GraphSAGE (being equal or outperforming at least one of them across the three datasets), and that the version with concatenated node features is as good as the GCN trained using DGI. For all the datasets, we observe a sharp decrease in performance after the optimal weight, suggesting that it needs to be chosen carefully to avoid removing the informative network structure. As seen in Figure 1, this occurs as the learned embeddings become randomly distributed at the origin once the regularization weight is too large.

4.3 Node classification experiments

To evaluate performance for the node classification task, we learn a network embedding without access to the node labels, and then learn/evaluate a one-versus-rest multinomial node classifier using 5%/95% stratified training/test splits of the node labels. We repeat this over 25 training runs of the embeddings, and a further 25 training/test splits for the node classifiers per embedding. Table 1 and Figure 2 show the average macro F1 scores and their standard deviation for each method and dataset. Similar to the link prediction experiments, we see that the optimally regularized node2vec methods are competitive, if not outperforming, the GCN and GraphSAGE trained through the node2vec loss, and is outperformed by the GCN learned using DGI by no more than two percentage points. In these experiments, the standard deviations correspond partially to the randomness induced by the training/test splits of the node labels, and suggest that the regularized version of node2vec is no less robust to particular choices of training/test splits than the other methods.

Interestingly, we note that the optimal performance on PubMed is given by the regularized node2vec with node features. However, as the highest level of performance is performed when the regularization weight is so large that the learnt embeddings are uninformative (as illustrated by the massive decrease in performance of the regularized method without node features), this suggests that the nodes can be classified using only the covariate information, and that the network features are not needed. As the dataset only has three distinct classes corresponding to the academic field, and the node features are TF-IDF embeddings of the academic papers, this is not too surprising.



(a) PR AUC scores on Cora (b) PR AUC scores on CiteSeer (c) PR AUC scores on PubMed

(d) Macro F1 scores on Cora (e) Macro F1 scores on CiteSeer (f) Macro F1 scores on PubMed

Figure 2: Plots of average PR AUC scores for link prediction and macro F1 scores for node classification (bands representing plus/minus one standard deviation), against the regularization penalty used for node2vec with (purple) and without (blue) concatenated node features. The results from GCN, GraphSAGE and DGI are given as horizontal bars in yellow, green and red respectively.

5 Conclusion

In this paper we have theoretically described the effects of performing ℓ_2 regularization on network embedding vectors learned by schemes such as node2vec, describing the asymptotic distribution of the embedding vectors learned through such schemes, and showed that the regularization helps to reduce the effective dimensionality of the learned embeddings by penalizing the singular values of the limiting distribution of the embeddings towards zero. We do so in the non-convex regime $d \ll n$, reflecting how embedding dimensions are chosen in the real world. We moreover highlight empirically that using ℓ_2 regularization with node2vec leads to competitive performance on downstream tasks, when compared to embeddings produced from more recent encoding and training architectures.

We end with a brief discussion of some limitations of our works, and directions for future work. Generally, graphons are not realistic models for graphs; we suggest that our work could be extended to frameworks such as graphexes [10, 11, 58] which can produce more realistic degree distributions for networks, but have enough underlying latent exchangeability for our arguments to go through. We ignore aspects of optimization, i.e whether the minima of (10) are obtained in practice, which we believe would be an interesting area of future research. As for our experimental results, we note that methods such as GraphSAGE are better than node2vec in that they provide embeddings for data unobserved during training, and also scale better to larger networks. Consequently, we believe our experiments should be used primarily as motivation to investigate better methods for incorporating nodal covariates into network embedding models, and how to regularize embeddings produced by encoder methods such as GCNs or GraphSAGE.

References

[1] D. J. Aldous. "Representations for partially exchangeable arrays of random variables". en. In: *Journal of Multivariate Analysis* 11.4 (Dec. 1981), pp. 581–598. ISSN: 0047-259X. DOI: 10.1016/0047-259X(81)90099-3.

[2] C. D. Aliprantis and K. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. en. 3rd ed. Berlin Heidelberg: Springer-Verlag, 2006. ISBN: 978-3-540-29586-0. DOI: 10.1007/3-540-29587-9.

[3] A. Athreya et al. "Statistical Inference on Random Dot Product Graphs: a Survey". In: *Journal of Machine Learning Research* 18.226 (2018), pp. 1–92. ISSN: 1533-7928.

[4] F. R. Bach. "Consistency of Trace Norm Minimization". In: *The Journal of Machine Learning Research* 9 (June 2008), pp. 1019–1048. ISSN: 1532-4435.

[5] V. Barbu and T. Precupanu. *Convexity and Optimization in Banach Spaces*. en. 4th ed. Springer Monographs in Mathematics. Springer Netherlands, 2012. ISBN: 978-94-007-2246-0.

[6] M. Belkin and P. Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation". In: *Neural Computation* 15.6 (June 2003), pp. 1373–1396. ISSN: 0899-7667. DOI: 10.1162/089976603321780317.

[7] D. P. Bertsekas and J. N. Tsitsiklis. "Gradient Convergence in Gradient methods with Errors". In: *SIAM Journal on Optimization* 10.3 (Jan. 2000). Publisher: Society for Industrial and Applied Mathematics, pp. 627–642. ISSN: 1052-6234. DOI: 10.1137/S1052623497331063.

[8] C. Borgs, J. Chayes, and A. Smith. "Private Graphon Estimation for Sparse Graphs". In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015.

[9] C. Borgs et al. "Revealing Network Structure, Confidentially: Improved Rates for Node-Private Graphon Estimation". In: *arXiv:1810.02183 [cs, math, stat]* (Oct. 2018).

[10] C. Borgs et al. "Sampling perspectives on sparse exchangeable graphs". In: *The Annals of Probability* 47.5 (Sept. 2019). Publisher: Institute of Mathematical Statistics, pp. 2754–2800. ISSN: 0091-1798, 2168-894X. DOI: 10.1214/18-AOP1320.

[11] C. Borgs et al. "Sparse Exchangeable Graphs and Their Limits via Graphon Processes". In: *Journal of Machine Learning Research* 18.210 (2018), pp. 1–71.

[12] L. Bottou. "Une approche théorique de l'apprentissage connexioniste ; applications à la reconnaissance de la parole". These de doctorat. Paris 11, Jan. 1991.

[13] L. Bottou, F. E. Curtis, and J. Nocedal. *Optimization Methods for Large-Scale Machine Learning*. Tech. rep. arXiv:1606.04838. arXiv:1606.04838 [cs, math, stat] type: article. arXiv, Feb. 2018. DOI: 10.48550/arXiv.1606.04838.

[14] H. Brézis. *Functional analysis, Sobolev spaces and partial differential equations*. New York London: Springer, 2011. ISBN: 978-0-387-70914-7.

[15] A. D. Broido and A. Clauset. "Scale-free networks are rare". In: *Nature Communications* 10.1 (Dec. 2019). arXiv: 1801.03400, p. 1017. ISSN: 2041-1723. DOI: 10.1038/s41467-019-08746-5.

[16] S. Chatterjee. "Concentration inequalities with exchangeable pairs (Ph.D. thesis)". In: *arXiv:math/0507526* (July 2005). arXiv: math/0507526.

[17] S. Chatterjee. "Matrix estimation by Universal Singular Value Thresholding". In: *The Annals of Statistics* 43.1 (Feb. 2015). Publisher: Institute of Mathematical Statistics, pp. 177–214. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/14-AOS1272.

[18] K. Chen, H. Dong, and K.-S. Chan. "Reduced rank regression via adaptive nuclear norm penalization". In: *Biometrika* 100.4 (Dec. 2013), pp. 901–920. ISSN: 0006-3444. DOI: 10.1093/biomet/ast036.

[19] C. Data61. *StellarGraph Machine Learning Library*. Publication Title: GitHub Repository. 2018.

[20] A. Davison and M. Austern. "Asymptotics of Network Embeddings Learned via Subsampling". In: *arXiv:2107.02363 [cs, math, stat]* (July 2021). arXiv: 2107.02363.

[21] C. Gao, Y. Lu, and H. H. Zhou. "Rate-optimal graphon estimation". In: *The Annals of Statistics* 43.6 (Dec. 2015). Publisher: Institute of Mathematical Statistics, pp. 2624–2652. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/15-AOS1354.

[22] S. Ghadimi and G. Lan. "Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming". In: *SIAM Journal on Optimization* 23.4 (Jan. 2013). Publisher: Society for

Industrial and Applied Mathematics, pp. 2341–2368. ISSN: 1052-6234. DOI: 10.1137/120880811.

[23] P. Goyal and E. Ferrara. "Graph embedding techniques, applications, and performance: A survey". In: *Knowledge-Based Systems* 151 (July 2018), pp. 78–94. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.03.022.

[24] A. Grover and J. Leskovec. "node2vec: Scalable Feature Learning for Networks". In: ACM, Aug. 2016, pp. 855–864. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939754.

[25] W. Hamilton, Z. Ying, and J. Leskovec. "Inductive Representation Learning on Large Graphs". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[26] W. L. Hamilton, R. Ying, and J. Leskovec. "Representation Learning on Graphs: Methods and Applications." In: *IEEE Data Eng. Bull.* 40.3 (2017). Number: 3, pp. 52–74.

[27] C. Heil. *Metrics, Norms, Inner Products, and Operator Theory*. en. Applied and Numerical Harmonic Analysis. Birkhäuser Basel, 2018. ISBN: 978-3-319-65321-1. DOI: 10.1007/978-3-319-65322-8.

[28] A. J. Hoffman and H. W. Wielandt. "The variation of the spectrum of a normal matrix". In: *Duke Mathematical Journal* 20 (1953), pp. 37–39. ISSN: 0012-7094.

[29] P. H. Le-Khac, G. Healy, and A. F. Smeaton. "Contrastive Representation Learning: A Framework and Review". In: *IEEE Access* 8 (2020). arXiv: 2010.05113, pp. 193907–193934. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3031549.

[30] T. N. Kipf and M. Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *arXiv:1609.02907 [cs, stat]* (Feb. 2017). arXiv: 1609.02907.

[31] O. Klopp, A. B. Tsybakov, and N. Verzelen. "Oracle Inequalities For Network Models and Sparse Graphon Estimation". In: *The Annals of Statistics* 45.1 (2017). Publisher: Institute of Mathematical Statistics, pp. 316–354. ISSN: 0090-5364.

[32] V. Koltchinskii and E. Giné. "Random Matrix Approximation of Spectra of Integral Operators". In: *Bernoulli* 6.1 (2000). Publisher: International Statistical Institute (ISI) and Bernoulli Society for Mathematical Statistics and Probability, pp. 113–167. ISSN: 1350-7265. DOI: 10.2307/3318636.

[33] V. Koltchinskii, K. Lounici, and A. B. Tsybakov. "Nuclear-Norm Penalization and Optimal Rates for Noisy Low-Rank Matrix Completion". In: *The Annals of Statistics* 39.5 (2011). Publisher: Institute of Mathematical Statistics, pp. 2302–2329. ISSN: 0090-5364.

[34] A. Krogh and J. A. Hertz. "A simple weight decay can improve generalization". In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS'91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Dec. 1991, pp. 950–957. ISBN: 978-1-55860-222-9.

[35] P. Latouche and S. Robin. "Variational Bayes model averaging for graphon functions and motif frequencies inference in W-graph models". en. In: *Statistics and Computing* 26.6 (Nov. 2016), pp. 1173–1185. ISSN: 1573-1375. DOI: 10.1007/s11222-015-9607-0.

[36] J. Lei. "Network representation using graph root distributions". In: *The Annals of Statistics* 49.2 (Apr. 2021). Publisher: Institute of Mathematical Statistics, pp. 745–768. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/20-AOS1976.

[37] J. Lei and A. Rinaldo. "Consistency of spectral clustering in stochastic block models". In: *The Annals of Statistics* 43.1 (Feb. 2015). arXiv: 1312.2050. ISSN: 0090-5364. DOI: 10.1214/14-AOS1274.

[38] K. D. Levin et al. "Limit theorems for out-of-sample extensions of the adjacency and Laplacian spectral embeddings". In: *Journal of Machine Learning Research* 22.194 (2021), pp. 1–59. ISSN: 1533-7928.

[39] X. Li and F. Orabona. "On the Convergence of Stochastic Gradient Descent with Adaptive Stepsizes". en. In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2019, pp. 983–992.

[40] L. Lovász. *Large Networks and Graph Limits.* Vol. 60. Colloquium Publications. American Mathematical Society, 2012. ISBN: 978-0-8218-9085-1.

[41] O. Marchal and J. Arbel. "On the sub-Gaussianity of the Beta and Dirichlet distributions". EN. In: *Electronic Communications in Probability* 22 (2017). Publisher: The Institute of Mathematical Statistics and the Bernoulli Society. ISSN: 1083-589X. DOI: 10.1214/17-ECP92.

[42] T. Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". en. In: *Advances in Neural Information Processing Systems* 26 (2013), pp. 3111–3119.

[43] K. Oono and T. Suzuki. "Graph Neural Networks Exponentially Lose Expressive Power for Node Classification". In: *arXiv:1905.10947 [cs, stat]* (Jan. 2021). arXiv: 1905.10947.

[44] B. Perozzi, R. Al-Rfou, and S. Skiena. "DeepWalk: Online Learning of Social Representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14* (2014). arXiv: 1403.6652, pp. 701–710. DOI: 10.1145/2623330.2623732.

[45] J. Peypouquet. *Convex Optimization in Normed Spaces: Theory, Methods and Examples*. en. SpringerBriefs in Optimization. Springer International Publishing, 2015. ISBN: 978-3-319-13709-4.

[46] J. Qiu et al. "Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18* (2018). arXiv: 1710.02971, pp. 459–467. DOI: 10.1145/3159652.3159706.

[47] J. B. Reade. "Eigen-values of Lipschitz kernels". en. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 93.1 (Jan. 1983). Number: 1 Publisher: Cambridge University Press, pp. 135–140. ISSN: 1469-8064, 0305-0041. DOI: 10.1017/S0305004100060412.

[48] J. B. Reade. "Eigenvalues of Positive Definite Kernels". In: *SIAM Journal on Mathematical Analysis* 14.1 (Jan. 1983). Number: 1 Publisher: Society for Industrial and Applied Mathematics, pp. 152–157. ISSN: 0036-1410. DOI: 10.1137/0514012.

[49] B. Recht, M. Fazel, and P. A. Parrilo. "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization". en. In: (June 2007). DOI: 10.1137/070697835.

[50] B. Recht, M. Fazel, and P. A. Parrilo. "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization". In: *SIAM Review* 52.3 (Jan. 2010). Publisher: Society for Industrial and Applied Mathematics, pp. 471–501. ISSN: 0036-1445. DOI: 10.1137/070697835.

[51] H. Robbins and S. Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (Sept. 1951). Publisher: Institute of Mathematical Statistics, pp. 400–407. ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177729586.

[52] P. Rubin-Delanchy et al. "A statistical interpretation of spectral embedding: the generalised random dot product graph". In: *arXiv:1709.05506 [cs, stat]* (Sept. 2017). arXiv: 1709.05506.

[53] V. S. Sunder. *Operators on Hilbert Space*. en. Texts and Readings in Mathematics. Springer Singapore, 2016. ISBN: 978-981-10-1816-9. DOI: 10.1007/978-981-10-1816-9.

[54] J. Tang et al. "LINE: Large-scale Information Network Embedding". In: *Proceedings of the 24th International Conference on World Wide Web* (May 2015). arXiv: 1503.03578, pp. 1067–1077. DOI: 10.1145/2736277.2741093.

[55] M. Tang and C. E. Priebe. "Limit theorems for eigenvectors of the normalized Laplacian for random graphs". In: *The Annals of Statistics* 46.5 (Oct. 2018). Publisher: Institute of Mathematical Statistics, pp. 2360–2415. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/17-AOS1623.

[56] S. Tu et al. "Low-rank Solutions of Linear Matrix Equations via Procrustes Flow". en. In: *Proceedings of The 33rd International Conference on Machine Learning*. ISSN: 1938-7228. PMLR, June 2016, pp. 964–973.

[57] M. Udell et al. "Generalized Low Rank Models". In: *arXiv:1410.0342 [cs, math, stat]* (May 2015). arXiv: 1410.0342.

[58] V. Veitch and D. M. Roy. "The Class of Random Graphs Arising from Exchangeable Random Measures". In: *arXiv:1512.03099 [cs, math, stat]* (Dec. 2015). arXiv: 1512.03099.

[59] V. Veitch et al. "Empirical Risk Minimization and Stochastic Gradient Descent for Relational Data". en. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2019, pp. 1733–1742.

[60] P. Veličković et al. "Deep Graph Infomax". In: *arXiv:1809.10341 [cs, math, stat]* (Dec. 2018). arXiv: 1809.10341.

[61] P. J. Wolfe and S. C. Olhede. "Nonparametric graphon estimation". In: *arXiv:1309.5936 [math, stat]* (Sept. 2013). arXiv: 1309.5936.

[62] J. Xu. "Rates of Convergence of Spectral Methods for Graphon Estimation". en. In: *Proceedings of the 35th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2018, pp. 5433–5442.

[63] Y. Zhang, J. Lu, and O. Shai. "Improve Network Embeddings with Regularization". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. CIKM '18. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 1643–1646. ISBN: 978-1-4503-6014-2. DOI: 10.1145/3269206.3269320.

[64] Y. Zhang and M. Tang. "Consistency of random-walk based network embedding algorithms". In: *arXiv:2101.07354 [cs, stat]* (Jan. 2021). arXiv: 2101.07354.

[65] B. Zhou, X. Meng, and H. E. Stanley. "Power-law distribution of degree–degree distance: A better representation of the scale-free property of complex networks". In: *Proceedings of the National Academy of Sciences* 117.26 (June 2020). Publisher: Proceedings of the National Academy of Sciences, pp. 14812–14818. DOI: 10.1073/pnas.1918901117.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See the first paragraph in the related works section; Remark 1; the paragraph between Theorem 3 and 4; the conclusion; Remark 6 in the appendix re: our rates of convergence.
   (c) Did you discuss any potential negative societal impacts of your work? [No] The paper discusses the theoretical effects of a particular type of regularization scheme, and does not advocate for any methodological changes or gives particular applications; I believe that making *any* types of claims about the societal impacts of this work would be excessively speculative, to be frank.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Yes (`https://neurips.cc/public/EthicsGuidelines`).
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2 and the relevant theorem statements; the one theorem in the main body which does not state them fully (Theorem 2) mentions the corresponding version of the theorem in the appendix (Theorem 6) which has the full regularity conditions.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix F.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix F.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1 and Figure 2.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix F.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.1 and Appendix F.

(b) Did you mention the license of the assets? [Yes] See Appendix F.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] No new assets were created.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Standard datasets were used, which do not contain information about/curated from individuals.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Datsets used were standard citation network datasets with no identifiable/textual information provided.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No crowdsourcing/human subjects were used.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No crowdsourcing/human subjects were used.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No crowdsourcing/human subjects were used.