Convergence Guarantees for the DeepWalk Embedding on Block Models

Christopher Harker [1]   **Aditya Bhaskara** [1]

Abstract

Graph embeddings have emerged as a powerful tool for understanding the structure of graphs. Unlike classical spectral methods, recent methods such as DeepWalk, Node2Vec, etc. are based on solving nonlinear optimization problems on the graph, using local information obtained by performing random walks. These techniques have empirically been shown to produce "better" embeddings than their classical counterparts. However, due to their reliance on solving a nonconvex optimization problem, obtaining theoretical guarantees on the properties of the solution has remained a challenge, even for simple classes of graphs. In this work, we show convergence properties for the DeepWalk algorithm on graphs obtained from the Stochastic Block Model (SBM). Despite being simplistic, the SBM has proved to be a classic model for analyzing the behavior of algorithms on large graphs. Our results mirror the existing ones for spectral embeddings on SBMs, showing that even in the case of one-dimensional embeddings, the output of the DeepWalk algorithm provably recovers the cluster structure with high probability.

1. Introduction

Inspired by the Skip-gram model (Mikolov et al., 2013a;b) and related word embedding algorithms in the field of natural language processing, Perozzi et al. (2014), Tang et al. (2015), and Grover & Leskovec (2016) developed new methods to embed the nodes of a graph into geometric space. These methods treat random walks as akin to sentences and construct analogues to n-grams by sliding a fixed size window along the random walk. The probability of a node j appearing in the context of a node i is modeled as a

function of the nodes' embeddings. This leads to a vector representation of each node in d dimensions, which is then useful in addressing several machine learning problems as input features to downstream tasks, including node classification (Hamilton et al., 2017; Perozzi et al., 2014; Grover & Leskovec, 2016; Tang et al., 2015), link prediction (Grover & Leskovec, 2016; Tang et al., 2015; Backstrom & Leskovec, 2011) and community detection (Wang et al., 2017; Barot et al., 2021; Zhang & Tang, 2021; Davison et al., 2023).

Predating nonlinear embedding methods, spectral embeddings have long been a classic topic in theoretical computer science and mathematics. Early algorithms for graph clustering and partitioning use the top eigenvectors of a graph's Laplacian matrix as node embeddings (Hall, 1970; Alon, 1986; Sinclair & Jerrum, 1989; Linial et al., 1994; McSherry, 2001; Spielman & Teng, 2007; Arora et al., 2009) and the properties of spectral embeddings have been thoroughly studied in works such as Belkin & Niyogi (2001), Ng et al. (2001), Von Luxburg (2007), and Rohe et al. (2011).

Most of the existing theoretical results surrounding embeddings produced by DeepWalk (Perozzi et al., 2014) and node2vec (Grover & Leskovec, 2016) reframe the algorithm as a matrix factorization problem, then use methods common in the analysis of spectral embeddings to study their properties (Barot et al., 2021; Zhang & Tang, 2021; Qiu et al., 2018). Levy & Goldberg (2014) show that Skip-gram with negative sampling (SGNS) is implicitly performing matrix factorization of a shifted point-wise mutual information (PMI) matrix when the embedding dimension is at least as large as the number of nodes in the graph. This result inspired analyses of the properties of DeepWalk and node2vec embeddings by showing that they are also performing matrix factorization of a shifted PMI matrix (Qiu et al., 2018). The works of Zhang & Tang (2021) and Barot et al. (2021) then show that a spectral decomposition of this matrix can be used to recover communities in graphs drawn from stochastic block models.

A major drawback of these prior works is that the matrix factorization characterization holds *only when* the embedding dimension is large ($> n$, the number of nodes in the graph!), which is not usually the case in practice (Perozzi et al., 2014). Furthermore, the matrix factorization formu-

---

[1]Kahlert School of Computing, University of Utah, Salt Lake City, UT, USA. Correspondence to: Christopher Harker <chris.harker@utah.edu>, Aditya Bhaskara <bhaskaraaditya@gmail.com>.

lation of DeepWalk and node2vec is not generally used in practice, but rather the embeddings are learned through optimizing an objective using gradient descent. In this work, we answer the fundamental question: *can the dynamics of gradient descent be formally analyzed for low-dimensional embeddings of natural graph classes?*

One key challenge in such analyses is the nonlinearity and nonconvexity of their objectives. When the embedding dimension is at least as large as the number of nodes in the graph, it turns out that a locally optimum solution has a closed form structure, and its properties can be analyzed (Zhang & Tang, 2021; Barot et al., 2021; Levy & Goldberg, 2014; Qiu et al., 2018).

However, since most applications use only constant-dimensional embeddings, these results do no apply. Recently, the work by Harker & Bhaskara (2023) analyzes the objective function for low-dimensional embeddings, and shows that for graphs drawn from the stochastic block model (SBM), the DeepWalk objective has a global minimum that has a well-clustered structure. However, their work is restricted to the case of 2-block SBMs, and more importantly, does not study the question of whether gradient descent (or any other heuristic) converges to the optimal solution. Another recent work of Davison et al. (2023) examines the asymptotic behavior of node2vec embeddings learned by minimizing the SGNS objective on graphs obtained fom SBMs. But once again, they do not answer the algorithmic question of how to obtain the optimal solution (e.g., via gradient descent).

In contrast to these works, we analyze the dynamics of the gradient descent update procedure and show that when the graph is drawn from a symmetric SBM, the embedding vectors of nodes within a block are clustered together, while the vectors corresponding to nodes in different blocks are farther apart. We perform the analysis for the DeepWalk procedure, which minimizes a nonconvex objective obtained by performing random walks.

Our work is inspired by recent works that study the theoretical properties of the t-SNE algorithm, which shares many of the same challenges as DeepWalk due to its nonlinear and nonconvex objective. The works of Linderman & Steinerberger (2019), Arora et al. (2018), and Cai & Ma (2022) examine the dynamical properties of t-SNE's gradient descent updates and show that similar data points are clustered together while separating from dissimilar data points.

1.1. Our Results

We consider graphs drawn from a stochastic block model (SBM). In the simplest setting, we have three parameters, an integer $K \geq 2$ (number of blocks) and probabilities $p, q \in (0, 1)$. The vertices are divided into K parts or

clusters at random, and an edge is placed between two vertices in the same cluster with probability p, and between vertices in two different clusters with probability q. We assume $p > q$, and that both parameters are $> n^{\rho-1}$ for some parameter $\rho \in (0, 1)$. For details about the SBM and graph generation, we refer to Section 2.3. Our main result can be stated as follows.

Theorem 1.1 (Informal). *Given a graph G drawn from an SBM with K blocks and parameters p and q, DeepWalk embeddings, obtained by initializing a solution in small enough ball $\|\boldsymbol{w}^{(0)}\| \leq \epsilon$, and training with gradient descent with learning rate $\eta > 0$ small enough, approximately recovers communities with high probability.*

For more formal statements, see theorems 3.8 and 3.10, which respectively upper bound the spread of embeddings within clusters and lower bound the separation of embeddings across clusters. Theorem 3.11 then gives a bound on the fraction of each community that can be recovered (thus formalizing the notion of approximate recovery above). The main technical challenge in our result is reasoning about the nonlinear gradient update rule. The idea in our proofs is to show that if we initialize our solution in a small enough ball, the gradient update is close enough to a "linearized" update, which results in a sufficiently good cluster structure for the overall solution. This is similar in spirit to other recent work that show that small random initializations result in dynamics that are like spectral updates (see (Stöger & Soltanolkotabi, 2021; Satpathi & Srikant, 2021) and references therein).

We also remark that we prove our main results even for the case of one-dimensional embeddings. We find this somewhat surprising because typically (e.g., for spectral algorithms), separation results for SBMs with K clusters hold only when the embedding dimension is $\geq K$. We expect our analysis to extend to the case of higher dimensional embeddings (as the 1D analysis can be applied to each coordinate).

One weakness of our result is that unlike traditional results for recovery guarantees for SBM, we do not give a precise characterization in terms of the difference $(p - q)$ and recovery accuracy. Instead, our arguments rely on a closeness assumption between the empirical and *expected* co-occurrence matrices, stated as Assumption 3.2. It is an interesting open direction to study tight recovery guarantees in terms of p, q, K.

On the experimental side, we validate our results: we show a clear separation between the embeddings of vertices across clusters for different choices of the embedding dimension. We also show that the approximation via linearization potentially holds for a fairly large regime of parameters, even beyond the ones we use for our theoretical guarantees.

2. Background

2.1. Basic Notation

We start with common notation that we use throughout the paper. For a column vector $\boldsymbol{x} = (x_1, ..., x_n) \in \mathbb{R}^n$, we define the ℓ_2 norm as $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ and the ℓ_∞ norm as $\|\boldsymbol{x}\|_\infty = \max_{1 \le i \le n} |x_i|$. We denote by $diag(\boldsymbol{x}) \in \mathbb{R}^{n \times n}$ the diagonal matrix whose i^{th} diagonal entry is x_i. We denote the all-ones vector as $\mathbf{1} = (1, ..., 1) \in \mathbb{R}^n$. The all-ones matrix (of dimensions $n \times n$ unless specified otherwise) will be denoted by $\boldsymbol{J}$. We denote the dot product between two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

For a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ we define the spectral norm as $\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|$, its ℓ_∞ norm as $\|\boldsymbol{A}\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^n |A_{ij}|$, its max norm as $\|\boldsymbol{A}\|_{max} = \max_{1 \le i,j \le n} |A_{ij}|$, and its Frobenius norm as $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n A_{ij}^2}$. We define the degree operator $\boldsymbol{D}$: $\mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ as $\boldsymbol{D}(\boldsymbol{A}) = diag(\boldsymbol{A}\mathbf{1})$. At times we use $\boldsymbol{D_A}$ in place of $\boldsymbol{D}(\boldsymbol{A})$ when it is more convenient.

For a graph $G = (V, E)$ with n nodes and $|E|$ edges, we denote the adjacency matrix of the graph as $\boldsymbol{A}$. For a node i, its degree is $d_i = \sum_{j=1} A_{ij}$. We let $\boldsymbol{P} = \boldsymbol{D_A}^{-1} \boldsymbol{A}$ denote the transition matrix of the graph.

2.2. The DeepWalk Algorithm

Let $G = (V, E)$ be a graph with n nodes. The DeepWalk algorithm consists of two main parts (Perozzi et al., 2014). First, a *co-occurrence matrix* is constructed using random walks on the graph. Second, two embedding vectors are learned for every vertex in the graph. These are referred to as the *node* and *context* embeddings; they are learned by minimizing a nonconvex objective function.[1]

Constructing the Co-occurrence Matrix. Given a graph $G = (V, E)$, the algorithm first performs r random walks of length L. For each random walk, a window of size T slides along the generated path. Let $w^{(m)}$ denote the path of L nodes generated by the mth random walk, and let $w_k^{(m)}$ denote the kth step of the mth random walk. Furthermore, we assume that the starting node of each walk is sampled from the stationary distribution $\Pr[w_1^{(m)} = i] = \frac{d_i}{2|E|}$ over the graph G.

As the window of size T is slid along the path, the entries of a matrix $\boldsymbol{C}$ are updated. The entries C_{ij} of this matrix contain the number of times that a node j appears in the context window of node i. Formally,

---

[1]While having separate word and context embeddings have intuitive meaning in the language context, the difference is not so clear for graphs. In some implementations of DeepWalk, the same embedding is used. We use separate embeddings to remain faithful to the original formulation.

$$C_{ij} = \sum_{t=1}^T \sum_{m=1}^r \sum_{k=1}^{L-t} \mathbb{1}\{w_k^{(m)} = i, w_{k+t}^{(m)} = j\}$$
$$+ \sum_{t=1}^T \sum_{m=1}^r \sum_{k=1}^{L-t} \mathbb{1}\{w_k^{(m)} = j, w_{k+t}^{(m)} = i\}.$$

Limiting forms of the co-occurrence matrix are explored in many prior works. Different variations of the limiting form exist depending on whether the length of the walk L goes to ∞ (Qiu et al., 2018), whether the number of walks r goes to ∞ (Zhang & Tang, 2021; Barot et al., 2021) or whether the window size goes to ∞ (Chanpuriya & Musco, 2020). Obtaining more quantitative concentration bounds on this matrix have also been studied (Qiu et al., 2020; Kloepfer et al., 2021). Proofs of the following lemma can be found in Zhang & Tang (2021), Barot et al. (2021), and Harker & Bhaskara (2023).

Lemma 2.1. *Let $\boldsymbol{A}$ be an adjacency matrix of a fixed graph G and let $w_k^{(m)}$ denote the kth step of the mth random walk generated by the DeepWalk algorithm. Let $\pi_i = \frac{d_i}{2|E|}$ and let $(\boldsymbol{P}^t)_{ij} = \Pr[w_{t+1} = j | w_1 = i]$. Then as $r \to \infty$,*

$$\frac{C_{ij}}{r} \xrightarrow{a.s} 2 \sum_{t=1}^T (L - t) \cdot \pi_i (\boldsymbol{P}^t)_{ij}. \tag{1}$$

Computing Embeddings Given a co-occurrence matrix $\boldsymbol{C}$, we compute d-dimensional node embeddings $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n \times d}$, where each row $\boldsymbol{x}_i, \boldsymbol{y}_i$ are the node and context representations of vertex i, by minimizing the following objective function:

$$\mathcal{L}(\boldsymbol{C}; \boldsymbol{X}, \boldsymbol{Y}) = -\sum_{i=1}^n \sum_{j=1}^n C_{ij} \log \left(\frac{\exp \langle \boldsymbol{x}_i, \boldsymbol{y}_j \rangle}{\sum_{k=1}^n \exp \langle \boldsymbol{x}_i \boldsymbol{y}_k \rangle} \right). \tag{2}$$

This is typically done using gradient descent. At any given iteration t of gradient descent, let $\boldsymbol{Q}^{(t)}$ be the matrix defined by

$$\boldsymbol{Q}_{ij}^{(t)} = \frac{\exp \langle \boldsymbol{x}_i^{(t)}, \boldsymbol{y}_j^{(t)} \rangle}{\sum_{k=1}^n \exp \langle \boldsymbol{x}_i^{(t)}, \boldsymbol{y}_k^{(t)} \rangle}. \tag{3}$$

For a co-occurrence matrix $\boldsymbol{C}$ and a learning rate $\eta > 0$, the update equations in matrix form are

$$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} - \eta \left(\boldsymbol{D_C} \boldsymbol{Q}^{(t)} - \boldsymbol{C} \right) \boldsymbol{Y}, \tag{4}$$

$$\boldsymbol{Y}^{(t+1)} = \boldsymbol{Y}^{(t)} - \eta \left(\boldsymbol{D_C} \boldsymbol{Q}^{(t)} - \boldsymbol{C} \right)^\top \boldsymbol{X}. \tag{5}$$

If we let $\boldsymbol{W}^{(t)} = \begin{bmatrix} \boldsymbol{X}^{(t)} \\ \boldsymbol{Y}^{(t)} \end{bmatrix}$ and $\boldsymbol{G}^{(t)} = \boldsymbol{D_C} \boldsymbol{Q}^{(t)} - \boldsymbol{C}$. Then we can write the updates jointly as

$$\boldsymbol{W}^{(t+1)} = \begin{bmatrix} \boldsymbol{I} & -\eta \boldsymbol{G}^{(t)} \\ -\eta \boldsymbol{G}^{(t)\top} & \boldsymbol{I} \end{bmatrix} \boldsymbol{W}^{(t)}. \tag{6}$$

The main challenge in the analysis is working with the matrices $Q^{(t)}$, whose entries change as we update the embeddings. To keep the analysis clean, we will work with the case $d = 1$, as discussed earlier. Here, the embeddings are defined simply by vectors x and y, but the essential difficulty (of dealing with Q) remains.

2.3. Stochastic Block Models

The stochastic block model (SBM) (Holland et al., 1983) is a generalization of Erdős-Renyi random graphs. This model naturally generates graphs containing communities; therefore, it has been a popular choice of generative model studied in the theoretical analysis of community recovery algorithms (see, e.g. Abbe (2017) and references therein).

A K block stochastic block model (SBM) generates a random graph $G = (V, E)$ through a simple procedure. First, each node is first assigned to one of K blocks. We refer to $\mathbb{V}_k$ as the set of vertices that belong to community k. We define a community membership matrix $\Theta \in \{0, 1\}^{n \times K}$ where its entries $\Theta_{ik} = 1$ if node i belongs to community k and is 0 otherwise. Next, edges are assigned to each pair of nodes. We define a symmetric matrix $B \in [0, 1]^{K \times K}$ whose entries B_{ij} denote the probability of a node in cluster i being connected to a node in cluster j. Then the matrix $\widetilde{B} = \Theta B \Theta^\top$ is a block matrix of probabilities defined by the community membership matrix Θ. In the remainder of the paper, we assume that the vertex indices are permuted such that the first n/K correspond to the first cluster, the next n/K to the second cluster, and so on. In this case, we can also write $\widetilde{B} = B \otimes J_{\frac{n}{K} \times \frac{n}{K}}$, where $\otimes$ denotes the Kronecker product. Given a probability matrix $\widetilde{B}$, edges of the graph $G(V, E)$ are then independent Bernoulli random variables with $A_{ij} \sim Bern(\widetilde{B}_{ij})$, and $A_{ij} = A_{ji}$ for all $i < j$. Therefore, the probability of node i and node j being connected depends only on the communities to which i and j belong. To be consistent with later notation, we also define the expected adjacency matrix $\overline{A}$ as the matrix whose entries are the expected values of the corresponding entries in the adjacency matrix A; by definition, $\overline{A} = \widetilde{B}$.

In our analysis, we assume that the number of communities K is fixed and the communities are of equal size: $|\mathbb{V}_k| = \frac{n}{K}$. We also assume that the matrix B has diagonal entries equal to p and off-diagonal entries equal to q. This makes the probability matrix $\widetilde{B}$ a block matrix with K equally sized blocks. It has blocks of p along the diagonal and blocks of q off the diagonal.

3. Analysis

We break up the analysis into three main parts. First, we will show properties of the co-occurrence matrix that will be important for the analysis, describe the algorithm, and

set up the main notation for the analysis. Second, we show the main step of "linear approximation", where we argue that the gradient descent update can be expressed as a linear update plus an error term that is controlled by the length of the embedding solution. Finally, we analyze the gradient descent dynamics, and show the desired properties of the final solution.

For simplicity, we assume that n is large, and K is a constant. We also assume (for the analysis) that when writing down the co-occurrence matrix C, the vertices are permuted so that the blocks $\mathbb{V}_1, \mathbb{V}_2, \ldots, \mathbb{V}_K$ appear together (thus leading to the form of $\overline{C}$ below).

3.1. Co-occurrence Structure and Algorithm

Assume that we have a graph drawn from the symmetric SBM with K blocks and parameters p and q, and let C be the symmetric co-occurrence matrix obtained using random walks as described in Section 2.2. Our first step will be to prove that C is spectrally close to the matrix $\overline{C}$ defined as

$$\overline{C} = 2 \sum_{t=1}^{T} \frac{(L-t)}{n\overline{d}} D_{\overline{A}} \overline{P}^t, \tag{7}$$

where $\overline{P} = D_{\overline{A}}^{-1} \overline{A}$ and $\overline{d} = \frac{n}{K} p + \frac{n(K-1)}{K} q$ is the expected degree of the graph. Since the expected adjacency matrix $\overline{A}$ has a block structure (as described in Section 2.3), the matrix $\overline{C}$ also has a block structure. In other words, for some parameters a, b,

$$\overline{C} = \underbrace{\begin{bmatrix} a & b & \ldots & b \\ b & a & \ldots & b \\ \vdots & \vdots & \ddots & \vdots \\ b & b & \ldots & a \end{bmatrix}}_{K \times K} \otimes J_{\frac{n}{K} \times \frac{n}{K}}. \tag{8}$$

As before, $\otimes$ denotes the Kronecker product, making $\overline{C}$ a block matrix with $K \times K$ blocks, each of size $\frac{n}{K} \times \frac{n}{K}$, with a in the diagonal blocks and b in the off-diagonal blocks. In the case of symmetric SBMs, we can express a and b explicitly in terms of the SBM parameters p and q (see Appendix A.1).

The following lemma shows that C is spectrally close to the matrix $\overline{C}$. The proof can be found in Appendix A.2.

Lemma 3.1. *Suppose C is an $n \times n$ co-occurrence matrix constructed as in Lemma 2.1 from a graph G drawn from an SBM with K blocks and parameters $p > q \geqq n^{\rho-1}$, for some parameter $\rho \in (0, 1)$, and that the matrix $\overline{C}$ is defined as in Equation (7). Then for appropriate parameters a, b (defined in terms of p, q), we have $\|C - \overline{C}\| \leq c\|\overline{C}\|\sqrt{\frac{\log n}{n^\rho}}$ for some absolute constant c.*

This lemma allows us to reason about the eigenvalues of C using those of $\overline{C}$, via Weyl's Theorem and other matrix

perturbation bounds. Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ are the eigenvalues of $\overline{C}$. Then,

$$\lambda_i = \begin{cases} \frac{n}{K}(a + (K-1)b) & \text{if } i = 1, \\ \frac{n}{K}(a - b) & \text{if } i = 2, ..., K, \\ 0 & \text{if } i = K+1, ..., n. \end{cases}$$

The assumption that $p > q \geq n^{\rho - 1}$ made in Lemma 3.1 will ensure that the nonzero eigenvalues $\lambda_1, ..., \lambda_K$ are all $\Theta(\frac{1}{n})$, assuming that the length of the random walk L and number of communities K are constant (see Appendix A.1). As a result, the spectral norm $\|\overline{C}\| = O\left(\frac{1}{n}\right)$, which implies that $\|C - \overline{C}\| = \widetilde{O}\left(\frac{1}{\sqrt{n^{\rho+2}}}\right)$. Therefore, in order to simplify our calculations, we make the following assumption throughout the remainder of the paper:

Assumption 3.2. Suppose C is the $n \times n$ symmetric co-occurrence matrix obtained via random walks of length $L = O(1)$ as in Lemma 2.1, and $\overline{C}$ is defined as in Equation (7). Then we assume that $C = \overline{C} + R$ for an error matrix R that is symmetric and satisfies $\|R\| < \frac{c}{\sqrt{n^{\rho+2}}}$ for some absolute constant c.

The corresponding eigenvectors are also easy to characterize: the top eigenvector is the all-ones vector, which we write as $\frac{1}{\sqrt{n}}$. The next $(K-1)$ eigenvalues are all equal, so the corresponding eigenvectors are not unique. One (nonorthogonal) basis for their span is $\{e_{\mathbb{V}_i} - e_{\mathbb{V}_1}\}_{i=2}^K$, where e_S is a shorthand for $\mathbf{1}_S$ (the vector that is 1 in the jth position if $j \in S$ and 0 otherwise).

Next, the following matrix M and the corresponding $\overline{M}$ play a key role in our analysis:

$$M := D_C \frac{J}{n} - C, \quad \overline{M} := D_{\overline{C}} \frac{J}{n} - \overline{C},$$

where D denotes the degree operator (see Section 2.1). Since $C - \overline{C} = R$ and the degree operator is linear, we have

$$M - \overline{M} = D_R \frac{J}{n} - R = -\left(I - \frac{J}{n}\right)R.$$

This implies that $\|M - \overline{M}\| \leq \|R\| < \frac{c}{\sqrt{n^{\rho+2}}}$. Now the eigenvalues of $\overline{M}$ are easy to see: along the all-ones vector, the eigenvalue becomes 0, and all the other eigenvalues stay the same. Thus, $\overline{M}$ has exactly $(K-1)$ nonzero eigenvalues, all equal to $\frac{n}{K}(b - a)$.

For concreteness, we provide the gradient descent procedure that we analyze in Algorithm 1. We initialize the embeddings randomly and normalize them so that their norm is small. Then we run gradient descent until the norm of the embeddings reaches a predefined size.

Algorithm 1 DeepWalk Gradient Descent

Input: Co-occurrence matrix $C \in \mathbb{R}^{n \times n}$; learning rate $\eta > 0$; parameters $\epsilon = \frac{1}{n^{2/3}}$ and $\Delta = n^{1/6}$

Initialize: $t = 0$; $x^{(0)}, y^{(0)} \in \mathbb{R}^n$ with $x_i, y_i \sim \mathcal{N}(0, 1)$ for all $i \in [n]$; $w^{(0)} = \begin{bmatrix} x^{(0)} \\ y^{(0)} \end{bmatrix}$

Normalize $w^{(0)}$: $w^{(0)} \leftarrow \frac{w^{(0)}}{\|w^{(0)}\|}\epsilon$

repeat
    $t \leftarrow t + 1$;
    Compute $Q^{(t)}$ where $Q_{ij}^{(t)} = \frac{\exp\langle x_i^{(t)}, y_j^{(t)}\rangle}{\sum_{k=1}^n \exp\langle x_i^{(t)}, y_k^{(t)}\rangle}$ for all $i, j \in [n]$;
    $G^{(t)} = D_C Q^{(t)} - C$;
    $w^{(t)} \leftarrow \begin{bmatrix} I & -\eta G^{(t)} \\ -\eta G^{(t)\top} & I \end{bmatrix} w^{(t)}$
until $\|w^{(t)}\| \geq \epsilon\Delta$
$t_f = t$;
Return: $w^{(t_f)}$; t_f

3.2. Linear Approximation of Gradient Update

Next, we show that the matrix $Q^{(t)}$ in Equation (6) (gradient descent step) can be approximated simply by the all-ones matrix (suitably scaled). This allows us to obtain a linear approximation to the w update.

Proposition 3.3. *Let $x, y \in \mathbb{R}^n$ and $w = \begin{bmatrix} x \\ y \end{bmatrix}$, and suppose that $\|w\| < \epsilon$ for some $\epsilon \in (0, 1/2)$. Let Q be the matrix defined by $Q_{ij} = \frac{\exp\langle x_i, y_j\rangle}{\sum_{k=1}^n \exp\langle x_i, y_k\rangle}$. Then*

$$\left\| Q - \frac{1}{n}J \right\|_F \leq \epsilon^2.$$

The proof is deferred to section B.1. It is important to note that ϵ^2 is a bound on the Frobenius norm (not the square of the Frobenius norm). This implies that if the node and context embeddings exist in a small enough ball, then the softmax matrix $Q^{(t)}$ behaves like the fixed matrix $\frac{1}{n}J$.

Utilizing this observation, we can rewrite the update Equation (6) in terms of a linear term and an "error" term. Recall that $M = D_C \frac{J}{n} - C$. Thus we can write (6) as

$$w^{(t+1)} = \begin{bmatrix} I & -\eta M \\ -\eta M & I \end{bmatrix} w^{(t)} - \eta E^{(t)} w^{(t)}, \quad (9)$$

where

$$E^{(t)} = \begin{bmatrix} 0 & D_C(Q^{(t)} - \frac{1}{n}J) \\ D_C(Q^{(t)} - \frac{1}{n}J)^\top & 0 \end{bmatrix}.$$

The following is a consequence of Proposition 3.3 and the choice of a and b (see Appendix B.2).

Lemma 3.4. *Suppose the iterate $\boldsymbol{w}^{(t)}$ satisfies $\|\boldsymbol{w}^{(t)}\| \leq \epsilon$ for some $\epsilon \in (0, 1/2)$. Then $\|\boldsymbol{E}^{(t)}\|_F \leq 4\epsilon^2$.*

Next, let denote the "linear portion" of the update by $\boldsymbol{L}$. In other words, we define

$$\boldsymbol{L} := \begin{bmatrix} \boldsymbol{I} & -\eta\boldsymbol{M} \\ -\eta\boldsymbol{M} & \boldsymbol{I} \end{bmatrix}. \tag{10}$$

We have the following observations about the spectrum of $\boldsymbol{L}$. The proof is deferred to Appendix B.3.

Lemma 3.5. *Suppose $\boldsymbol{L}$ is defined as in (10). Then $\boldsymbol{L}$ has precisely $(K-1)$ eigenvalues that are $> (1 + \eta\gamma)$, where $\gamma = \frac{n(a-b)}{2K}$. All the other eigenvalues are $< (1 + \eta\frac{c}{\sqrt{n^{\rho+2}}})$. Finally, all the eigenvalues are in the interval $(1 - 4\eta\gamma, 1 + 4\eta\gamma)$, for γ as above.*

In what follows, let Π be the projector onto the span of the eigenvectors corresponding to the top $(K-1)$ eigenvalues of $\boldsymbol{L}$.

We begin by obtaining bounds on the number of iterations performed by the algorithm.

Lemma 3.6. *Let t_f be the number of iterations performed by Algorithm 1. With probability at least 0.9 over the choice of the initialization, we have*

$$\frac{1}{\eta} < t_f < \frac{4\log(n/\Delta)}{\eta}.$$

Proof. Using Lemma 3.5 along with the fact that $4\gamma \leq 1$, we have

$$\|\boldsymbol{L}\| \leq 1 + \eta.$$

Now, as long as $\|\boldsymbol{w}^{(t)}\| \leq \epsilon\Delta$, we have $\|\boldsymbol{E}^{(t)}\| \leq 4(\epsilon\Delta)^2$ using Lemma 3.4. By the choice of parameters ϵ, Δ, we will ensure that $4(\epsilon\Delta)^2 < 1$. Thus, we obtain that

$$\|\boldsymbol{w}^{(t+1)}\| < (1 + 2\eta)\|\boldsymbol{w}^{(t)}\|.$$

Now, since $\|\boldsymbol{w}^{(0)}\| = \epsilon$, $\|\boldsymbol{w}^{(t_f)}\| \geq \epsilon\Delta$, and $\Delta > 2$, the number of iterations $t_f > \frac{2}{\log(1+2\eta)} > \frac{1}{\eta}$.

To see the upper bound, we use more properties of $\boldsymbol{L}$ from Lemma 3.5. Specifically, suppose we define $\boldsymbol{z}^{(t)} = \Pi\boldsymbol{w}^{(t)}$ where Π is the projector onto the span of the eigenvectors corresponding to the top $(K-1)$ eigenvalues of $\boldsymbol{L}$. We argue that this component of $\boldsymbol{w}^{(t)}$ itself grows sufficiently. First, we note that because of the randomness in the initialization, we have that $\|\boldsymbol{z}^{(0)}\| \geq \frac{1}{10\sqrt{n}}\|\boldsymbol{w}^{(0)}\| = \frac{\epsilon}{10\sqrt{n}}$, with probability ≥ 0.9.

Then, by multiplying the update Equation (9) by Π, we get

$$\begin{aligned} \|\boldsymbol{z}^{(t+1)}\| &\geq (1 + \eta\gamma)\|\Pi\boldsymbol{w}^{(t)}\| - \|\Pi\boldsymbol{E}^{(t)}\boldsymbol{w}^{(t)}\| \\ &\geq (1 + \eta\gamma)\|\boldsymbol{z}^{(t)}\| - 4\eta(\epsilon\Delta)^3 \\ &\geq (1 + \frac{\eta\gamma}{2})\|\boldsymbol{z}^{(t)}\|. \end{aligned}$$

For the last inequality, we used the fact that $(\epsilon\Delta)^3 < \frac{\gamma\epsilon}{20\sqrt{n}}$, which follows from our assumptions on the parameters. We are also using the fact that $\boldsymbol{z}^{(t)}$ always has norm $> \frac{\epsilon}{10\sqrt{n}}$ which follows inductively from the above.

This establishes that $\|\boldsymbol{z}^{(t)}\| \geq (1 + \frac{\eta\gamma}{2})^{t_f}\|\boldsymbol{z}^{(0)}\|$. Since $\|\boldsymbol{z}^{(t)}\| \leq \|\boldsymbol{w}^{(t)}\| \leq 2\epsilon\Delta$, the desired upper bound on t_f follows. $\square$

As the final result in this part of the proof, we argue that most of the mass of $\boldsymbol{w}^{(t_f)}$ at the end of the algorithm is on the top $(K-1)$ eigenspace of $\boldsymbol{L}$. This is precisely what we would expect if the entire update was linear, i.e., if $\boldsymbol{w}^{(t_f)}$ where equal to $\boldsymbol{L}^{t_f}\boldsymbol{w}^{(0)}$. It seems natural to try showing that the $\boldsymbol{w}^{(t)}$ stays close to the linearized update, $\boldsymbol{L}^t\boldsymbol{w}^{(0)}$. However, the error in this step turns out to be difficult to control unless we choose the parameter ϵ to be really small (e.g., $\frac{1}{n^2}$). In our analysis, we take a different route that lets us analyze a wider parameter range.

The key will be to look at the difference between $\boldsymbol{w}^{(t)}$ and its projection to the top-$(K-1)$ eigenspace of the matrix $\boldsymbol{L}$. Recall that Π is the projection matrix onto this space. As above, define

$$\boldsymbol{z}^{(t)} := \Pi\boldsymbol{w}^{(t)} \text{ for all } t. \tag{11}$$

Lemma 3.7. *At every iteration t of Algorithm 1, we have:*

$$\|\boldsymbol{w}^{(t+1)} - \boldsymbol{z}^{(t+1)}\| \leq (1 + \eta\frac{c}{\sqrt{n^{\rho+2}}})\|\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}\| + \eta(\epsilon\Delta)^3.$$

Consequently, when the algorithm terminates, we have

$$\|\boldsymbol{w}^{(t_f)} - \boldsymbol{z}^{(t_f)}\| \leq \frac{4\|\boldsymbol{w}^{(t_f)}\|}{\Delta}.$$

Proof. We start with the update, Equation (9), and left-multiply both sides with $(\boldsymbol{I} - \Pi)$. Thus, we get

$$\begin{aligned} \boldsymbol{w}^{(t+1)} - \boldsymbol{z}^{(t+1)} &= (\boldsymbol{I} - \Pi)\boldsymbol{w}^{(t+1)} \\ &= (\boldsymbol{I} - \Pi)\boldsymbol{L}\boldsymbol{w}^{(t)} + \eta(\boldsymbol{I} - \Pi)\boldsymbol{E}^{(t)}\boldsymbol{w}^{(t)}. \end{aligned}$$

The second term on the RHS can be bounded using

$$\|(I - \Pi)\boldsymbol{E}^{(t)}\boldsymbol{w}^{(t)}\| \leq \|\boldsymbol{E}^{(t)}\boldsymbol{w}^{(t)}\|.$$

Then, using Lemma 3.4 and the fact that $\|\boldsymbol{w}^{(t)}\| \leq \epsilon\Delta$, we can bound this term by $\eta(\epsilon\Delta)^3$. For the first term, since $\boldsymbol{L}$ commutes with $(\boldsymbol{I} - \Pi)$, we have

$$\|\boldsymbol{L}(\boldsymbol{I} - \Pi)\boldsymbol{w}^{(t)}\| \leq (1 + \eta\frac{c}{\sqrt{n^{\rho+2}}})\|\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}\|.$$

This follows from the property of $\boldsymbol{L}$ from Lemma 3.5, that in the space orthogonal to the top $(K-1)$ eigenvectors, the spectral norm of $\boldsymbol{L}$ is $(1 + \eta\frac{c}{\sqrt{n^{\rho+2}}})$. Combining the two observations, the desired inequality follows.

To see the second part of the lemma, let us introduce some notation: write $\xi_t = \|\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}\|$, $\theta = \eta\frac{c}{\sqrt{n^{\rho+2}}}$, and $\delta = \eta(\epsilon\Delta)^3$. So the bound above can be written as

$$\xi_{t+1} \leq (1 + \theta)\xi_t + \delta.$$

Expanding, we have

$$\xi_{t_f} \leq (1 + \theta)^{t_f}\xi_0 + \delta\left[1 + (1 + \theta) + \cdots + (1 + \theta)^{t_f - 1}\right]$$
$$\leq (1 + \theta)^{t_f}(\xi_0 + \delta t_f).$$

From the bound on t_f from Lemma 3.6 and since n is large (so $\sqrt{n} \gg \log n$), we have that $(1 + \theta)^{t_f} \leq 2$, and from our setting of parameters ϵ, Δ, we have $\delta t_f < \epsilon$ and $\xi_0 \leq \epsilon$. This implies that $\xi_{t_f} \leq 4\epsilon$.

Since $\|\boldsymbol{w}^{(t_f)}\| \geq \epsilon\Delta$ at the end of the algorithm, the desired bound follows. $\qquad\square$

3.3. Convergence Analysis

As the final step, we show that the solution obtained at the end of the algorithm satisfies the desired cluster structure. We use Lemma 3.7 to primarily argue about the vector $\boldsymbol{z}^{(t_f)}$.

The first result says that the obtained x-embedding is well clustered.

Theorem 3.8 (Clustering property). *Suppose G is drawn from a symmetric SBM with K communities and suppose the co-occurrence matrix C used for constructing the embeddings satisfies Assumption 3.2. Suppose x is the x-embedding obtained by Algorithm 1 (i.e., the first n coordinates of $\boldsymbol{w}^{(t)}$). Let $\boldsymbol{\mu}$ be the "vector of means", i.e., for any index j in (ground truth) cluster $\mathbb{V}_i$, the entry μ_j is equal to $\mu_{\mathbb{V}_i} := \frac{\sum_{r \in \mathbb{V}_i} x_r}{|\mathbb{V}_i|}$. Then we have:*

$$\|\boldsymbol{x} - \boldsymbol{\mu}\| \leq \frac{5\|\boldsymbol{x}\|}{\Delta}.$$

Remark. Since Δ is large (grows as $n^{1/6}$), this implies that the variation in the embedding values *within clusters* is small. Quantitatively, we expect all the entries of $\boldsymbol{x}$ to be around $\frac{\|\boldsymbol{x}\|}{\sqrt{n}}$ in magnitude. Suppose they are in the interval $[-2\frac{\|\boldsymbol{x}\|}{\sqrt{n}}, 2\frac{\|\boldsymbol{x}\|}{\sqrt{n}}]$. The theorem says that the distance of a typical $\boldsymbol{x}$ to the cluster center is only about $\frac{\|\boldsymbol{x}\|}{\Delta\sqrt{n}}$. The catch with this theorem, however, is that it does not imply that there is a separation between the embedding values *across* clusters. This will be the subject of Theorem 3.10. But first, we prove Theorem 3.8

Proof. The outline of the proof is as follows: first we argue that $\boldsymbol{x} - \boldsymbol{\mu}$ is, in fact, precisely the projection of $\boldsymbol{x}$ to the space orthogonal to the span of the top K eigenvectors of the

matrix $\overline{C}$ (the "ideal" co-occurrence matrix). This implies that

$$\|\boldsymbol{x} - \boldsymbol{\mu}\|^2 \leq \|(I - \overline{\Pi})\boldsymbol{w}^{(t_f)}\|^2,$$

where $\overline{\Pi}$ is a projection onto the top $(K - 1)$ subspace of $\overline{L}$. Noting that the difference between Π and $\overline{\Pi}$ is small (by eigenspace perturbation theorems), we can then apply Lemma 3.7 to conclude the argument. The details of the proof are deferred to Appendix C.1 $\qquad\square$

Next, we wish to prove a cluster "gap" property: in other words, for the obtained embedding $\boldsymbol{x}$, $|\mu_{\mathbb{V}_i} - \mu_{\mathbb{V}_j}|$ is large enough. To show this, we proceed by observing that during our iterations, the projection $\boldsymbol{z}^{(t)} = \Pi\boldsymbol{w}^{(t)}$ gets updated in a very simple manner. This property is only true in the case of the *symmetric* SBM.

Lemma 3.9. *Let $\boldsymbol{z}^{(t)}$ be defined as in (11), and suppose we start with $\|\boldsymbol{z}^{(0)}\| \geq \frac{\epsilon}{10\sqrt{n}}$. Then for all $t \leq t_f = O\left(\frac{\log(n/\Delta)}{\eta}\right)$, we have*

$$\boldsymbol{z}^{(t)} = (1 + \eta\theta)^t\boldsymbol{z}^{(0)} + \boldsymbol{err}^{(t)},$$

where $\theta = \frac{n(a-b)}{K}$ and $\|\boldsymbol{err}^{(t)}\| \leq O(\log\frac{n}{\Delta})\frac{c}{\sqrt{n^{\rho+2}}}\|\boldsymbol{z}^{(t)}\|$.

The lemma is a strong structural statement, saying that $\boldsymbol{z}$ essentially only gets scaled in each iteration!

Proof. By multiplying the update, Equation (9), by Π, we see that

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{L}\boldsymbol{z}^{(t)} - \eta\Pi\boldsymbol{E}^{(t)}\boldsymbol{w}^{(t)}.$$

As we have done in the proof of Lemma 3.6, the norm of the second term can be bounded by $4\eta(\epsilon\Delta)^3 < \frac{\|\boldsymbol{z}^{(t)}\|}{\sqrt{n}}$. Now for the first term, observe that the top $(K - 1)$ eigenvalues of Π are all in the range $(1 + \eta\theta - \eta\frac{c}{\sqrt{n^{\rho+2}}}, 1 + \eta\theta + \eta\frac{c}{\sqrt{n^{\rho+2}}})$. Thus $\boldsymbol{L}$ acts as a scaling of the identity (within the space of the top $(K - 1)$ eigenvectors), up to an additive error $\eta\frac{c}{\sqrt{n^{\rho+2}}}$. This implies that

$$\boldsymbol{z}^{(t+1)} = (1 + \eta\theta)\boldsymbol{z}^{(t)} + \eta\frac{2c}{\sqrt{n^{\rho+2}}}\xi^{(t)},$$

for some error vector $\xi^{(t)}$ with $\|\xi^{(t)}\| \leq \|\boldsymbol{z}^{(t)}\|$. Once we have this for all t, we can sum up to obtain

$$\boldsymbol{z}^{(t)} = (1 + \eta\theta)^t\boldsymbol{z}^{(0)} + \frac{2c\eta}{\sqrt{n^{\rho+2}}} \cdot \boldsymbol{err}, \qquad (12)$$

where

$$\boldsymbol{err} = \left[(1 + \eta\theta)^{t-1}\xi^{(0)} + (1 + \eta\theta)^{t-2}\xi^{(1)} + \cdots\right].$$

Now we can inductively maintain the property that $\|\boldsymbol{z}^{(j)}\| \leq 2(1 + \eta\theta)^j\|\boldsymbol{z}^{(0)}\|$, to conclude that

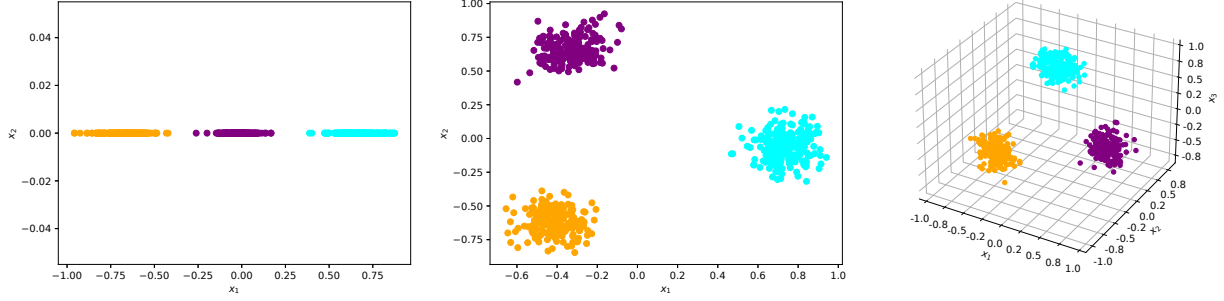$$\|\boldsymbol{err}\| \leq t \cdot (1 + \eta\theta)^t\|\boldsymbol{z}^{(0)}\|.$$

Figure 1. For a random graph drawn from a stochastic block model with $K = 3$ blocks and $n = 600$, we show the calculated (left) 1-dimensional (middle) 2-dimensional and (right) 3-dimensional embeddings.

Noting that $t \leq \frac{\log(n/\Delta)}{\eta}$, we can plug this into Equation (12) to complete the proof of the lemma. □

This structural theorem is very powerful. To see it, suppose $\boldsymbol{u}$ is any vector in $\mathbb{R}^{2n}$, and suppose we originally had

$$|\langle \boldsymbol{u}, \boldsymbol{z}^{(0)} \rangle| \geq \beta \|\boldsymbol{z}^{(0)}\|.$$

Then, we can use Lemma 3.9 to conclude that

$$|\langle \boldsymbol{u}, \boldsymbol{z}^{(t_f)} \rangle| \geq \beta \|\boldsymbol{z}^{(t_f)}\| - \frac{c \log(n/\Delta)}{\sqrt{n^{\rho+2}}} \|\boldsymbol{z}^{(t_f)}\|.$$

Thus, if $\beta > \frac{4c \log \frac{n}{\Delta}}{\sqrt{n^{\rho+2}}}$, we can conclude that

$$|\langle \boldsymbol{u}, \boldsymbol{z}^{(t_f)} \rangle| \geq \frac{\beta}{2} \|\boldsymbol{z}^{(t_f)}\|.$$

This leads us to the following result.

Theorem 3.10. *Suppose G is drawn from a symmetric SBM with K communities and suppose the co-occurrence matrix $\boldsymbol{C}$ used for constructing the embeddings satisfies Assumption 3.2. Let $\boldsymbol{x}$ be the (x component of the) embedding obtained by Algorithm 1. Then with probability ≥ 0.9 over the initialization, the embedding satisfies:*

$$\forall \text{ clusters } i \neq j, \ |\mu_{\mathbb{V}_i} - \mu_{\mathbb{V}_j}| \geq \frac{\epsilon \Delta}{20 K^2 \sqrt{n}}.$$

(As before, $\mu_{\mathbb{V}_i} := \frac{\sum_{r \in \mathbb{V}_i} x_r}{|\mathbb{V}_i|}$.)

Remark. It is natural to ask if this separation is sufficient. For this, we do the following heuristic calculation. In the end, we have $\|x\| = \epsilon \Delta$. This means all the coordinates are roughly of magnitude $\frac{\epsilon \Delta}{\sqrt{n}}$ (say they are in the range $[-2 \frac{\epsilon \Delta}{\sqrt{n}}, 2 \frac{\epsilon \Delta}{\sqrt{n}}]$. The theorem says that the typical gap between the cluster centers is a constant factor of this interval length (assuming K is a constant), with high probability.

The main idea of the proof is to show that a sufficient amount of "initial separation" between cluster means must exist because the initialization is random. Then, the discussion preceding the theorem can be used to show that this separation must persist as we update. The key point is that as a fraction of the length $\boldsymbol{z}^{(0)}$, the initial separation is significant. The proof details are deferred to Appendix C.2.

Theorems 3.8 and 3.10 lead to the following weak recovery result.

Theorem 3.11. *Suppose G is drawn from a symmetric SBM with K communities and suppose the co-occurrence matrix $\boldsymbol{C}$ used for constructing the embeddings satisfies Assumption 3.2. Suppose we run Algorithm 1 for $t_f < \frac{4 \log(n/\Delta)}{\eta}$ iterations. Then with high probability, we recover $1 - o(1)$ fraction of each community.*

Proof. From Theorem 3.8, after t_f iterations, we have

$$\|\boldsymbol{x} - \mu\|^2 = \sum_{i \in [K]} \sum_{j \in V_i} |x_j - \mu_i|^2 \leq \frac{c_1 \|\boldsymbol{x}\|^2}{\Delta^2}.$$

for a positive constant c_1. Furthermore, from Theorem 3.10 we have for all pairs of clusters $i \neq j$,

$$|\mu_i - \mu_j|^2 \geq \frac{(\epsilon \Delta)^2}{c_2 K^4 n}$$

for a positive constant c_2.

The result follows from a simple application of Markov's inequality:

$$\Pr\left(|x_j - \mu_i|^2 \geq \frac{(\epsilon \Delta)^2}{4 c_2 K^4 n}\right) \leq \frac{4 c_2 K^4 n \mathbb{E}[|x_j - \mu_i|^2]}{(\epsilon \Delta)^2}. \tag{13}$$

Each cluster has size n/K. This, along with Theorem 3.8, implies that within a cluster V_i we have $\mathbb{E}[|x_j - \mu_i|^2] \leq \frac{K}{n} \frac{2 c_1 (\epsilon \Delta)^2}{n^{1/3}}$. (Recall that the algorithm terminates after t_f
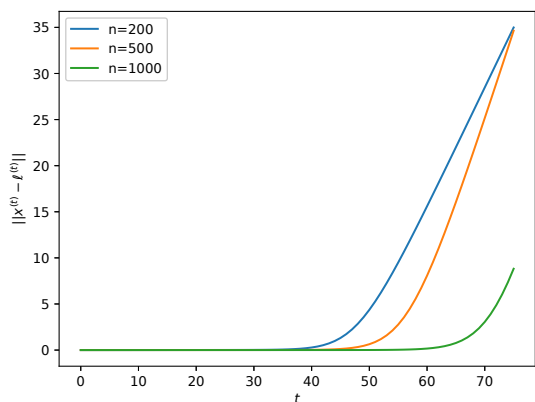
Figure 2. For random graphs drawn from a stochastic block model with $K = 2$ blocks and $n = 200, 500, 1000$ nodes, we track the distance between the original nonlinear gradient descent update for deepwalk and its linear approximation.

iterations when $\epsilon\Delta \leq \|w^{t_f}\| \leq 2\epsilon\Delta$.) Plugging this into Equation (13) we have

$$\Pr\left(|x_j - \mu_i|^2 \geq \frac{(\epsilon\Delta)^2}{4c_2 K^4 n}\right) \leq \frac{8c_1 c_2 K^5}{n^{1/3}}. \qquad (14)$$

This implies that the number of vertices in a cluster V_i that are not within a distance of $\frac{1}{2} \cdot \frac{\epsilon\Delta}{(c_2 K^4 n)^{1/2}}$ of their cluster mean (or half the distance between cluster means) is no greater than $\frac{8c_1 c_2 K^5}{n^{1/3}}$. (We're also assuming that the number of clusters K is constant.) This concludes the proof. $\square$

4. Experiments

This section presents experimental results that support our theoretical analysis. Section 4.1 shows that DeepWalk embeddings can completely recover the community structure in a graph even when the embedding dimension is lower than the number of communities. Section 4.2 shows that embeddings trained using a linear approximation to the update equation remains close to embeddings trained using the nonlinear update.

4.1. Calculated Embeddings

This section presents 1-dimensional, 2-dimensional, and 3-dimensional embeddings produced by training the deepwalk algorithm. A graph was drawn from a stochastic block model with $K = 3$ communities, $n = 600$ nodes and parameters $q = 0.1$ and $p = 4q$. We run the algorithm for $T = 100$ iterations and used a learning rate of $\eta = 0.01$. The embeddings are initialized randomly so that $\|x^{(t)}\|_\infty \leq 0.01$ and $\|y^{(t)}\|_\infty \leq 0.01$.

All three sets of embeddings completely recover the community structure in the graph (see Figure 1). As our analysis shows, even 1-dimensional embeddings show a clear separation between the clusters.

4.2. Linear Approximation to the Update

We note in our analysis that the linear part of the update dominates the convergence behavior, suggesting that the update equations can be approximated by a linear function. This section provides experimental results that supports this observation even for reasonably large graphs.

Three random graphs are drawn from a stochastic block model with $K = 2$ clusters with $n = 200, 500, 1000$ nodes and parameters $q = 0.1$ and $p = 4q$. On each graph, embeddings are calculated using both the original nonlinear gradient descent update equations and the linear approximation to the update equation. The distance $d(x^{(t)}, \ell^{(t)}) = \|x^{(t)} - \ell^{(t)}\|$ between an embedding trained with a nonlinear update $x^{(t)}$ and an embedding trained with a linear update $\ell^{(t)}$ is tracked throughout the training procedure (see Figure 2). The embeddings were initialized randomly so that $\|x^{(0)}\|_\infty = \|\ell^{(0)}\|_\infty \leq \frac{1}{\sqrt{n}}$ and $w^{(0)} = \ell^{(0)}$. The learning rate was set for $\eta = \frac{1}{n}$ and training was rate for $T = 75$ iterations.

The distance $d(x^{(t)}, w^{(t)})$ remains nearly zero for the first several iterations, allowing ample time for the clusters to separate, despite the graphs being a reasonable size. The results also suggests that the linear approximation may hold for a larger parameter regime than the ones we use in our analysis. Determining the right trade-offs between the radius of initialization ϵ, the learning rate η, and the SBM parameters K, p, q, is an interesting open question.

5. Conclusion

We give the first provable convergence guarantees, that we are aware of, for the DeepWalk algorithm and obtaining low-dimensional embeddings of vertices of a graph. We show that for a graph drawn from a stochastic block model (SBM), the DeepWalk embeddings obtained by a random initialization in a small enough ball and trained with gradient descent, recovers the community structure with high probability. Unlike previous works, we analyze the gradient descent dynamics directly; the key technical tool is to show that when embeddings are of small enough length, the gradient descent update can be approximated by a linear update up to a small error. Our approximation technique may be applicable to other related graph embedding problems, which gives an interesting avenue for future work.

Acknowledgements

Impact Statement

The paper's contributions are primarily theoretical, aiming to help obtain a better understanding of a popular technique used in Social Network analysis. There are many potential societal consequences, none of which we feel must be specifically highlighted here.

References

Abbe, E. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Alon, N. Eigenvalues and expanders. *Combinatorica*, 6(2): 83–96, June 1986.

Arora, S., Rao, S., and Vazirani, U. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56 (2), apr 2009.

Arora, S., Hu, W., and Kothari, P. K. An analysis of the t-sne algorithm for data visualization. In *Conference On Learning Theory*, pp. 1455–1462. PMLR, 2018.

Backstrom, L. and Leskovec, J. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pp. 635–644, New York, NY, USA, 2011.

Barot, A., Bhamidi, S., and Dhara, S. Community detection using low-dimensional network embedding algorithms. *arXiv preprint arXiv:2111.05267*, 2021.

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 585–591, Cambridge, MA, USA, 2001.

Cai, T. T. and Ma, R. Theoretical foundations of t-sne for visualizing high-dimensional clustered data, 2022.

Chanpuriya, S. and Musco, C. Infinitewalk: Deep network embeddings as laplacian embeddings with a nonlinearity. *CoRR*, abs/2006.00094, 2020.

Davison, A., Morgan, S. C., and Ward, O. G. Community detection and classification guarantees using embeddings learned by node2vec. *arXiv preprint arXiv:2310.17712*, 2023.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Hall, K. M. An r-dimensional quadratic placement algorithm. *Manage. Sci.*, 17(3):219–229, nov 1970.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Harker, C. and Bhaskara, A. Structure of nonlinear node embeddings in stochastic block models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6764–6782. PMLR, 25–27 Apr 2023.

Holland, P., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social Networks*, 5:109–137, 1983.

Kloepfer, D., Aviles-Rivero, A. I., and Heydecker, D. Delving into deep walkers: A convergence analysis of random-walk-based vertex embeddings. *arXiv preprint arXiv:2107.10014*, 2021.

Levy, O. and Goldberg, Y. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Linderman, G. C. and Steinerberger, S. Clustering with t-sne, provably. *SIAM Journal on Mathematics of Data Science*, 1(2):313–332, 2019.

Linial, N., London, E., and Rabinovich, Y. The geometry of graphs and some of its algorithmic applications. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 577–591, 1994.

McSherry, F. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 529–537, 2001.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b.

Ng, A., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. 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*, pp. 459–467, 2018.

Qiu, J., Wang, C., Liao, B., Peng, R., and Tang, J. A matrix chernoff bound for markov chains and its application to co-occurrence matrices. *Advances in Neural Information Processing Systems*, 33:18421–18432, 2020.

Rohe, K., Chatterjee, S., and Yu, B. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

Satpathi, S. and Srikant, R. The dynamics of gradient descent for overparametrized neural networks. In *Learning for Dynamics and Control*, pp. 373–384. PMLR, 2021.

Sinclair, A. and Jerrum, M. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82(1):93–133, 1989.

Spielman, D. A. and Teng, S.-H. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2):284–305, 2007.

Stewart, G. and Sun, J. *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Elsevier Science, 1990.

Stöger, D. and Soltanolkotabi, M. Small random initialization is akin to spectral learning: Optimization and generalization guarantees for overparameterized low-rank matrix reconstruction. *Advances in Neural Information Processing Systems*, 34:23831–23843, 2021.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pp. 1067–1077, Republic and Canton of Geneva, CHE, 2015.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Vu, V. H. Spectral norm of random matrices. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pp. 423–430, New York, NY, USA, 2005.

Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., and Yang, S. Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 203–209. AAAI Press, 2017.

Zhang, Y. and Tang, M. Consistency of random-walk based network embedding algorithms. *arXiv preprint arXiv:2101.07354*, 2021.

A. Proofs From Section 3.1

A.1. Properties of Co-occurrence Matrix

Lemma A.1. *Suppose $\overline{A}$ is the expected adjacency matrix of a graph G drawn from an SBM with K communities and parameters p, q as described in Section 2.3. Also, suppose that $\overline{C}$ is the co-occurrence matrix constructed as in Lemma 2.1 from the expected adjacency matrix $\overline{A}$ with random walks of length L and a window size of T such that $L > T$. Let $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n$ be the eigenvalues of $\overline{A}$ and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ be the eigenvalues of $\overline{C}$. Then the matrix $\overline{C}$ has the following properties:*

1. *$\overline{C}$ has a block structure. In other words, it has the form*

$$\overline{C} = \underbrace{\begin{bmatrix} a & b & ... & b \\ b & a & ... & b \\ \vdots & \vdots & \ddots & \vdots \\ b & b & ... & a \end{bmatrix}}_{K \times K} \otimes J_{\frac{n}{K} \times \frac{n}{K}}$$

   for some parameters a, b.

2. *The eigenvalues of $\overline{C}$ are*

$$\lambda_i = \begin{cases} \frac{2}{n}\left(TL - \frac{T(T+1)}{2}\right) & \text{if } i = 1, \\ \frac{2}{n}\sum_{t=1}^{T}(L - t)\left(\frac{\alpha_i}{\alpha_1}\right)^t & \text{if } i = 2, \ldots, K, \\ 0 & \text{if } i = K+1, \ldots, n, \end{cases}$$

   where $\alpha_1 = \frac{n}{K}p + \frac{n(K-1)}{K}q$ and $\alpha_i = \frac{n}{K}(p - q)$ for $i = 2, \ldots, K$.

3. *The difference $a - b$ between the on-diagonal block and off-diagonal block entries a and b of $\overline{C}$ is*

$$a - b = \frac{2K}{n^2}\sum_{t=1}^{T}(L - t)\left(\frac{p - q}{p + (K - 1)q}\right)^t.$$

4. *The on-diagonal block entries a and off-diagonal block entries b of $\overline{C}$ are:*

$$a = \frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) + (K - 1)\sum_{t=1}^{T}(L - t)\left(\frac{p - q}{p + (K - 1)q)}\right)^t\right),$$

$$b = \frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) - \sum_{t=1}^{T}(L - t)\left(\frac{p - q}{p + (K - 1)q}\right)^t\right).$$

Proof. (*Proof of 1.*) Straightforward from the block structure of $\overline{A}$.

(*Proof of 2.*) From Lemma 2.1, we can write $\overline{C}$ as

$$\overline{C} = 2\sum_{t=1}^{T}\frac{(L - t)}{n\overline{d}}D_{\overline{A}}\left(\overline{P}^t\right) \tag{15}$$

where $\overline{d} = \frac{n}{K}p + \frac{n(K-1)}{K}q$ and $\overline{P} = D_{\overline{A}}^{-1}\overline{A}$. Since $D_{\overline{A}}^{-1}$ is a diagonal matrix with diagonal entries $\frac{1}{\overline{d}}$, the matrix $\overline{P}$ is a real-symmetric matrix and we can write it as

$$\overline{P} = V\Sigma V^\top,$$

where the columns of V are the eigenvectors of $\overline{P}$ and $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ are the eigenvalues of $\overline{P}$.

Recall that the eigenvalues of $\overline{A}$ are

$$
\alpha_i = \begin{cases} \frac{n}{K}p + \frac{n(K-1)}{K}q & \text{if } i = 1, \\ \frac{n(p-q)}{K} & \text{if } i = 2, \ldots, K, \\ 0 & \text{if } i = K+1, \ldots, n. \end{cases}
$$

Therefore, since $\overline{d} = \alpha_1$, the eigenvalues of $\overline{P}$ are:

$$
\sigma_i = \begin{cases} 1 & \text{if } i = 1, \\ \frac{\alpha_i}{\alpha_1} & \text{if } i = 2, \ldots, K \\ 0 & \text{if } i = K+1, \ldots, n. \end{cases} \tag{16}
$$

We can now write $\overline{C}$ as

$$
\begin{aligned}
\overline{C} &= \frac{2}{n} \sum_{t=1}^{T} (L - t) \left(V \Sigma^t V^\top \right) \\
&= V \left(\frac{2}{n} \sum_{t=1}^{T} (L - t) \cdot \Sigma^t \right) V^\top \\
&= V \Lambda V^\top
\end{aligned}
$$

where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ with $\lambda_i = \frac{2}{n} \sum_{t=1}^{T} (L - t) \sigma_i^t$ for all $i = 1, ..., n$. Therefore, the eigenvalues of $\overline{C}$ are

$$
\lambda_i = \begin{cases} \frac{2}{n} \left(TL - \frac{T(T+1)}{2} \right) & \text{if } i = 1, \\ \frac{2}{n} \sum_{t=1}^{T} (L - t) \left(\frac{\alpha_i}{\alpha_1} \right)^t & \text{if } i = 2, \ldots, K, \\ 0 & \text{if } i = K+1, \ldots, n, \end{cases} \tag{17}
$$

where we used $\sum_{t=1}^{T} (L - t) = TL - \frac{T(T+1)}{2}$.

(*Proof of 3.*) First, notice that we can express the eigenvalues of $\overline{C}$ in terms of its on-diagonal entries a and off-diagonal entries b:

$$
\lambda_i = \begin{cases} \frac{n}{K}a + \frac{n(K-1)}{K}b & \text{if } i = 1, \\ \frac{n(a-b)}{K} & \text{if } i = 2, \ldots, K, \\ 0 & \text{if } i = K+1, \ldots, n. \end{cases} \tag{18}
$$

From equations (17) and (18), we know that for $i = 2, \ldots, K$ we have

$$
\frac{n}{K}(a - b) = \frac{2}{n} \sum_{t=1}^{T} (L - t) \left(\frac{p - q}{p + (K-1)q} \right)^t.
$$

By multiplying both sides by $\frac{K}{n}$, the result immediately follows.

(*Proof of 4.*) We prove the result for b first. From Equation (18) we know that

$$\lambda_1 - \lambda_2 = \frac{n}{K}a + \frac{n(K-1)}{K}q - \frac{n}{K}(a-b) = nb. \tag{19}$$

Meanwhile, from Equation (17) we know that

$$\lambda_1 - \lambda_2 = \frac{2}{n}\left(TL - \frac{T(T+1)}{2}\right) - \frac{2}{n}\sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q}\right)^t. \tag{20}$$

Combining equations (19) and (20) and solving for b gives

$$b = \frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) - \sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q}\right)^t\right). \tag{21}$$

Now, we find a. From Equation (21) we have

$$a - b = a - \left(\frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) - \sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q}\right)^t\right)\right). \tag{22}$$

From Property 3, we have

$$a - b = \frac{2K}{n^2}\sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q}\right)^t. \tag{23}$$

Combining equations (22) and (23) and solving for a gives:

$$a = \frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) + (K-1)\sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q}\right)^t\right). \tag{24}$$

This completes the proof.

$\square$

A.2. Proof of Lemma 3.1

First, we prove the following lemma:

Lemma A.2. *Suppose $\boldsymbol{A}$ is the adjacency matrix of a graph G that is drawn from a symmetric SBM with K equally sized communities and parameters p, q of at least $n^{\rho-1}$ with $\rho \in (0, 1)$. Let $\boldsymbol{P} = \boldsymbol{D}_{\boldsymbol{A}}^{-1}\boldsymbol{A}$ be the transition matrix of the graph. Similarly, let $\overline{\boldsymbol{P}} = \boldsymbol{D}_{\overline{\boldsymbol{A}}}^{-1}\overline{\boldsymbol{A}}$ be the transition matrix of the expected graph and $\overline{d} = \frac{n}{K}p + \frac{n(K-1)}{K}q$ be the expected degree. Then for any integer $t \geq 1$, we have*

$$\left\|\boldsymbol{P}^t - \overline{\boldsymbol{P}}^t\right\| = O\left(\sqrt{\frac{\log n}{\overline{d}}}\right)$$

with high probability.

Proof. We prove this by induction. First, we prove the base case. Let $t = 1$. Then

$$
\begin{aligned}
\left\| \boldsymbol{P} - \overline{\boldsymbol{P}} \right\| &= \left\| \boldsymbol{D}_{\boldsymbol{A}}^{-1} \boldsymbol{A} - \boldsymbol{D}_{\overline{\boldsymbol{A}}}^{-1} \overline{\boldsymbol{A}} \right\| \\
&= \left\| \boldsymbol{D}_{\boldsymbol{A}}^{-1} \left(\boldsymbol{A} - \overline{\boldsymbol{A}} \right) + \left(\boldsymbol{D}_{\boldsymbol{A}}^{-1} - \boldsymbol{D}_{\overline{\boldsymbol{A}}}^{-1} \right) \overline{\boldsymbol{A}} \right\| \\
&\leq \underbrace{\left\| \boldsymbol{D}_{\boldsymbol{A}}^{-1} \left(\boldsymbol{A} - \overline{\boldsymbol{A}} \right) \right\|}_{(i)} + \underbrace{\left\| \left(\boldsymbol{D}_{\boldsymbol{A}}^{-1} - \boldsymbol{D}_{\overline{\boldsymbol{A}}}^{-1} \right) \overline{\boldsymbol{A}} \right\|}_{(ii)}.
\end{aligned}
$$

Given a graph G drawn from a symmetric SBM with parameters K, p, q with $p \geq n^{\rho-1}$ for $\rho \in (0, 1)$, we have

$$
\left| d_i - \overline{d} \right| = O \left(\sqrt{\overline{d} \log n} \right), \forall i \in [n], \tag{25}
$$

with probability $1 - n^{-5}$. This follows from standard Bernstein/Chernoff bounds (Vershynin, 2018). Also, with probably $1 - n^{-5}$, we have

$$
\left\| \boldsymbol{A} - \overline{\boldsymbol{A}} \right\| = O \left(\sqrt{\overline{d}} \right), \tag{26}
$$

which follows from standard spectral bounds (Vu, 2005). Therefore, we can bound (i) by

$$
\left\| \boldsymbol{D}_{\boldsymbol{A}}^{-1} \left(\boldsymbol{A} - \overline{\boldsymbol{A}} \right) \right\| \leq \frac{2}{\overline{d}} \cdot O \left(\sqrt{\overline{d}} \right) = O \left(\frac{1}{\sqrt{\overline{d}}} \right).
$$

Similarly, we can bound (ii) by

$$
\begin{aligned}
\left\| \left(\boldsymbol{D}_{\boldsymbol{A}}^{-1} - \boldsymbol{D}_{\overline{\boldsymbol{A}}}^{-1} \right) \overline{\boldsymbol{A}} \right\| &\leq \max_i \left| \frac{d_i - \overline{d}}{d_i \overline{d}} \right| \cdot \left\| \overline{\boldsymbol{A}} \right\| \\
&= O \left(\overline{d}^{-3/2} \sqrt{\log n} \right) \cdot \overline{d} \\
&= O \left(\sqrt{\frac{\log n}{\overline{d}}} \right).
\end{aligned}
$$

Combining the bounds of (i) and (ii), we have

$$
\left\| \boldsymbol{P} - \overline{\boldsymbol{P}} \right\| = O \left(\sqrt{\frac{\log n}{\overline{d}}} \right).
$$

Now, assume that the statement holds for $t - 1$, i.e.,

$$
\left\| \boldsymbol{P}^{t-1} - \overline{\boldsymbol{P}}^{t-1} \right\| = O \left(\sqrt{\frac{\log n}{\overline{d}}} \right). \tag{27}
$$

Therefore, by induction we have

$$
\begin{aligned}
\left\| \boldsymbol{P}^t - \overline{\boldsymbol{P}}^t \right\| &\leq \left\| \boldsymbol{P}^{t-1} \left(\boldsymbol{P} - \overline{\boldsymbol{P}} \right) \right\| + \left\| \left(\boldsymbol{P}^{t-1} - \overline{\boldsymbol{P}}^{t-1} \right) \overline{\boldsymbol{P}} \right\| \\
&= O \left(\sqrt{\frac{\log n}{\overline{d}}} \right).
\end{aligned}
$$

This concludes the proof.

$\square$

Now, we precede to prove Lemma 3.1. Using Lemma 2.1, we can express the co-occurrence matrix C as

$$C = 2\sum_{t=1}^{T} \frac{(L-t)}{2|E|} D_A P^t,$$

and a corresponding expected co-occurrence matrix $\overline{C}$ as

$$\overline{C} = 2\sum_{t=1}^{T} \frac{(L-t)}{n\overline{d}} D_{\overline{A}} \overline{P}^t,$$

where $P = D_A^{-1}A$ and $\overline{P} = D_{\overline{A}}^{-1}\overline{A}$. From Property 4 of Lemma A.1, we know that $\overline{C}$ has a block structure with entries a and b equal to

$$a = \frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) + (K-1)\sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q)}\right)^t \right),$$

$$b = \frac{2}{n^2}\left(\left(TL - \frac{T(T+1)}{2}\right) - \sum_{t=1}^{T}(L-t)\left(\frac{p-q}{p+(K-1)q}\right)^t \right).$$

Now, we can express the difference $C - \overline{C}$ as

$$C - \overline{C} = 2\sum_{t=1}^{T}(L-t)\left(\frac{1}{2|E|}D_A P^t - \frac{1}{n\overline{d}}D_{\overline{A}}\overline{P}^t\right) \tag{28}$$

We proceed by showing that each term in the summation has a spectral norm $\leq \frac{\sqrt{\log n}}{n\sqrt{\overline{d}}}$. Notice that we can write each term in (28) as

$$\frac{1}{2|E|}D_A P^t - \frac{1}{n\overline{d}}D_{\overline{A}}\overline{P}^t = \frac{1}{2|E|}D_A P^t - \frac{1}{n\overline{d}}D_A P^t + \frac{1}{n\overline{d}}D_A P^t - \frac{1}{n\overline{d}}D_{\overline{A}}\overline{P}^t$$

$$= \underbrace{\left(\frac{n\overline{d} - 2|E|}{2|E|n\overline{d}}\right)D_A P^t}_{(i)} + \underbrace{\frac{1}{n\overline{d}}\left(D_A P^t - D_{\overline{A}}\overline{P}^t\right)}_{(ii)}. \tag{29}$$

We bound (i) and (ii) in (29) separately. Given a graph G drawn from a symmetric SBM with parameters K, p, q with $p \geq n^{\rho-1}$ for $\rho \in (0,1)$, we have

$$|d_i - \overline{d}| = O\left(\sqrt{\overline{d}\log n}\right), \forall i \in [n], \tag{30}$$

$$\left||E| - \frac{n\overline{d}}{2}\right| = O\left(\sqrt{n\overline{d}\log n}\right) \tag{31}$$

with probability $1 - n^{-5}$. These follow from standard Bernstein/Chernoff bounds (Vershynin, 2018). Therefore, with high probability, the norm of the first term (i) in (29) can be bounded:

$$\left\| \left(\frac{n\overline{d} - 2|E|}{2|E|n\overline{d}} \right) \boldsymbol{D_A} \boldsymbol{P}^t \right\| \leq \left| \frac{n\overline{d} - 2|E|}{2|E|n\overline{d}} \right| \cdot \max_i d_i \cdot \left\| \boldsymbol{P}^t \right\|$$

$$\leq O\left((n\overline{d})^{-3/2} \sqrt{\log n} \right) \cdot \left(\overline{d} + O\left(\sqrt{\overline{d} \log n} \right) \right) \cdot (1)$$

$$= O\left(\frac{1}{n\sqrt{\overline{d}}} \right).$$

Next, we bound the second term (ii) in (29). We can write (ii) as

$$\frac{1}{n\overline{d}} \left(\boldsymbol{D_A} \boldsymbol{P}^t - \boldsymbol{D}_{\overline{mA}} \overline{\boldsymbol{P}}^t \right) = \underbrace{\frac{1}{n\overline{d}} \left(\boldsymbol{D_A} - \boldsymbol{D}_{\overline{A}} \right) \boldsymbol{P}^t}_{(iii)} + \underbrace{\frac{1}{n\overline{d}} \boldsymbol{D}_{\overline{A}} \left(\boldsymbol{P}^t - \overline{\boldsymbol{P}}^t \right)}_{(iv)}. \tag{32}$$

Using (30), the norm of (iii) in (32) can be bounded with high probability:

$$\left\| \frac{1}{n\overline{d}} \left(\boldsymbol{D_A} - \boldsymbol{D}_{\overline{A}} \right) \boldsymbol{P}^t \right\| \leq \frac{1}{n\overline{d}} \left\| \boldsymbol{D_A} - \boldsymbol{D}_{\overline{A}} \right\| \cdot \left\| \boldsymbol{P}^t \right\|$$

$$= \frac{1}{n\overline{d}} \cdot O\left(\sqrt{\overline{d} \log n} \right)$$

$$= O\left(\frac{\sqrt{\log n}}{n\sqrt{\overline{d}}} \right).$$

Now, using Lemma A.2 and (30), we bound the norm of (iv) in (32):

$$\left\| \frac{1}{n\overline{d}} \boldsymbol{D}_{\overline{A}} \left(\boldsymbol{P}^t - \overline{\boldsymbol{P}}^t \right) \right\| \leq \frac{1}{n\overline{d}} \left\| \boldsymbol{D}_{\overline{A}} \right\| \cdot \left\| \boldsymbol{P}^t - \overline{\boldsymbol{P}}^t \right\|$$

$$= O\left(\frac{\sqrt{\log n}}{n\sqrt{\overline{d}}} \right).$$

Combining the bounds for (i), (iii), and (iv), gives

$$\left\| \boldsymbol{C} - \overline{\boldsymbol{C}} \right\| \leq 2 \sum_{t=1}^{T} (L - t) \left\| \frac{1}{2|E|} \boldsymbol{D_A} \boldsymbol{P}^t - \frac{1}{n\overline{d}} \boldsymbol{D}_{\overline{A}} \overline{\boldsymbol{P}}^t \right\|$$

$$= 2 \left(TL - \frac{T(T+1)}{2} \right) \cdot O\left(\frac{\sqrt{\log n}}{n\sqrt{\overline{d}}} \right)$$

$$= \left\| \overline{\boldsymbol{C}} \right\| \cdot O\left(\sqrt{\frac{\log n}{n^\rho}} \right).$$

This concludes the proof.

B. Proofs From Section 3.2

B.1. Proof of Proposition 3.3

Suppose that without loss of generality that $y_1 \leq y_2 \leq ... \leq y_n$. We have

$$\left\|Q - \frac{1}{n}J\right\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{e^{x_iy_j}}{\sum_{k=1}^{n}e^{x_iy_k}} - \frac{1}{n}\right)^2$$

$$= \sum_{x_i\leq 0}\sum_{j=1}^{n}\left(\frac{e^{x_i(y_j-y_n)}}{\sum_{k=1}^{n}e^{x_i(y_k-y_n)}} - \frac{1}{n}\right)^2 + \sum_{x_i>0}\sum_{j=1}^{n}\left(\frac{e^{x_i(y_j-y_1)}}{\sum_{k=1}^{n}e^{x_i(y_k-y_1)}} - \frac{1}{n}\right)^2$$

$$\leq \sum_{x_i\leq 0}\frac{n}{4}\left(\frac{e^{x_i(y_1-y_n)}-1}{\sum_{k=1}^{n}e^{x_i(y_k-y_n)}}\right)^2 + \sum_{x_i>0}\frac{n}{4}\left(\frac{e^{x_i(y_n-y_1)}-1}{\sum_{k=1}^{n}e^{x_i(y_k-y_1)}}\right)^2$$

$$\leq \frac{1}{4}\left(e^{2\epsilon^2}-1\right)^2$$

$$\leq \epsilon^4.$$

The first inequality is an application of Popoviciu's inequality (Theorem D.2) and in the last step we use the fact that $e^x - 1 \leq xe^x$ for $x > 0$ and $xe^x \leq 2x$ when $x < \log 2$. Taking the square root gives the result.

B.2. Proof of Lemma 3.4

Note that for a block matrix $Y = \begin{bmatrix} 0 & X \\ X^\top & 0 \end{bmatrix}$ we have $\|Y\|_F^2 = 2\|X\|_F^2$. This means that

$$\|E^{(t)}\|_F^2 = 2\left\|D_C\left(Q^{(t)} - \frac{J}{n}\right)\right\|_F^2$$

$$\leq 2\|D_C\|^2\left\|Q^{(t)} - \frac{1}{n}J\right\|_F^2$$

$$\leq 2\|D_C\|^2 \cdot \epsilon^4,$$

where we used proposition 3.3 in the last inequality.

Now we need to bound $\|D_C\|$. We have

$$\|D_C\| = \|C\|_\infty \leq \|C\| \leq \frac{n}{K}(a + (K-1)b) + \|R\| \leq 1 + \frac{c}{\sqrt{n^{\rho+2}}} \leq 2,$$

where we have used the fact that $\frac{n}{K}(a + (K-1)b) = O\left(\frac{1}{n}\right)$ (see Lemma A.1) and $\|\overline{C} - C\| = \|R\| \leq \frac{c}{\sqrt{n^{\rho+2}}}$. Therefore

$$\|E^{(t)}\|_F \leq \sqrt{8}\epsilon^2 < 4\epsilon^2.$$

This concludes the proof.

B.3. Proof of Lemma 3.5

First, note that for a real symmetric matrix X with eigenvalues $\sigma_1, ..., \sigma_n$, the matrix $Y = \begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix}$ has eigenvalues $\alpha_i = \pm\sigma_i$.

Let

$$\overline{L} = \begin{bmatrix} I & -\eta\overline{M} \\ -\eta\overline{M} & I \end{bmatrix}.$$

Recall from section 3.1, that $\overline{M}$ has $(K-1)$ nonzero eigenvalues of $\frac{n}{K}(b-a)$ and $(n-K+1)$ eigenvalues of 0. This implies that $\overline{L}$ has eigenvalues

$$\overline{\alpha}_i = \begin{cases} 1 + \eta\frac{n}{K}(a-b) & \text{if } i = 1, \ldots, K-1, \\ 1 & \text{if } i = K, \ldots, 2n-K, \\ 1 - \eta\frac{n}{K}(a-b) & \text{if } i = 2n-K+1, \ldots, 2n. \end{cases}$$

For an eigenvalue α_i of $\boldsymbol{L}$, Weyl's theorem (Theorem D.1) implies that for $i = 1, \ldots, K-1$, we have $|\alpha_i - \overline{\alpha}_i| \leq \|\boldsymbol{R}\|$, where $\boldsymbol{R} = \boldsymbol{L} - \overline{\boldsymbol{L}}$. This means that for $i = 1, \ldots, K-1$, we have

$$\alpha_i \geq 1 + \eta \frac{n}{K}(a-b) - \eta \frac{c}{\sqrt{n^{\rho+2}}} \geq 1 + \eta \frac{n(a-b)}{2K} = 1 + \eta \gamma.$$

For $i = K, \ldots, 2n - K$, we have

$$\alpha_i \leq 1 + \eta \frac{c}{\sqrt{n^{\rho+2}}},$$

and for $i = 2n - K + 1, \ldots, 2n$, we have

$$\alpha_i \leq 1 - \eta \frac{n}{K}(a-b) + \eta \frac{c}{\sqrt{n^{\rho+2}}} \leq 1 + \eta \frac{c}{\sqrt{n^{\rho+2}}}.$$

Lastly, it is now easy to see that the largest eigenvalues have an upper bound of $\alpha_i \leq 1 + \eta \frac{n}{K}(a-b) + \eta \frac{c}{\sqrt{n^{\rho+2}}} \leq 1 + 2\eta \frac{n}{K}(a-b) = 1 + 4\eta\gamma$. Similarly, the smallest eigenvalues have a lower bound of $\alpha_i \geq 1 - \eta \frac{n}{K}(a-b) - \eta \frac{c}{\sqrt{n^{\rho+2}}} \geq 1 - 2\eta \frac{n}{K}(a-b) = 1 - 4\eta\gamma$.

This concludes the proof.

C. Proofs From section 3.3

C.1. Proof of Theorem 3.8

From the properties of $\overline{\boldsymbol{C}}$ described in Section 3, the top K eigenvectors of $\overline{\boldsymbol{C}}$ lie in the space spanned by the vectors $\boldsymbol{1}_{\mathbb{V}_1}, \boldsymbol{1}_{\mathbb{V}_2}, \ldots, \boldsymbol{1}_{\mathbb{V}_K}$. In the span of these vectors, suppose we wish to find a vector $\boldsymbol{x}'$ that minimizes $\|\boldsymbol{x} - \boldsymbol{x}'\|^2$. Writing a general vector in the span as $\boldsymbol{x}' = \sum_i \alpha_i \boldsymbol{1}_{\mathbb{V}_i}$, we see that in order to minimize $\|\boldsymbol{x} - \boldsymbol{x}'\|^2$, we must set $\alpha_i = \mu_{\mathbb{V}_i}$. Thus, if Γ is the projection matrix onto the span of $\{\boldsymbol{1}_{\mathbb{V}_i}\}_{i=1}^K$, we have

$$\|\boldsymbol{x} - \boldsymbol{\mu}\| = \|(I - \Gamma)\boldsymbol{x}\|.$$

Let us now compare $(I - \Gamma)\boldsymbol{x}$ and $(I - \overline{\Pi})\boldsymbol{w}^{(T)}$. The two main differences are the following: first, $\overline{\Pi}$ is a $2n \times 2n$ projection matrix (as opposed to Γ, which is $n \times n$). Second, $\overline{\Pi}$ is a projection onto a $(K-1)$ dimensional subspace (as opposed to Γ, which projects to K dimensions). The structure of the eigenvectors of $\overline{\boldsymbol{L}}$ (see the proof of Lemma 3.5) implies that the projection $(I - \overline{\Pi})$ is equivalent to applying an appropriate projection to the first n and the second n coordinates separately. Thus, the term $(I - \overline{\Pi})\boldsymbol{w}^{(t_f)}$ includes the projection error for both $\boldsymbol{x}$ and $\boldsymbol{y}$. Second, since the error in projecting to a K dimensional space is only smaller than the error in projecting to a $(K-1)$ dimensional subspace of it, we get:

$$\|(I - \Gamma)\boldsymbol{x}\| \leq \|(I - \overline{\Pi})\boldsymbol{w}^{(t_f)}\|.$$

Next, we try to use the conclusion of Lemma 3.7. For this, we need to relate Π and $\overline{\Pi}$. But this turns out to be easy in our case. Recall that Π and $\overline{\Pi}$ are, respectively, the projections onto the span of the top $(K-1)$ eigenvectors of $\boldsymbol{M}$ and $\overline{\boldsymbol{M}}$ respectively. Since the gap between the $(K-1)$ and Kth largest eigenvalues for $\boldsymbol{M}$ (and also $\overline{\boldsymbol{M}}$) is $\geq \frac{n(a-b)}{2K}$ (which is $\Omega\left(\frac{1}{n}\right)$), we can use the classic Davis-Kahan Sin-Theta theorem (Stewart & Sun, 1990), applied to the spectral norm, to obtain:

$$\|\Pi - \overline{\Pi}\| \leq \frac{2K}{n(a-b)}\|\boldsymbol{R}\|,$$

where $\|\boldsymbol{R}\| \leq \frac{c}{\sqrt{n^{\rho+2}}}$ as we saw before. This implies that

$$\|(I - \Gamma)\boldsymbol{x}\| \leq \|(I - \Pi)\boldsymbol{w}^{(t_f)}\| + O(\frac{1}{\sqrt{n^\rho}})\|\boldsymbol{w}^{(t_f)}\|.$$

Combining this with Lemma 3.7 and noting that $\frac{1}{\Delta}$ dominates $\frac{1}{\sqrt{n^\rho}}$, the theorem follows.

C.2. Proof of Theorem 3.10

Fix some $i, j \in [K]$ with $i \neq j$. Consider the $2n$ dimensional vector $\boldsymbol{u}$ obtained by taking $\frac{\mathbf{1}_{\mathbb{V}_i}}{\sqrt{|\mathbb{V}_i|}} - \frac{\mathbf{1}_{\mathbb{V}_j}}{\sqrt{|\mathbb{V}_j|}}$ and appending 0 for the remaining n coordinates. Since the partitions $\mathbb{V}_i$ are all of size n/K, this is simply the vector $\sqrt{\frac{K}{n}}(\mathbf{1}_{\mathbb{V}_i} - \mathbf{1}_{\mathbb{V}_j})$ with 0s appended. Note that by construction, $\|\boldsymbol{u}\|^2 = 2$.

Now, since the initial vector $\boldsymbol{w}^{(0)}$ was chosen uniformly at random, its projection $\boldsymbol{z}^{(0)}$ is a random vector in the space spanned by the top $(K-1)$ eigenvectors of $\boldsymbol{L}$. Since u is a vector also in this span, and since it has constant length, we expect u to have an inner product of roughly $\|\boldsymbol{z}^{(0)}\|/\sqrt{(K-1)}$ with $\boldsymbol{z}^{(0)}$. In fact, since the inner product is distributed as a Gaussian, we have that

$$\Pr[|\langle \boldsymbol{u}, \boldsymbol{z}^{(0)} \rangle| < \frac{1}{10K^2\sqrt{K}}\|\boldsymbol{z}^{(0)}\|] < \frac{1}{10K^2}.$$

Now, from the discussion preceding the theorem, we have that with probability $1 - \frac{1}{10K^2}$,

$$|\langle \boldsymbol{u}, \boldsymbol{z}^{(t_f)} \rangle| \geq \frac{1}{20K^2\sqrt{K}}\|\boldsymbol{z}^{(t_f)}\|.$$

By definition, note that $|\mu_{\mathbb{V}_i} - \mu_{\mathbb{V}_j}| = \sqrt{\frac{K}{n}}|\langle \boldsymbol{u}, \boldsymbol{z}^{(t_f)} \rangle|$. This implies that with probability $1 - \frac{1}{10K^2}$, we have (for fixed i, j),

$$|\mu_{\mathbb{V}_i} - \mu_{\mathbb{V}_j}| \geq \frac{1}{20K^2\sqrt{n}}\|\boldsymbol{z}^{(t_f)}\|.$$

Taking a union bound over all i, j and using the value of $\|\boldsymbol{z}^{(t_f)}\|$, the theorem follows.

D. Auxiliary Lemmas

Theorem D.1 (Weyl's Theorem). *Let $\overline{\boldsymbol{A}}$ and $\boldsymbol{E}$ be $n \times n$ symmetric matrices. Let $\overline{\lambda}_1 \geq ... \geq \overline{\lambda}_n$ be the eigenvalues of $\overline{\boldsymbol{A}}$ and $\lambda_1 \geq ... \geq \lambda_n$ be the eigenvalues of $\boldsymbol{A} = \overline{\boldsymbol{A}} + \boldsymbol{E}$. Then $|\lambda_1 - \overline{\lambda}_1| \leq \|\boldsymbol{E}\|$.*

Theorem D.2 (Popoviciu's Inequality). *Let M and m be the upper and lower bounds of the entries of a probability vector $\boldsymbol{p} = (p_1, ..., p_n)$. Then*

$$\frac{1}{n}\sum_{i=1}^{n}(p_i - \frac{1}{n})^2 \leq \frac{1}{4}(M - m)^2. \tag{33}$$

Proof. First, notice that

$$0 \leq \frac{1}{n}\sum_{i=1}^{n}(M - p_i)(p_i - m)$$

$$= \frac{1}{n}\sum_{i=1}^{n}(Mp_i - mM - p_i^2 + mp_i)$$

$$= -mM + \frac{M + m}{n} - \frac{1}{n}\sum_{i=1}^{n}p_i^2.$$

Therefore,

$$\frac{1}{n}\sum_{i=1}^{n}(p_i - \frac{1}{n})^2 = \frac{1}{n}\sum_{i=1}^{n}p_i^2 - \frac{1}{n}$$
$$\leq -mM + \frac{M+m}{n} - \frac{1}{n^2}$$
$$= (M - \frac{1}{n})(\frac{1}{n} - m)$$
$$\leq \left(\frac{M - \frac{1}{n} + \frac{1}{n} - m}{2}\right)^2$$
$$= \frac{1}{4}(M - m)^2,$$

where the last inequality makes use of the AM-GM inequality.

$\square$