From which world is your graph?

Cheng Li
College of William and Mary

Felix Ming Fai Wong
Google Inc

Zhenming Liu
College of William and Mary

Varun Kanade
University of Oxford

Abstract

Discovering statistical structure from links is a fundamental problem in the analysis of social networks. Choosing a misspecified model, or equivalently, an incorrect inference algorithm will result in an invalid analysis or even falsely uncover patterns that are in fact artifacts of the model. This work focuses on unifying two of the most widely used link-formation models: the stochastic blockmodel (SBM) and the small world (or latent space) model (SWM). Integrating techniques from kernel learning, spectral graph theory, and nonlinear dimensionality reduction, we develop the first statistically sound polynomial-time algorithm to discover latent patterns in sparse graphs for both models. When the network comes from an SBM, the algorithm outputs a block structure. When it is from an SWM, the algorithm outputs estimates of each node's latent position.

1 Introduction

Discovering statistical structures from links is a fundamental problem in the analysis of social networks. Connections between entities are typically formed based on underlying feature-based similarities; however these features themselves are partially or entirely hidden. A question of great interest is to what extent can these latent features be inferred from the observable links in the network. This work focuses on the so-called assortative setting, the principle that *similar individuals are more likely to interact with each other*. Most stochastic models of social networks rely on this assumption, including the two most famous ones – the stochastic blockmodel [1] and the small-world model [2, 3], described below.

Stochastic Blockmodel (SBM). In a stochastic blockmodel [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], nodes are grouped into disjoint "communities" and links are added randomly between nodes, with a higher probability if nodes are in the same community. In its simplest incarnation, an edge is added between nodes within the same community with probability p, and between nodes in different communities with probability q, for $p > q$. Despite arguably naïve modelling choices, such as the independence of edges, algorithms designed with SBM work well in practice [14, 15].

Small-World Model (SWM). In a small-world model, each node is associated with a latent variable x_i, *e.g.,* the geographic location of an individual. The probability that there is a link between two nodes is proportional to an inverse polynomial of some notion of distance, $\mathrm{dist}(x_i, x_j)$, between them. The presence of a small number of "long-range" connections is essential to some of the most intriguing properties of these networks, such as small diameter and fast decentralized routing algorithms [3]. In general, the latent position may reflect geographic location as well as more abstract concepts, *e.g.,* position on a political ideology spectrum.

The Inference Problem. Without observing the latent positions, or knowing which model generates the underlying graph, the adjacency matrix of a social graph typically looks like the one shown in Fig. 5(a) (App. A.1). However, if the model generating the graph is known, it is then possible to run a suitable "clustering algorithm" [14, 16] that reveals the hidden structure. When the vertices

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

are ordered suitably, the SBM's adjacency matrix looks like the one shown in Fig. 5(b) (App. A.1) and that of the SWM looks like the one shown in Fig. 5(c) (App. A.1). Existing algorithms typically depend on knowing the "true" model and are tailored to graphs generated according to one of these models, *e.g.,* [14, 16, 17, 18].

Our Contributions. We consider a latent space model that is general enough to include both these models as special cases. In our model, an edge is added between two nodes with a probability that is a decreasing function of the distance between their latent positions. This model is a fairly natural one, and it is quite likely that a variant has already been studied; however, to the best of our knowledge there is no known statistically sound and computationally efficient algorithm for latent-position inference on a model as general as the one we consider.

1. A unified model. We propose a model that is a natural generalization of both the stochastic blockmodel and the small-world model that captures some of the key properties of real-world social networks, such as small out-degrees for ordinary users and large in-degrees for celebrities. We focus on a simplified model where we have a modest degree graph only on "celebrities"; the supplementary material contains an analysis of the more realistic model using somewhat technical machinery.

2. A provable algorithm. We present statistically sound and polynomial-time algorithms for inferring latent positions in our model(s). Our algorithm approximately infers the latent positions of almost all "celebrities" ($1 - o(1)$-fraction), and approximately infers a constant fraction of the latent positions of ordinary users. We show that it is statistically impossible to err on at most $o(1)$ fraction of ordinary users by using standard lower bound arguments.

3. Proof-of-concept experiments. We report several experiments on synthetic and real-world data collected on Twitter from Oct 1 and Nov 30, 2016. Our experiments demonstrate that our model and inference algorithms perform well on real-world data and reveal interesting structures in networks.

Additional Related Work. We briefly review the relevant published literature. *1. Graphon-based techniques*. Studies using graphons to model networks have focused on the statistical properties of the estimators [19, 20, 21, 22, 23, 24, 25, 26, 27], with limited attention paid to computational efficiency. The "USVT" technique developed recently [28] estimates the kernel well when the graph is dense. Xu et al. [29] consider a polynomial time algorithm for a sparse model similar to ours, but focus on edge classification rather than latent position estimation. *2. Correspondence analysis in political science*. Estimating the ideology scores of politicians is an important research topic in political science [30, 31, 32, 33, 34, 35, 17, 18]. High accuracy heuristics developed to analyze dense graphs include [17, 18].

Organization. Section 2 describes background, our model and results. Section 3 describes our algorithm and an gives an overview of its analysis. Section 4 contains the experiments.

2 Preliminaries and Summary of Results

Basic Notation. We use c_0, c_1, etc. to denote constants which may be different in each case. We use whp to denote with high probability, by which we mean with probability larger $1 - \frac{1}{n^c}$ for any c. All notation is summarized in Appendix H for quick reference.

Stochastic Blockmodel. Let n be the number of nodes in the graph with each node assigned a label from the set $\{1, \ldots, k\}$ uniformly at random. An edge is added between two nodes with the same label with probability p and between the nodes with different labels with probability q, with $p > q$ (assortative case). In this work, we focus on the $k = 2$ case, where $p, q = \Omega\left((\log n)^c/n\right)$ and the community sizes are exactly the same. (Many studies of the regimes where recovery is possible have been published [36, 9, 5, 8].)

Let A be the adjacency matrix of the realized graph and let $M = \mathbb{E}[A] = \begin{pmatrix} P & Q \\ Q & P \end{pmatrix}$, where P and $Q \in R^{\frac{n}{2} \times \frac{n}{2}}$ with every entry equal to p and q, respectively. We next explain the inference algorithm, which uses two key observations. *1. Spectral Properties of M.* M has rank 2 and the non-trivial eigenvectors are $(1, \ldots, 1)^\mathsf{T}$ and $(1, \ldots, 1, -1, \ldots, -1)$ corresponding to eigenvalues $n(p+q)/2$ and $n(p-q)/2$, respectively. If one has access to M, the hidden structure in the graph is revealed merely by reading off the second eigenvector. *2. Low Discrepancy between A and M.* Provided the average degree $n(p+q)/2$ and the gap $p - q$ are large enough, the spectrum and eigenspaces of the matrices A and M can be shown to be close using matrix concentration inequalities and the Davis-Kahan theorem [37, 38]. Thus, it is sufficient to look at the projection of the columns of A onto the top two eigenvectors of A to identify the hidden latent structure.

Small-World Model (SWM). In a 1-dim. SWM, each node v_i is associated with an independent latent variable $x_i \in [0,1]$ that is drawn from the uniform distribution on $[0,1]$. The probability of a link between two nodes is $\Pr[\{v_i, v_j\} \in E] \propto \frac{1}{|x_i - x_j|^{\Delta} + c_0}$, where $\Delta > 1$ is a hyper-parameter.

The inference algorithm for small-world models uses different ideas. Each edge in the graph is considered as either "short-range" or "long-range." Short-range edges are those between nodes that are nearby in latent space, while long-range edges have end-points that are far away in latent space. After removing the long-range edges, the shortest path distance between two nodes scales proportionally to the corresponding latent space distance (see Fig. 6 in App. A.2). After obtaining estimates for pairwise distances, standard buidling blocks are used to find the latent positions x_i [39]. The key observation used to remove the long-range edges is: an edge $\{v_i, v_j\}$ is a short-range edge if and only if v_i and v_j will share many neighbors.

A Unified Model. Both SBM and SWM are special cases of our unified latent space model. We begin by describing the full-fledged bipartite (heterogeneous) model that is a better approximation of real-world networks, but requires sophisticated algorithmic techniques (see Appendix C for a detailed analysis). Next, we present a simplified (homogeneous) model to explain the key ideas.

Bipartite Model. We use graphon model to characterize the stochastic interactions between users. Each individual is associated with a latent variable in $[0,1]$. The bipartite graph model consists of two types of users: the left side of the graph $\mathbf{Y} = \{y_1, \ldots, y_m\}$ are the *followers* (ordinary users) and the right side $\mathbf{X} = \{x_1, \ldots, x_n\}$ are the *influencers* (celebrities). Both y_i and x_i are i.i.d. random variables from a distribution $\mathcal{D}$. This assumption follows the convention of existing heterogeneous models [40, 41]. The probability that two individuals y_i and x_j interact is $\kappa(y_i, x_j)/n$, where $\kappa : [0,1] \times [0,1] \to (0,1]$ is a kernel function (these are sometimes referred to as graphon-based models [42, 23, 22]). Throughout this paper we assume that κ is a small-world kernel, *i.e.,* $\kappa(x,y) = c_0/(\|x-y\|^{\Delta} + c_1)$ for some $\Delta > 1$ and suitable constants c_0, c_1, and that $m = \Theta(n \cdot \mathrm{polylog}(n))$. Let $B \in R^{m \times n}$ be a binary matrix that $B_{i,j} = 1$ if and only if there is an edge between y_i and x_j. Our goal is to estimate $\{x_i\}_{i \in [n]}$ based on B for suitably large n.

Simplified Model. The graph only has the node set is $\mathbf{X} = \{x_1, ..., x_n\}$ of celebrity users. Each x_i is again an i.i.d. random variable from $\mathcal{D}$. The probability that two users v_i and v_j interact is $\kappa(x_i, x_j)/C(n)$. The denominator is a normalization term that controls the edge density of the graph. We assume $C(n) = n/\mathrm{polylog}(n)$, *i.e.,* the average degree is $\mathrm{polylog}(n)$. Unlike the SWM where the x_i are drawn uniformly from $[0,1]$, in the unified model $\mathcal{D}$ can be flexible. When $\mathcal{D}$ is the uniform distribution, the model is the standard SWM. When $\mathcal{D}$ has discrete support (*e.g.,* $x_i = 0$ with prob. 1/2 and $x_i = 1$ otherwise), then the unified model reduces to the SBM. Our distribution-agnostic algorithm can automatically select the most suitable model from SBM and SWM, and infer the latent positions of (almost) all the nodes.

Bipartite vs. Simplified Model. The simplified model suffers from the following problem: If the average degree is $O(1)$, then we err on estimating every individual's latent position with a constant probability (*e.g.,* whp the graph is disconnected), but in practice we usually want a high prediction accuracy on the subset of nodes corresponding to high-profile users. Assuming that the average degree is $\omega(1)$ mismatches empirical social network data. Therefore, we use a bipartite model that introduces heterogeneity among nodes: By splitting the nodes into two classes, we achieve high estimation accuracy on the influencers and the degree distribution more closely matches real-world data. For example, in most online social networks, nodes have $O(1)$ average degree, and a small fraction of users (influencers) account for the production of almost all "trendy" content while most users (followers) simply consume the content.

Additional Remarks on the Bipartite Model. *1. Algorithmic contribution.* Our algorithm computes $B^{\mathsf{T}}B$ and then regularizes the product by shrinking the diagonal entries before carrying out spectral analysis. Previous studies of the bipartite graph in similar settings [43, 44, 45] attempt to construct a regularized product using different heuristics. Our work presents the first theoretically sound regularization technique for spectral algorithms. In addition, some studies have suggested running SVD on B directly (*e.g.,* [27]). We show that the (right) singular vectors of B *do not* converge to the eigenvectors of K (the matrix with entries $\kappa(x_i, x_j)$). Thus, it is necessary to take the product and use regularization. *2. Comparison to degree-corrected models (DCM).* In DCM, each node v_i is associated with a degree parameter $D(v_i)$. Then we have $\Pr[\{v_i, v_j\} \in \mathbb{E}] \propto D(v_i)\kappa(x_i, x_j)D(v_j)$. The DCM model implies the subgraph induced by the highest degree nodes is dense, which is inconsistent with real-world networks. There is a need for better tools to analyze the asymptotic behavior of such models and we leave this for future work (see, *e.g.,* [40, 41]).

Theoretical Results. Let F be the cdf of $\mathcal{D}$. We say F and κ are **well-conditioned** if:

(1) F has finitely many points of discontinuity, i.e., the closure of the support of F can be expressed as the union of non-overlapping closed intervals I_1, I_2, ..., I_k for a finite number k.

(2) F is near-uniform, i.e., for any interval I that has non-empty overlap with F's support, $\int_I dF(x) \geq c_0|I|$, for some constant c_0.

(3) Decay Condition: The eigenvalues of the integral operator based on κ and F decay sufficiently fast. We define the $\mathcal{K}f(x) = \int \kappa(x,x')f(x')dF(x')$ and let $(\lambda_i)_{i\geq 1}$ denote the eigenvalues of $\mathcal{K}$. Then, it holds that $\lambda_i = O(i^{-2.5})$.

If we use the small-word kernel $\kappa(x,y) = c_0/(|x-y|^{\Delta} + c_1)$ and choose F that give rise to SBM or SWM, in each case the pair F and κ are well-conditioned, as described below. As the decay condition is slightly more invoved, we comment upon it. The condition is a mild one. When F is uniformly distributed on $[0,1]$, it is equivalent to requiring $\mathcal{K}$ to be twice differentiable, which is true for the small world kernel (Theorem F.2). When F has a finite discrete support, there are only finitely many non-zero eigenvalues, *i.e.,* this condition also holds. The decay condition holds in more general settings, *e.g.,* when F is piecewise linear [46] (see App. F). Without the decay condition, we would require much stronger assumptions: Either the graph is very dense or $\Delta \gg 2$. Neither of these assumptions is realistic, so effectively our algorithm fails to work. In practice, whether the decay condition is satisfied can be checked by making a log-log plot and it has been observed for several real-world networks, the eigenvalues follow a power-law distribution [47].

Next, we define the notion of latent position recovery for our algorithms.

Definition 2.1 ((α,β,γ)-Aproximation Algorithm)**.** *Let I_i, F, and $\mathcal{K}$ be defined as above, and let $R_i = \{x_j : x_j \in I_i\}$. An algorithm is called an (α,β,γ)-approximation algorithm if*
1. It outputs a collection of disjoint points $C_1, C_2, ..., C_k$ such that $C_i \subseteq R_i$, which correspond to subsets of reconstructed latent variables.
2. For each C_i, it produces a distance matrix $D^{(i)}$. Let $G_i \subseteq C_i$ be such that for any $i_j, i_k \in G_i$

$$D^{(i)}_{i_j,i_k} \leq |x_{i_j} - x_{i_k}| \leq (1+\beta)D^{(i)}_{i_j,i_k} + \gamma. \tag{1}$$

3. $|\bigcup_i G_i| \geq (1-\alpha)n$.
In bipartite graphs, Eq.(1) *is required for only influencers.*

We do not attempt to optimize constants in this paper. We set $\alpha = o(1)$, β a small constant, and $\gamma = o(1)$. Definition 2.1 allows two types of errors: C_is are not required to form a partition *i.e.,* some nodes can be left out, and a small fraction of estimation errors is allowed in C_i, *e.g.,* if $x_j = 0.9$ but $\widehat{x}_j = 0.2$, then the j-th "row" in $D^{(i)}$ is incorrect. To interpret the definition, consider the blockmodel with 2 communities. Condition 1 means that our algorithm will output two disjoint groups of points. Each group corresponds to one block. Condition 2 means that there are pairwise distance estimates within each group. Since the true distances for nodes within the same block are zero, our estimates must also be zero to satisfy Eq.1. Condition 3 says that the portion of misclassified nodes is $\alpha = o(1)$. We can also interpret the definition when we consider a small-world graph, in which case $k = 1$. The algorithm outputs pairwise distances for a subset C_1. We know that there is a sufficiently large $G_1 \subseteq C_1$ such that the pairwise distances are all correct in C_1.

Our algorithm does not attempt to estimate the distance between C_i and C_j for $i \neq j$. When the support contains multiple disjoint intervals, *e.g.,* in the SBM case, it first pulls apart the nodes in different communities. Estimating the distance between intervals, given the output of our algorithm is straightforward. Our main result is the following.

Theorem 2.2. *Using the notation above, assume F and κ are well-conditioned, and $C(n)$ and m/n are $\Omega(\log^c n)$ for some suitably large c. The algorithm for the simplified model shown in Figure 1 and that for the bipartite model shown in Figure 8 give us an $(1/\log^2 n, \epsilon, O(1/\log n))$-approximation algorithm w.h.p. for any constant ϵ. Furthermore, the distance estimates $D^{(i)}$ for each C_i are constructed using the shortest path distance of an unweighted graph.*

We focus only on the simplified model and the analysis for the bipartite graph algorithm is in Appendix C.

Pairwise Estimation to Line-embedding and High-dimensional Generalization. Our algorithm builds estimates on pairwise latent distance and uses well-studied metric-embedding methods [48, 49] as blackboxes to infer latent positions. Our inference algorithm can be generalized to d-dimensional space with d being a constant. But the metric-embedding on ℓ_p^d becomes increasingly difficult, *e.g.,* when $d = 2$, the approximation ratio for embedding a graph is $\Omega(\sqrt{n})$ [50].

LATENT-INFERENCE(A)

// **Step 1. Estimate** Φ .
$\widehat{\Phi} =$ SM-EST(A).
// **Step 2. Execute isomap algo.**
$D =$ ISOMAP-ALGO($\widehat{\Phi}$)
// **Step 3. Find latent variables.**
Run a line embedding algorithm [48, 49].

ISOMAP-ALGO($\widehat{\Phi}$)

Execute $S \leftarrow$ DENOISE($\widehat{\Phi}$) (See Section 3.2)
// S is a subset of $[n]$.
Build $G = \{S, E\}$ s.t. $\{i, j\} \in E$ iff
    $|(\tilde{\Phi}_d)_i - (\tilde{\Phi}_d)_j| \leq \ell/\log n$ (ℓ a constant).
Compute D such $D(i, j)$ is the shortest
    path distance between i and j when $i, j \in S$.
return D

SM-EST(A)

$[\tilde{U}_A, \tilde{S}_A, \tilde{V}_A] = \mathrm{svd}(A)$.
Let also λ_i be i-th singular value of A.
// let t be a suitable parameter.
$d =$ DECIDETHRESHOLD$(t, \rho(n))$.
S_A: diagonal matrix comprised of $\{\lambda_i\}_{i \leq d}$
U_A, V_A: the singular vectors
    corresponding to S_A.
Let $\widehat{\Phi} = \sqrt{C(n)} U_A S_A^{1/2}$.
return $\widehat{\Phi}$

DECIDETHRESHOLD$(t, \rho(n))$

// This procedure decides d the number
    of Eigenvectors to keep.
// t is a tunable parameter. See Proposition 3.1.
$d = \arg\max_d \{\lambda_d(\frac{A}{\rho(n)}) - \lambda_{d+1}(\frac{A}{\rho(n)}) \geq 10(\frac{t}{\rho(n)})^{\frac{2}{29}}\}$.

Figure 1: Subroutines of our Latent Inference Algorithm.

3 Our algorithm

As previous noted, SBM and SWM are special cases of our unified model and both require different algorithmic techniques. Given that it is not surprising that our algorithm blends ingredients from both sets of techniques. Before proceeding, we review basics of kernel learning.

Notations. Let A be the adjacency matrix of the observed graph (simplified model) and let $\rho(n) \triangleq n/C(n)$. Let K be the matrix with entries $\kappa(x_i, x_j)$. Let $\tilde{U}_K \tilde{S}_K \tilde{V}_K^\mathsf{T}$ ($\tilde{U}_A \tilde{S}_A \tilde{V}_A^\mathsf{T}$) be the SVD of K (A). Let d be a parameter to be chosen later. Let S_K (S_A) be a $d \times d$ diagonal matrix comprising the d-largest eigenvalues of K (A). Let U_K (U_A) and V_K (V_A) be the corresponding singular vectors of K (A). Finally, let $\bar{K} = U_K S_K V_K^\mathsf{T}$ ($\bar{A} = U_A S_A V_A^\mathsf{T}$) be the low-rank approximation of K (A). Note that when a matrix is positive definite and symmetric SVD coincides with eigen-decomposition; as a consequence $U_K = V_K$ and $U_A = V_A$.

Kernel Learning. Define an integral operator $\mathcal{K}$ as $\mathcal{K}f(x) = \int \kappa(x, x') f(x') dF(x')$. Let $\psi_1, \psi_2, \ldots$ be the eigenfunctions of $\mathcal{K}$ and $\lambda_1, \lambda_2, \ldots$ be the corresponding eigenvalues such that $\lambda_1 \geq \lambda_2 \geq \cdots$ and $\lambda_i \geq 0$ for each i. Also let $N_{\mathcal{H}}$ be the number of eigenfunctions/eigenvalues of $\mathcal{K}$, which is either finite or countably infinite. We recall some important properties of $\mathcal{K}$ [51, 24]. For $x \in [0, 1]$, define the feature map $\Phi(x) = (\sqrt{\lambda_j}\psi_j(x) : j = 1, 2, ...)$, so that $\langle \Phi(x), \Phi(x') \rangle = \kappa(x, x')$. We also consider a truncated feature $\Phi_d(x) = (\sqrt{\lambda_j}\psi_j(x) : j = 1, 2, ..., d)$. Intuitively, if λ_j is too small for sufficiently large j, then the first d coordinates (*i.e.,* Φ_d) already approximate the feature map well. Finally, let $\Phi_d(\mathbf{X}) \in \mathbb{R}^{n \times d}$ such that its (i, j)-th entry is $\sqrt{\lambda_j}\psi_j(x_i)$. Let's further write $(\Phi_d(\mathbf{X}))_{:,i}$ be the i-th column of $\Phi_d(\mathbf{X})$. Let $\Phi(\mathbf{X}) = \lim_{d \to \infty} \Phi_d(\mathbf{X})$. When the context is clear, shorten $\Phi_d(\mathbf{X})$ and $\Phi(\mathbf{X})$ to Φ_d and Φ, respectively.

There are two main steps in our algorithm which we explain in the following two subsections.

3.1 Estimation of Φ through K and A

The mapping $\Phi : [0, 1] \to \mathbb{R}^{N_{\mathcal{H}}}$ is bijective so a (reasonably) accurate estimate of $\Phi(x_i)$ can be used to recover x_i. Our main result is the design of a data-driven procedure to choose a suitable number of eigenvectors and eigenvalues of A to approximate Φ (see SM-EST(A) in Fig. 1).

Proposition 3.1. *Let t be a tunable parameter such that $t = o(\rho(n))$ and $t^2/\rho(n) = \omega(\log n)$. Let d be chosen by* DECIDETHRESHOLD$(\cdot)$. *Let $\widehat{\Phi} \in \mathbb{R}^{N_{\mathcal{H}}}$ be such that its first d-coordinates are equal to $\sqrt{C(n)} U_A S_A^{1/2}$, and its remaining entries are 0. If $\rho(n) = \omega(\log n)$ and $\mathcal{K}$ (F and κ) is well-conditioned, then with high probability:*

$$\|\widehat{\Phi} - \Phi\|_F = O\left(\sqrt{n}\,(t/(\rho(n)))^{\frac{2}{29}}\right) \tag{2}$$

Specifically, by letting $t = \rho^{2/3}(n)$, we have $\|\widehat{\Phi} - \Phi\|_F = O\left(\sqrt{n}\rho^{-2/87}(n)\right)$.

Remark on the Eigengap. In our analysis, there are three groups of eigenvalues: the eigenvalues of $\mathcal{K}$, those of K, and those of A. They are in different scales: $\lambda_i(\mathcal{K}) \leq 1$ (resulting from the

fact that $\kappa(x,y) \leq 1$ for all x and y), and $\lambda_i(A/\rho(n)) \approx \lambda_i(K/n) \approx \lambda_i(\mathcal{K})$ if n and $\rho(n)$ are sufficiently large. Thus, $\lambda_d(\mathcal{K})$ are *independent of* n for a *fixed* d and should be treated as $\Theta(1)$. Also $\delta_d \triangleq \lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K}) \to 0$ as $d \to \infty$. Since *the procedure of choosing* d *depends on* $C(n)$ (and thus also on n), δ_d *depends on* n *and can be bounded by a function in* n. *This is the reason why Proposition 3.1 does not explicitly depend on the eigengap*. We also note that we cannot directly find δ_d based on the input matrix A. But standard interlacing results can give $\delta_d = \Theta(\lambda_d(A/\rho(n)) - \lambda_{d+1}(A/\rho(n)))$ (see Lemma B.6 in Appendix.)

Intuition of the algorithm. Using Mercer's theorem, we have $\langle \Phi(x_i), \Phi(x_j) \rangle = \lim_{d\to\infty} \langle \Phi_d(x_i), \Phi_d(x_j) \rangle = \kappa(x_i, x_j)$. Thus, $\lim_{d\to\infty} \Phi_d \Phi_d^\mathsf{T} = K$. On the other hand, we have $(\tilde{U}_K \tilde{S}_K^{1/2})(\tilde{U}_K \tilde{S}_K^{1/2})^\mathsf{T} = K$. Thus, $\Phi_d(\mathbf{X})$ and $\tilde{U}_K \tilde{S}_K^{1/2}$ are approximately the same, up to a unitary transformation. We need to identify sources of errors to understand the approximation quality.

Error source 1 *Finite samples to learn the kernel*. We want to infer about "continuous objects" κ and $\mathcal{D}$ (specifically the eigenfunctions of $\mathcal{K}$) but K gives only the kernel values of a finite set of pairs.From standard results in Kernel PCA [52, 24], we have with probability $\geq 1 - \epsilon$,

$$\|U_K S_K^{1/2} W - \Phi_d(X)\|_F \leq 2\sqrt{2} \frac{\sqrt{\log \epsilon^{-1}}}{\lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})} = 2\sqrt{2} \frac{\sqrt{\log \epsilon^{-1}}}{\delta_d}.$$

Error source 2. *Only observe* A. We observe only the realized graph A and not K, though it holds that $\mathbb{E}A = K/C(n)$. Thus, we can only use singular vectors of $C(n)A$ to approximate $\tilde{U}_K \tilde{S}_K^{1/2}$. We have: $\left\| \sqrt{C(n)} U_A S_A^{1/2} W - U_K S_K^{1/2} \right\|_F = O\left(\frac{t\sqrt{dn}}{\delta_d^2 \rho(n)} \right)$. When A is dense (*i.e.,* $C(n) = O(1)$), the problem is analyzed in [24]. We generalize the results in [24] for the sparse graph case. See Appendix B for a complete analysis.

Error source 3. *Truncation error*. When i is large, the noise in $\lambda_i(A)(\tilde{U}_A)_{:,i}$ "outweighs" the signal. Thus, we need to decide a d such that only the first d eigenvectors/eigenvalues of A are used to approximate Φ_d. Here, we need to address *the truncation error*: the tail $\{\sqrt{\lambda_i}\psi_i(x_j)\}_{i>d}$ is thrown away.

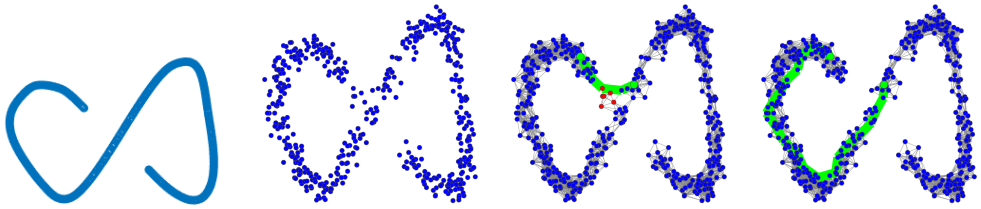
Next we analyze the magitude of the tail. We abuse notation so that $\Phi_d(x)$ refers to both a d-dimensional vector and a $N_\mathcal{H}$-dimensional vector in which all the entries after d-th one are 0. We have $\mathbb{E}\|\Phi(x) - \Phi_d(x)\|^2 = \sum_{i>d} \mathbb{E}[(\sqrt{\lambda_i}\psi_i(x))^2] = \sum_{i>d} \lambda_i \int |\psi_i(x)|^2 dF(x) = \sum_{i>d} \lambda_i$. (A Chernoff bound is used to obtain that $\|\Phi - \Phi_d\|_F = O(\sqrt{n}/(\sqrt{\sum_{i>d} \lambda_i}))$. Using the decay condition, we show that a d can be identified so that the tail can be bounded by a polynomial in δ_d. The details are technical and are provided in the supplementary material (cf. Proof of Prop. 3.1 in Appendix B).

3.2 Estimating Pairwise Distances from $\widehat{\Phi}(x_i)$ through Isomap

See ISOMAP-ALGO$(\cdot)$ in Fig. 1 for the pseudocode. After we construct our estimate $\widehat{\Phi}_d$, we estimate K by letting $\widehat{K} = \widehat{\Phi}_d \widehat{\Phi}_d^\mathsf{T}$. Recalling $K_{i,j} = c_0/(|x_i - x_j|^\Delta + c_1)$, a plausible approach is to estimate $|x_i - x_j| = (c_0/\widehat{K}_{i,j} - c_1)^{1/\Delta}$. However, $\kappa(x_i, x_j)$ is a convex function in $|x_i - x_j|$. Thus, when $K_{i,j}$ is small, a small estimation error here will result in an amplified estimation error in $|x_i - x_j|$ (see also Fig. 7 in App. A.3). But when $|x_i - x_j|$ is small, $K_{i,j}$ is reliable (see the "reliable" region in Fig. 7).

Thus, our algorithm only uses large values of $K_{i,j}$ to construct estimates. The isomap technique introduced in topological learning [53, 54] is designed to handle this setting. Specifically, the set $\mathcal{C} = \{\Phi(x)\}_{x\in[0,1]}$ forms a curve in $\mathbb{R}^{N_\mathcal{H}}$ (Fig. 2(a)). Our estimate $\{\widehat{\Phi}(x_i)\}_{i\in[n]}$ will be a noisy approximation of the curve (Fig. 2(b)). Thus, we build up a graph on $\{\Phi(x_i)\}_{i\leq n}$ so that x_i and x_j are connected if and only if $\widehat{\Phi}(x_i)$ and $\widehat{\Phi}(x_j)$ are close (Fig. 2(c) and/or (d)). Then the shortest path distance on G approximates the geodesic distance on $\mathcal{C}$. By using the fact that κ is a radial basis kernel, the geodesic distance will also be proportional to the latent distance.

Corrupted nodes. Excessively corrupted nodes may help build up "undesirable bridges" and interfere with the shortest-path based estimation (cf.Fig. 2(c)). Here, the shortest path between two green nodes "jumps through" the excessively corrupted nodes (labeled in red) so the shortest path distance is very different from the geodesic distance.



(a) True features (b) Estimated features (c) Isomap w/o denoising (d) Isomap + denoising

Figure 2: Using the Isomap Algorithm to recover pairwise distances. (a) The true curve $\mathcal{C} = \{\Phi(x)\}_{x\in[0,1]}$ (b) Estimate $\widehat{\Phi}$ (c) Shows that an undesirable short-cut may exist when we run the Isomap algorithm and (d) Shows the result of running the Isomap algorithm after removal of the corrupted nodes.

Below, we describe a procedure to remove excessively corrupted nodes and then explain how to analyze the isomap technique's performance after their removal. Note that d in this section mostly refers to the shortest path distance (rather than the number of eigenvectors we keep as used in the previous section).

Step 1. Eliminate corrupted nodes. Recall that $x_1, x_2, ..., x_n$ are the latent variables. Let $z_i = \Phi(x_i)$ and $\widehat{z}_i = \widehat{\Phi}(x_i)$. For any $z \in \mathbb{R}^{N_{\mathcal{H}}}$ and $r > 0$, we let $\mathbf{Ball}(z, r) = \{z' : \|z' - z\| \le r\}$. Define projection $\mathrm{Proj}(z) = \arg\min_{z'\in\mathcal{C}} \|z' - z\|$, where $\mathcal{C}$ is the curve formed by $\{\phi(x)\}_{x\in[0,1]}$. Finally, for any point $z \in \mathcal{C}$, define $\Phi^{-1}(z)$ such that $\Phi(\Phi^{-1}(z)) = z$ (*i.e.,* z's original latent position). For the points that fall outside of $\mathcal{C}$, define $\Phi^{-1}(z) = \Phi^{-1}(\mathrm{Proj}(z))$.

Let us re-parametrize the error term in Propostion 3.1. Let $f(n)$ be that $\|\widehat{\Phi} - \Phi\|_F \le \sqrt{n}/f(n)$, where $f(n) = \rho^{2/87}(n) = \Omega(\log^2 n)$ for sufficiently large $\rho(n)$. By a Markov inequality, we have $\Pr_i[\|\widehat{\Phi}(x_i) - \Phi(x_i)\|^2 \ge 1/\sqrt{f(n)}] \le 1/f(n)$.

Intuitively, when $\|\widehat{\Phi}(x_i) - \Phi(x_i)\|^2 \ge 1/\sqrt{f(n)}$, i becomes a candidate that can serve to build up undesirable shortcuts. Thus, we want to eliminate these nodes.

Looking at a ball of radius $O(1/\sqrt{f(n)})$ centered at a point $\widehat{z}_i$, consider two cases.

Case 1. If $\widehat{z}_i$ is close to $\mathrm{Proj}(\widehat{z}_i)$, *i.e.,* corresponding to the blue nodes in Figure 2(c). For exposition purpose, let us assume $\widehat{z}_i = z_i$. Now for any point z_j, if $|x_i - x_j| = O(f^{-1/\Delta}(n))$, then by Lemma E.1, we have $\|\widehat{z}_i - \widehat{z}_j\| = O(1/\sqrt{f(n)})$, which means z_j is in $\mathbf{Ball}(z_i, O(1/\sqrt{f(n)}))$. The total number of such nodes will be in the order of $\Theta(n/f^{1/\Delta}(n))$, by using the near-uniform density assumption.

Case 2. If $\widehat{z}_i$ is far away from any point in $\mathcal{C}$, *i.e.,* corresponding to the red ball in Figure 2(c), any points in $\mathbf{Ball}(\widehat{z}_i, O(1/\sqrt{f(n)}))$ will also be far from $\mathcal{C}$. Then the total number of such nodes will be $O(n/f(n))$.

As $n/f^{1/\Delta}(n) = \omega(n/f(n))$ for $\Delta > 1$, there is a phase-transition phenomenon: When $\widehat{z}_i$ is far from $\mathcal{C}$, then a neighborhood of $\widehat{z}_i$ contains $O(n/f(n))$ nodes. When $\widehat{z}_i$ is close to $\mathcal{C}$, then a neighborhood of $\widehat{z}_i$ contains $\omega(n/f(n))$ nodes.

We can leverage this intuition to design a counting-based algorithm to eliminate nodes that are far from $\mathcal{C}$:

$$\textsc{Denoise}(\widehat{z}_i) : \text{If } |\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})| < n/f(n), \text{ remove } \widehat{z}_i. \tag{3}$$

Theoretical result. We classify a point i into three groups:
1. **Good**: Satisfying $\|\widehat{z}_i - \mathrm{Proj}(\widehat{z}_i)\| \le 1/\sqrt{f(n)}$. We further partition the set of good points into two parts. Good-I are points such that $\|\widehat{z}_i - z_i\| \le 1/\sqrt{f(n)}$, while Good-II are points that are good but not in Good-I.
2. **Bad**: when $\|z_i - \mathrm{Proj}(z_i)\| > 4/\sqrt{f(n)}$.
3. **Unclear**: otherwise.

We prove the following result (see Appendix E for a proof).

Lemma 3.2. *After running* DENOISE *that uses the counting-based decision rule, all good points are kept, all bad points are eliminated, and all unclear points have no performance guarantee. The total number of eliminated nodes is* $\le n/f(n)$.

Step 2. An isomap-based algorithm. Wlog assume there is only one closed interval for $\mathrm{support}(F)$. We build a graph G on $[n]$ so that two nodes $\widehat{z}_i$ and $\widehat{z}_j$ are connected if and only if $\|\widehat{z}_i - \widehat{z}_j\| \le \ell/\sqrt{f(n)}$, where ℓ is a sufficiently large constant (say 10). Consider the shortest path distance between arbitrary pairs of nodes i and j (that are not eliminated.) Because the corrupted nodes are removed, the whole path is around $\mathcal{C}$. Also, by the uniform density assumption, walking on the shortest path in G is equivalent to walking on $\mathcal{C}$ with "uniform speed", *i.e.,* each edge on the path will map to an approximately fixed distance on $\mathcal{C}$. Thus, the shortest path distance scales with the latent distance, *i.e.,* $(d-1)\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta} \le |x_i - x_j| \le d\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell+8}{\sqrt{f(n)}}\right)^{2/\Delta}$, which implies Theorem 2.2. See Appendix E.3 for a detailed analysis.

Bipartite Model. Although we have focused our discussion on the simplified model, we make a few remarks about inference in the more realistic bipartite model. A more detailed discussion and the inference algorithm is available at the beginning of Appendix C and full details follow in that appendix. In the bipartite case, we no longer have access to the kernel matrix K for pairs of celebrity nodes; however, any non-diagonal entry of $B^{\mathsf{T}}B$, say the ij^{th} one, can be written as $\sum_k Z_{ik}Z_{jk}$ where Z_{ik} and Z_{jk} are independent Bernoulli random variables with parameters $\kappa(x_i, y_k)$ and $\kappa(x_j, y_k)$. This gives rise to a *square* kernel (of κ) which can be used to identify the eigenvalues and eigenvectors of the kernel operator $\mathcal{K}$ used in the analysis of the simplified model. The diagonal entries have to be regularized as there is no independence in the corresponding terms.

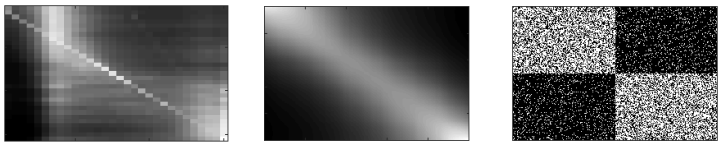
Discussion: "Gluing together" two algorithms? The unified model is much more flexible than SBM and SWM. We were intrigued that the generalized algorithm needs only to "glue together" important techniques used in both models: Step 1 uses the spectral technique inspired by SBM inference methods, while Step 2 resembles techniques used in SWM: the isomap G only connects between two nodes that are close, which is the same as throwing away the long-range edges.

4 Experiments

Algo.	ρ	Slope of β	S.E.	p-value
Ours	**0.53**	9.54	0.28	< 0.001
Mod. [55]	0.16	1.14	0.02	< 0.001
CA [18]	0.20	0.11	7e-4	< 0.001
Maj [56]	0.13	0.09	0.02	< 0.001
RW [54]	0.01	1.92	0.65	< 0.001
MDS [49]	0.05	30.91	120.9	0.09

Figure 3: Latent Estimates vs. Ground-truth.

We apply our algorithm to a social interaction graph from Twitter to construct users' ideology scores. We assembled a dataset by tracking keywords related to the 2016 US presidential election for 10 million users. First, we note that as of 2016 the Twitter interaction graph behaves "in between" the small-world and stochastic blockmodels (see Figure 4), *i.e.,* the latent distributions are bimodal but not as extreme as the SBM.



(a) Inferred kernel (b) SWM (c) SBM

Figure 4: Visualization of real and synthetic networks. (a) Our inferred kernel matrix, which is "in-between" the small-world model (b) and the stochastic blockmodel (c).

Ground-truth data. Ideology scores of the US Congress (estimated by third parties [57]) are usually considered as a "ground-truth" (see, *e.g.,* [18]) dataset. We apply our algorithm and other baselines on Twitter data to estimate the ideology score of politicians (members of the 114th Congress), and observe that our algorithm has the highest correlation with ground-truth. See Fig. 3. Beyond correlation, we also need to estimate the statistical significance of our estimates. We set up a linear model $y \sim \beta_1 \widehat{x} + \beta_0$, in which $\widehat{x}$'s are our estimates and y's are ground-truth. We then use bootstrapping to compute the standard error of our estimator, and then use the standard error to estimate the p-value of our estimator. The details of this experiment and additional empirical evaluation are available in Appendix G.

References

[1] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[2] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440–442, 1998.

[3] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000.

[4] Se-Young Yun and Alexandre Proutière. Optimal cluster recovery in the labeled stochastic block model. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 965–973, 2016.

[5] Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461, 2015.

[6] Emmanuel Abbe and Colin Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. *arXiv preprint arXiv:1512.09080*, 2015.

[7] Emmanuel Abbe and Colin Sandon. Community detection in the general stochastic block model: Fundamental limits and efficient algorithms for recovery. In *Proceedings of 56th Annual IEEE Symposium on Foundations of Computer Science, Berkely, CA, USA*, pages 18–20, 2015.

[8] Laurent Massoulié. Community detection thresholds and the weak Ramanujan property. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 694–703. ACM, 2014.

[9] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *arXiv preprint arXiv:1311.4115*, 2013.

[10] Peter J. Bickel and Aiyou Chen. A nonparametric view of network models and newmangirvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[11] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

[12] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[13] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

[14] Frank McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE, 2001.

[15] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.

[16] Ittai Abraham, Shiri Chechik, David Kempe, and Aleksandrs Slivkins. Low-distortion inference of latent similarities from a multiplex social network. In *SODA*, pages 1853–1872. SIAM, 2013.

[17] Pablo Barberá. Birds of the Same Feather Tweet Together. Bayesian Ideal Point Estimation Using Twitter Data. 2012.

[18] Pablo Barberá, John T. Jost, Jonathan Nagler, Joshua A. Tucker, and Richard Bonneau. Tweeting from left to right. *Psychological Science*, 26(10):1531–1542, 2015.

[19] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, 97:1090–1098, 2001.

[20] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[21] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[22] Edo M Airoldi, Thiago B Costa, and Stanley H Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 692–700. Curran Associates, Inc., 2013.

[23] Sofia C. Olhede Patrick J. Wolfe. Nonparametric graphon estimation. 2013.

[24] Minh Tang, Daniel L. Sussman, and Carey E. Priebe. Universally consistent vertex classification for latent positions graphs. *Ann. Statist.*, 41(3):1406–1430, 06 2013.

[25] Patrick J. Wolfe and David Choi. Co-clustering separately exchangeable network data. *The Annals of Statistics*, 42(1):29–63, 2014.

[26] Varun Kanade, Elchanan Mossel, and Tselil Schramm. Global and local information in clustering labeled block models. *IEEE Trans. Information Theory*, 62(10):5906–5917, 2016.

[27] Karl Rohe, Tai Qin, and Bin Yu. Co-clustering directed graphs to discover asymmetries and directional communities. *Proceedings of the National Academy of Sciences*, 113(45):12679–12684, 2016.

[28] Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *Ann. Statist.*, 43(1):177–214, 02 2015.

[29] Jiaming Xu, Laurent Massoulié, and Marc Lelarge. Edge label inference in generalized stochastic block models: from spectral theory to impossibility results. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvri, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 903–920, Barcelona, Spain, 13–15 Jun 2014. PMLR.

[30] K.T̃. Poole and H. Rosenthal. A spatial model for legislative roll call analysis. *American Journal of Political Science*, 29(2):357–384, 1985.

[31] M. Laver, K. Benoit, and J. Garry. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(2), 2003.

[32] J. Clinton, S. Jackman, and D. Rivers. The statistical analysis of roll call data. *American Political Science Review*, 98(2):355–370, 2004.

[33] S. Gerrish and D. Blei. How the vote: Issue-adjusted models of legislative behavior. In *Proc. NIPS*, 2012.

[34] S. Gerrish and D. Blei. Predicting legislative roll calls from text. In *Proc. ICML*, 2011.

[35] J. Grimmer and B. M. Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 2013.

[36] Emmanuel Abbe. Community detection and the stochastic block model. 2016.

[37] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[38] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. *SIAM J. Numer. Anal.*, 7:1–46, 1970.

[39] Piotr Indyk and Jiri Matoušek. Low-distortion embeddings of finite metric spaces. *Handbook of discrete and computational geometry*, page 177, 2004.

[40] Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *Ann. Statist.*, 40(4):2266–2292, 08 2012.

[41] Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In C.j.c. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3120–3128. 2013.

[42] L. Lovasz. *Large Networks and Graph Limits*. American Mathematical Society colloquium publications. American Mathematical Society, 2012.

[43] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 269–274, New York, NY, USA, 2001. ACM.

[44] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang. Bipartite network projection and personal recommendation. 76(4):046115, October 2007.

[45] Felix Ming Fai Wong, Chee-Wei Tan, Soumya Sen, and Mung Chiang. Quantifying political leaning from tweets, retweets, and retweeters. *IEEE Trans. Knowl. Data Eng.*, 28(8):2158–2172, 2016.

[46] H. König. *Eigenvalue Distribution of Compact Operators*. Operator Theory: Advances and Applications. Birkhäuser, 1986.

[47] Milena Mihail and Christos Papadimitriou. On the eigenvalue power law. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 254–262. Springer, 2002.

[48] Mihai Badoiu, Julia Chuzhoy, Piotr Indyk, and Anastasios Sidiropoulos. Low-distortion embeddings of general metrics into the line. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 225–233, 2005.

[49] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[50] Piotr Indyk and Jiri Matousek. Low-distortion embeddings of finite metric spaces. In *in Handbook of Discrete and Computational Geometry*, pages 177–196. CRC Press, 2004.

[51] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[52] Lorenzo Rosasco, Mikhail Belkin, and Ernesto De Vito. On learning with integral operators. *J. Mach. Learn. Res.*, 11:905–934, March 2010.

[53] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

[54] Vin De Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, 2003.

[55] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74, 2006.

[56] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 2007.

[57] Joshua Tauberer. Observing the unobservables in the us congress. *Law Via the Internet*, 2012.

[58] Tosio Kato. Variation of discrete spectra. *Communications in Mathematical Physics*, 111(3):501–504, 1987.

[59] Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 1649–1656, 2005.

[60] Roberto Imbuzeiro Oliveira et al. Sums of random hermitian matrices and an inequality by rudelson. *Electron. Commun. Probab*, 15(203-212):26, 2010.

[61] Roberto Oliveira. Sums of random hermitian matrices and an inequality by rudelson. *Electron. Commun. Probab.*, 15:203–212, 2010.

[62] H. Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71:441–479, 1912.

[63] Andrew J. Wathen and Shengxin Zhu. On spectral distribution of kernel matrices related to radial basis functions. *Numerical Algorithms*, 70(4):709–726, 2015.

[64] Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3120–3128, USA, 2013. Curran Associates Inc.

[65] Raviv Cohen and Derek Ruths. Classifying political orientation on twitter: Its not easy! In *International AAAI Conference on Weblogs and Social Media*, 2013.

[66] Jeffrey R Lax and Justin H Phillips. The democratic deficit in the states. *American Journal of Political Science*, 56(1):148–166, 2012.

A Additional Illustrations

This section provides additional illustrations related to our work.

A.1 Model Selection Probem (presented in Section 1)

Without observing the latent positions or knowing which model generated the underlying graph, the adjacency matrix of a social graph typically looks like the one shown in Fig. 5(a). However, if the model generating the graph is known, it is then possible to run a suitable "clustering algorithm" [14, 16] that reveals the hidden structure. When the vertices are ordered suitably, the SBM's adjacency matrix looks like the one shown in Fig. 5(b) and that of the SWM looks like the one shown in Fig. 5(c).

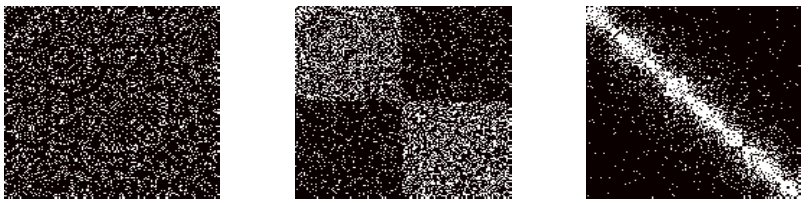


Figure 5: Inference problem in social graphs: Given an input graph (a), are we able to shuffle the nodes so that statistical patterns are revealed? A major problem in network inference is the model selection problem. The input here can come from stochastic block model (b) or small-world model (c).

A.2 Algorithm for the Small-world Model (presented in Section 2)

The inference algorithm for small-world networks uses different ideas. Each edge in the graph can be thought of as a "short-range" or "long-range" one. Short-range edges are those between nodes that are nearby in latent space, while long-range ones have end-points that are far away in latent space. After the removal of all the long-range edges, the shortest path distance between two nodes scales proportionally to the corresponding latent space distance (see Fig. 6). Once estimates for pairwise distances are obtained, standard buidling blocks may be used to find the latent positions x_i [39].

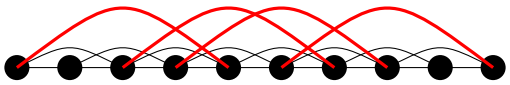


Figure 6: In the small-world model, after removal of long range edges (red thick), the shortest-path distance between two nodes approximates latent space distance

A.3 Sensitivity of the Gram matrix K (presented in Section 3.2)

After we construct our estimate $\widehat{\Phi}_d$, we may estimate K by letting $\widehat{K} = \widehat{\Phi}_d \widehat{\Phi}_d^{\mathsf{T}}$. Recalling $K_{i,j} = c_0/(|x_i - x_j|^{\Delta} + c_1)$, one plausible approach would be estimating $|x_i - x_j| = (c_0/\widehat{K}_{i,j} - c_1)^{1/\Delta}$. A main issue with this approach is that $\kappa(x_i, x_j)$ is a convex function in $|x_i - x_j|$. Thus, when $K_{i,j}$ is small, a small estimation error here will result in an amplified estimation error in $|x_i - x_j|$ (cf. Fig. 7). But when $|x_i - x_j|$ is small, $K_{i,j}$ is reliable (see the "reliable" region in Fig. 7).

B Simplified model case: Using A to approximate $\Phi(X)$

This section proves the following proposition.

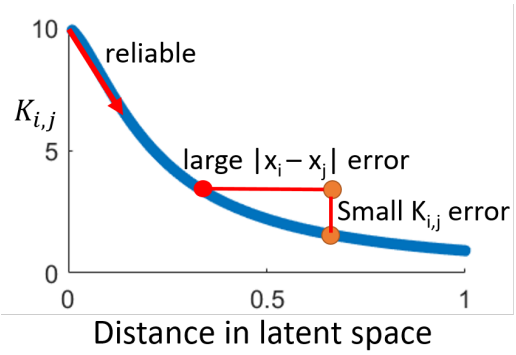


Figure 7: The behavior of $\widehat{K}_{i,j}$.

Proposition B.1 (Repeat of Proposition 3.1)**.** *Let t be a tunable parameter such that $t = o(\rho(n))$ and $t^2/\rho(n) = \omega(\log n)$. Let d be chosen by* DECIDETHRESHOLD$(\cdot)$*. Let $\widehat{\Phi} \in \mathbb{R}^{N_{\mathcal{H}}}$ be such that its first d-coordinates are equal to $\sqrt{C(n)}U_A S_A^{1/2}$. If $\rho(n) = \omega(\log n)$ and $\mathcal{K}$ is well-conditioned, then with high probability:*

$$\|\widehat{\Phi} - \Phi\|_F = O\left(\sqrt{n}\left(\frac{t}{\rho(n)}\right)^{\frac{2}{29}}\right) \tag{4}$$

We will break down our analysis into three components, each of which corresponds to an approximation error source presented in Section 3.1. Some statements that appeared earlier are repeated in this section to make it self-contained.

Before proceeding, we need more notation.

Additional Notation. Let $\mathcal{H}$ denote the reproducing kernel Hilbert space of κ, so that each element $\eta \in \mathcal{H}$ can be uniquely expressed as $\eta = \sum_j a_j \sqrt{\lambda_j}\psi_j$. The inner product of elements in $\mathcal{H}$ is given by $\left\langle \sum_j a_j \sqrt{\lambda_j}\psi_j, \sum_j b_j \sqrt{\lambda_j}\psi_j \right\rangle_{\mathcal{H}} = \sum_j a_j b_j$.

B.1 Error Source 1: Finite Samples to Learn the Kernel

Recall that we want to infer about "continous objects" κ and $\mathcal{D}$ (more specifically eigenfunctions of the integral operator $\mathcal{K}$ derived using κ and F) but K gives only the kernel values for a finite set of pairs, so estimates constructed from K are only approximations. Here, we need only an existing result from Kernel PCA [52, 24].

Lemma B.2. *Using the notations above, we have*

$$\|U_K S_K^{1/2} W - \Phi_d(X)\|_F \leq 2\sqrt{2}\frac{\sqrt{\log \epsilon^{-1}}}{\lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})} = 2\sqrt{2}\frac{\sqrt{\log \epsilon^{-1}}}{\delta_d} \tag{5}$$

We remark on the (implicit) dependence on the sample size in (5). Here, the right-hand side is the total error on all the samples, which is independent of n, and hence the *average* square error shrinks as $O(1/n)$.

B.2 Error Source 2: Only Observe A

We observe only the realized graph A rather than the gram matrix K, such that $\mathbb{E}A = K/C(n)$. Thus, we can use only singular vectors of $C(n)A$ to approximate $\tilde{U}_K \tilde{S}_K^{1/2}$. Our main goal is to prove the following lemma.

Lemma B.3. *Using the notation above, we have*

$$\left\| \sqrt{C(n)} U_A S_A^{1/2} - U_K S_K^{1/2} \right\|_F = O\left(\frac{t\sqrt{dn}}{\delta_d^2 \rho(n)} \right) \tag{6}$$

The outline of the proof is as follows.

Step 1. Show that $\|A - K/C(n)\|$ is small. This can be done by observing that $A_{i,j}$ are independent for different pairs of $i < j$ and applying a tail inequality on independent matrix sum.

Step 2. Apply a Davis-Kahan theorem to show that $\mathcal{P}_A$ and $\mathcal{P}_K$ are close.. Let $\mathcal{P}_A = U_A U_A^\intercal$ and $\mathcal{P}_K = U_K U_K^\intercal$ be the projection operators onto the linear subspaces spanned by the eigenvectors corresponding to the d largest eigenvalues of A and K respectively. Davis-Kahan theorem gives a sufficient condition that U_A and U_K are close (upto a unitary operation), *i.e.,* $\|A - K/C(n)\|$ needs to be small (from step 1) and $\delta_d = \lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})$ needs to be large (from d is a suitable constant). Thus U_A and U_K are close up to a unitary operation, which implies $\mathcal{P}_A$ and $\mathcal{P}_K$ are close. We will specifically show that $\|\mathcal{P}_A - \mathcal{P}_K\|_{\text{HS}}$ is small. $\|\cdot\|_{\text{HS}}$ refers to the Hilbert-Schmidt norm.

Definition B.4. *The Hilbert-Schmidt norm of a bounded operator A on a Hilbert space H is*

$$\|A\|_{\text{HS}}^2 = \sum_{i \in I} \|Ae_i\|^2, \tag{7}$$

where $\{e_i : i \in I\}$ is an orthonormal basis of H.

Step 3. Show that $\sqrt{C(n)} U_A S_A^{1/2}$ and $U_K S_K^{1/2}$ are close (up to a unitary operation). We first argue that $\|\mathcal{P}_A C(n) A - \mathcal{P}_K K)\|$ is small. Then by observing that $\sqrt{C(n)} U_A S_A^{1/2}$ and $U_K S_K^{1/2}$ are "square root" of $\mathcal{P}_A(C(n)A)$ and $\mathcal{P}_K K$, we can show $\sqrt{C(n)} U_A S_A^{1/2}$ and $U_K S_K^{1/2}$ are close.

We now follow the workflow to prove the proposition.

B.2.1 Step 1. $\|A - K\|$ is small

We use the following concentration bound for matrix [37].

Theorem B.5. *Consider a finite sequence $\{X_k\}$ of independent random, self-adjoint matrices with dimension d. Assume that each random matrix satisfies*

$$\mathbb{E}[X_k] = 0 \quad \textit{and} \quad \lambda_{\max}(X_k) \leq R \quad a.s. \tag{8}$$

Then for all $t \geq 0$,

$$\Pr[\lambda_{\max}(\sum_k X_k) \geq t] \leq d \exp\left(-\frac{t^2/2}{\sigma^2 + Rt/3} \right), \tag{9}$$

where $\sigma^2 = \|\sum_k \mathbb{E}[X_k^2]\|$.

We apply the above theorem to bound $\|A - K/(C(n))\|$. Let $p_{i,j} = K_{i,j}/C(n)$ represent the probability that there is a link between v_i and v_j. Let random matrix $E_{i,j} \in \mathbb{R}^{n \times n}$ be that the (i,j)-th entry and (j,i)-th entry are 1 with probability $p_{i,j}$, and 0 otherwise. The remaining entries in $E_{i,j}$ are all 0. Let $F_{i,j} = E_{i,j} - \mathbb{E}[E_{i,j}]$. Note that $A = \sum_{i \leq j} E_{i,j}$ and $\{E_{i,j}\}_{i \leq j}$ are all independent. We also have $\|A - K/C(n)\| = \|\sum_{i \leq j} F_{i,j}\|$.

Note that:

1. $\lambda_{\max}(F_{i,j}) = \Theta(1)$ a.s.
2. $F_{i,j}^2 \in \mathbb{R}^2$ is a matrix such that only (i,i)-th and (j,j)-th entries can non-zero. Furthermore,
$$(F_{i,j}^2)_{i,i} = (F_{i,j}^2)_{j,j} = \begin{cases} p_{i,j}^2 & \text{with probability } 1 - p_{i,j} \\ (1 - p_{i,j})^2 & \text{with probability } p_{i,j} \end{cases}$$
Thus, $\mathbb{E}[(F_{i,j}^2)_{i,i}] \leq p_{i,j}$. One can see that $\sum_{i \leq j} \mathbb{E}[(F_{i,j}^2)]$ is a diagonal matrix such that the (i,i)-th entry is $\leq 2 \sum_{j \leq n} \mathbb{E}[(F_{i,j})_{i,i}^2] = O(np_{i,j}) = O(\rho(n))$. Thus σ^2 in the theorem shall be $O(\rho(n))$.

We then have

$$\Pr\left[\lambda_{\max}\left(\sum_{i,j} F_{i,j}\right) \geq t\right] \leq \exp\left(-\frac{t^2}{O(\rho(n)+t)}\right) = \exp\left(-\Theta\left(\min\left(\frac{t^2}{\rho(n)}, t\right)\right)\right) \quad (10)$$

We shall also see that $\rho(n) = \omega(t)$ is needed. Thus,

$$\Pr\left[\left\|A - \frac{K}{C(n)}\right\| \geq t\right] \leq \exp\left(-\Theta\left(\frac{t^2}{\rho(n)}\right)\right). \quad (11)$$

B.2.2 Step 2. Show that $\|\mathcal{P}_A - \mathcal{P}_K\|_{\mathrm{HS}}$ is small

Recall that $\delta_d = \lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})$. Because the projection is scale-invariant, we work on the matrices $C(n)A/n = A/\rho(n)$ and K/n instead of A and K. By standard results from kernel PCA [52], we have with probability $\geq 1-\epsilon$,

$$\lambda_d\left(\frac{K}{n}\right) - \lambda_{d+1}\left(\frac{K}{n}\right) \geq \delta_d - 4\sqrt{2}\sqrt{\frac{\log(1/\epsilon)}{n}} \quad (12)$$

Define

$$\begin{array}{ll} S_1 & = \{\lambda : \lambda \geq \lambda_d(K/n) - t/\rho(n)\} \\ S_2 & = \{\lambda : \lambda \leq \lambda_{d+1}(K/n) + t/\rho(n)\}. \end{array} \quad (13)$$

Comparing δ with $t/\rho(n)$. We next relate δ_d with $t/\rho(n)$. Recall that $t = o(\rho(n))$ our algorithm DECIDETHRESHOLD$(t, \rho(n))$ in Fig. 1). We claim that $\delta_d = \omega(t/\rho(n))$.

Lemma B.6. *Using the notations above. Suppose we use* DECIDETHRESHOLD$(t, \rho(n))$ *in Fig. 1 to decide the number of eigenvectors/eigenvalues to keep, we have*

$$\delta_d = \Theta(\lambda_d(A/\rho(n)) - \lambda_{d+1}(A/\rho(n))) = \omega(t/\rho(n)).$$

Proof. Note that

$$|\lambda_d(\mathcal{K}) - \delta_d(A/\rho(n))| \leq |\lambda_d(\mathcal{K}) - \lambda_d(K/n)| + |\lambda_d(K/n) + \lambda_d(A/\rho(n))|.$$

From [52], we have

$$|\lambda_d(\mathcal{K}) - \lambda_d(K/n)| = O\left(\sqrt{\frac{\log(1/\epsilon)}{n}}\right) = O\left(\frac{t}{\rho(n)}\right).$$

Then from Step 1 and [58], we have

$$|\lambda_d(K/n) - \lambda_d(A/\rho(n))| \leq \|K/n - A/\rho(n)\| = O(t/\rho(n)).$$

Thus, we have

$$|\lambda_d(\mathcal{K}) - \lambda_d(A/\rho(n))| = O(t/\rho(n)).$$

Similarly, we can show that

$$|\lambda_{d+1}(\mathcal{K}) - \lambda_{d+1}(A/\rho(n))| = O(t/\rho(n)).$$

Finally, note that

$$\begin{array}{rcl} |\lambda_d(A/\rho(n)) - \lambda_{d+1}(A/\rho(n))| & \leq & |\lambda_d(A/\rho(n)) - \lambda_d(\mathcal{K})| + |\lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})| \\ & & +|\lambda_{d+1}(A/\rho(n)) - \lambda_{d+1}(\mathcal{K})|. \end{array}$$

Thus,

$$\begin{array}{rcl} |\lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})| & \geq & |\lambda_d(A/\rho(n)) - \lambda_{d+1}(A/\rho(n))| - |\lambda_d(A/\rho(n)) - \lambda_d(\mathcal{K})| \\ & & +|\lambda_{d+1}(A/\rho(n)) - \lambda_{d+1}(\mathcal{K})| \\ & = & \omega(t/\rho(n)) \quad \text{(Using the way DECIDETHRESHOLD}(t, \rho(n)) \text{ chooses } d.) \end{array}$$

□

We have $\operatorname{dist}(S_1, S_2) \geq \delta_d - 2t/\rho(n) - 8\sqrt{2}\sqrt{\frac{\log(2/\epsilon)}{n}} \geq \delta_d/2$.

One can also see that the first d eigenvalues of K/n and $C(n)A/n$ are in S_1, and the rest are in S_2. Then by a Davis-Kahan theorem [38], we have whp

$$\|\mathcal{P}_A - \mathcal{P}_K\| \leq \frac{\|C(n)A/n - K/n\|}{\operatorname{dist}(S_1, S_2)} = O\left(\frac{t}{\rho(n)\delta_d}\right). \tag{14}$$

Step 3. Show that $\sqrt{C(n)}U_A S_A^{1/2}$ and $U_K S_K^{1/2}$ are close (up to a unitary operation). We first argue that $\|\mathcal{P}_A(C(n)A) - \mathcal{P}_K K\|$ is small. Before proceeding, let us re-scale the matrices so that their eigenvalues are in the same magnitude of those of $\mathcal{K}$. We have $\mathcal{P}_A(C(n)A) - \mathcal{P}_K(K) = n\left(\mathcal{P}_A(A/\rho(n)) - \mathcal{P}_P(K/n)\right)$.

Note that $\|K/n\| = O(1)$ and $A/\rho(n) = O(1)$ whp when $\rho = \omega(\log n)$. We have

$$\|\mathcal{P}_A A/\rho(n) - \mathcal{P}_K K/n\| = \|(\mathcal{P}_A - \mathcal{P}_K)K/n\| + \|\mathcal{P}_A(A/\rho(n) - K/n)\| = O\left(\frac{t}{\rho(n)\delta_d}\right).$$

Observing that $\mathcal{P}_A A/\rho(n) = U_A S_A U_A^\intercal/\rho(n)$ and $\mathcal{P}_K K/n = U_K S_K U_K^\intercal/n$, we see that $U_A S_A^{1/2}$ and $U_K S_K^{1/2}$ are "square root" of $\mathcal{P}_A A/\rho(n)$ and $\mathcal{P}_K K/n$ (up to scaling). We use the following lemma to relate $U_A S_A^{1/2}$ and $U_K S_K^{1/2}$ (Lemma A.1 from [24]).

Lemma B.7. *Let A and B be n by n positive semi-definite matrices with $\operatorname{rank}(A) = \operatorname{rank}(B) = d$. Let $X, Y \in \mathbb{R}^{n,d}$ be full column rank matrices such that $XX^\intercal = A$ and $YY^\intercal = B$. Let δ be the smallest non-zero eigenvalues of B. Then there exists a rotational matrix W such that*

$$\|XW - Y\|_F \leq \frac{\|A - B\|(\sqrt{d\|A\|} + \sqrt{d\|B\|})}{\delta}. \tag{15}$$

By treating $A/\rho(n)$ and K/n as A and B in the Lemma, we have

$$\left\|U_A S_A^{1/2}/\sqrt{\rho(n)} - U_K S_K^{1/2}/\sqrt{n}\right\|_F = O\left(\frac{t\sqrt{d}}{\delta_d^2 \rho(n)}\right) \tag{16}$$

In other words,

$$\left\|\sqrt{C(n)}U_A S_A^{1/2} - U_K S_K^{1/2}\right\|_F = O\left(\frac{t\sqrt{dn}}{\delta_d^2 \rho(n)}\right) \tag{17}$$

This completes our proof of Lemma B.1.

B.3 Error source 3: truncation error

This section analyzes the error $\|\Phi_d - \Phi\|_F^2$. Recall that we abuse the notation to let $\Phi_d \in \mathbb{R}^{N_\mathcal{H}}$ by "padding" 0's after the d-th coordinate. We make an additional assumption that the eigengaps $\delta_d = \lambda_d(\mathcal{K}) - \lambda_{d+1}$ monotonically decreases whenever the number of non-zero eigenvalues is infinite. Removing this assumption requires arduous analysis with limited insights. Section D presents an analysis without the assumption.

We have $\mathbb{E}\|\Phi(x) - \Phi_d(x)\|^2 = \sum_{i>d} \mathbb{E}[(\sqrt{\lambda_i}\psi_i(x))^2] = \sum_{i>d} \lambda_i \int |\psi_i(x)|^2 dF(x) = \sum_{i>d} \lambda_i$. Then we may apply a standard Chernoff bound to obtain $\|\Phi - \Phi_d\|_F = O(\sqrt{n}/(\sqrt{\sum_{i>d} \lambda_i}))$.

In general small δ_d does not imply small tail *e.g.,* when $\lambda_i = \Theta(1/(i \log^2 i))$. Thus, we need to rely on the decay assumption in Theorem 2.2. *i.e.,* $\lambda_i(\mathcal{K}) = O(i^{-2.5})$. One can see that when this condition is given, $\sum_{i>d} \lambda_i$ can by bounded by $\delta_d^{1/3}$ and $d = O(\delta_d^{1/2})$.

BIPARTITE-EST(B)

// **Step 1. Regulating** B.
$A = B^{\mathsf{T}}B$.
$\mathrm{diag}(A) = \mathrm{diag}(A)^{\theta}$
// $\theta < 1$ so diagonal entries of A are shrinked.
// **Step 2. PCA with data-driving thresholding**
$[\tilde{U}_A, \tilde{S}_A, \tilde{V}_A] = \mathrm{svd}(A)$.
Let also λ_i be i-th singular value of A.
$d = \max_d\{\lambda_d - \lambda_{d+1} > \left(\frac{m}{n}\right)^{12/43}$.
S_A: diagonal matrix comprised of $\{\lambda_i\}_{i\le d}$
U_A, V_A: the corresponding singular vectors of S_A.
Let $\widehat{\Phi}_d = \frac{n^{3/4}}{m^{1/4}} U_A S_A^{1/4}$.
return $\widehat{\Phi}_d$.

Figure 8: Estimation of Φ for bipartite graphs.

Together with Lemma B.2 and Lemma B.3, we have

$$\|\widehat{\Phi} - \Phi\|_F = O\left(\sqrt{n}\left(\frac{t\sqrt{d}}{\rho(n)\delta_d^2} + \delta_d^{1/6}\right)\right).$$

By setting $\delta_d = (t/\rho(n))^{12/29}$, we have

$$\|\widehat{\Phi} - \Phi\|_F = O\left(\sqrt{n}\left(\frac{t}{\rho(n)}\right)^{\frac{2}{29}}\right) \tag{18}$$

This completes the proof of Proposition 3.1.

C Estimation of $\Phi(X)$ in the bipartite graph model

This section explains how we can use B to estimate $\Phi(\mathbf{X})$. See BIPARTITE-EST(B) in Fig. 8 for the pseudocode. A major difficulties in our analysis is that we cannot decouple the error into different approximation error sources like we did for the undirected graph case, *i.e.,* the approximation error sources interference with each other. So more involved analysis is needed.

Below is our main proposition.

Proposition C.1. *Consider the algorithm* BIPARTITE-EST($\cdot$). *Let* $\widehat{\Phi} \in R^{N_{\mathcal{H}}}$ *be that its first* d*-coordinates coincide with* $\widehat{\Phi}_d$ *returned by* BIPARTITE-EST *and the rest coordinates are* 0. *If the eigenvalues of* $\mathcal{K}$ *satisfies the decay condition, we have whp*

$$\|\widehat{\Phi} - \Phi\|_F = O\left(\sqrt{n}\left(\frac{n}{m}\right)^{2/43}\log n\right). \tag{19}$$

In other words, when $m = n\mathrm{poly}\log^c n$ for a suitably large c, then $\|\widehat{\Phi} - \Phi\|_F \le \sqrt{n}/\log^2 n$.

Intuition of the algorithm. Recall that $K \in R^{n\times n}$ such that $K_{i,j} = \kappa(x_i, x_j)$, K is the Gram matrix of the kernel $\kappa(\cdot,\cdot)$ obtained using the latent positions of the influencers, $x_1, \ldots, x_n$. Standard Kernel PCA results suggest that as long as n is sufficiently large, we may use K to estimate the eigenfunctions of $\mathcal{K}$. But we do not directly observe the matrix K. Instead, we observe B such that $\mathbb{E}[B_{j,i}] = \kappa(y_j, x_i)$. In other words, our "raw observations" are about the relationship between followers $\{y_j\}_{j=1}^m$ and influencers $\{x_i\}_{i=1}^n$, but our principal goal is to understand the relationships within $\{x_i\}_{i=1}^n$. Our algorithm does so by computing $B^{\mathsf{T}}B$. This product corresponds to another kernel. Specifically, let

$$\mu(x,x') = \int \kappa(x,z)\kappa(z,x')dF(z) \text{ and } \mathcal{M}f(x) = \int \mu(x,y)f(y)dF(y) \tag{20}$$

Finally, let $M \in R^{n \times n}$ such that $M_{i,j} = \mu(x_i, x_j)$. For $i \neq j$, one can see that $B^\mathsf{T} B$ and M are related as follows:

$$\mathbb{E}[(B^\mathsf{T} B)_{i,j}] = m \int \frac{\kappa(x_i, y)\kappa(y, x_j)}{n^2} dF(y) = \frac{m}{n^2} \mathbb{E}[M_{i,j}].$$

Regularization of the diagonals. Observing that $\mathbb{E}[(B^\mathsf{T} B)_{i,i}] \neq \frac{m}{n^2}\mathbb{E}[M_{i,j}]$, we need to shrink the diagonals of $B^\mathsf{T} B$ to construct A. Proposition C.1 works for all $\theta < 0.75$ but for exposition purpose, we focus on only the case $\theta = -\infty$, *i.e.,* setting the whole diagonal to be 0. One can use simple triangle inequalities on top of our techniques to analyze the general θ case.

The kernel μ. μ is a Mercer kernel as the Gram matrix for any $\{x_i\}_{i=1}^n$ is positive definite; however, μ *is not* a radial-basis kernel. This can seen from the fact that μ depends on the measure F. The quality of the isomap-based algorithm presented in the next section crucially depends on the kernel being a radial basis kernel (RBK).[1] Thus, we need to find a way to reconstruct κ from μ, and reconstruct μ from M.

Note that μ is a "square" of κ.

Lemma C.2. *Consider the linear operators $\mathcal{K}$ and $\mathcal{M}$. Let $\{\psi_i\}_{i\geq 1}$ and $\{\lambda_i\}_{i\geq 1}$ be the eigenfunctions and eigenvalues of $\mathcal{K}$. Then the eigenfunctions and eigenvalues of $\mathcal{M}$ are $\{\psi_i\}_{i\geq 1}$ and $\{\lambda_i^2\}_{i\geq 1}$, respectively.*

Proof. Let ψ be an eigenfunction of $\mathcal{K}$ with eigenvalue λ. We can see that

$$\begin{aligned}
\mathcal{M}\psi(x) &= \int \mu(x, y)\psi(y) dF(y) \\
&= \int\int \kappa(x, z)\kappa(z, y) dF(z)\psi(y) dF(y) \\
&= \int \kappa(x, z) \int \kappa(z, y)\psi_i(y) dF(y) dF(z) \\
&= \lambda \int \kappa(x, z)\psi(z) dF(z) \\
&= \lambda^2 \psi(x).
\end{aligned}$$

We can also verify that any function that is orthogonal to $\mathcal{K}$ will also be orthogonal to $\mathcal{M}$, showing that the dimension of $\mathcal{K}$ and $\mathcal{M}$ are the same. □

We break down the analysis into smaller steps:

- **Step 0 (known results):** Given K, we can approximate Φ.
- **Step 1.** If we have access to $\mathbb{E}A \approx M$, then we can approximate K^2, *i.e.,* $M \propto K^2$ ($A \propto B$ refer to that a suitable scalar s exists such that $\|sA - B\| = o(1)$).
- **Step 2.** Show that $A \propto M$ using Chernoff type inequalities for matrices (together with step 1, we have $A \propto K^2$).
- **Step 3.** Show that if $A \propto K^2$, then $A^{1/2} \propto K$ (note that we need to be able to properly define taking the square root of A as *e.g., A* could have negative eigenvalues). Thus, we can construct $\widehat{\Phi}_d$ from A.
- **Step 4.** Finally, argue that Φ_d approximates Φ well, *i.e.,* it is fine to truncate all the tail eigenvalues and eigenfunctions. Thus, we can construct $\widehat{\Phi} \in R^{N_\mathcal{H}}$ by appending a suitable number of 0's after the d-th coordinate so that $\widehat{\Phi}$ approximates Φ well.

We now walk through each step. In the proofs, constants c_0, c_1, etc. are used as "intermediate variables." Constants that appear in different proof should not be treated as the same unless stated explicitly.

[1] The isomap-based algorithm will still work but the approximation guarantee will be worse.

C.1 Notations and Step 0.

Recall that we let $\tilde{U}_K \tilde{S}_K \tilde{V}_K^\mathsf{T}$ ($\tilde{U}_M \tilde{S}_M \tilde{V}_M^\mathsf{T}$ and $\tilde{U}_A \tilde{S}_A \tilde{V}_A^\mathsf{T}$) be the SVD of K (M and A). Let S_K (S_M and S_A) be a $d \times d$ diagonal matrix comprising the d-largest eigenvalues of K (M and A). Let U_K (U_M and U_A) and V_K (V_M and V_A) be the corresponding singular vectors of K (M and A). Finally let $\bar{K} = U_K S_K V_K^\mathsf{T}$ ($\bar{M} = U_M S_M V_M^\mathsf{T}$ and $\bar{A} = U_A S_A V_A^\mathsf{T}$) be the low rank approximation of K (M and A).

Recall that Φ is the feature map associated with $\mathcal{K}$. We also let $\Phi^{\mathcal{M}}$ be the feature map associated with $\mathcal{M}$ and $\Phi_d^{\mathcal{M}}$ be the first d coordinates of $\Phi^{\mathcal{M}}$. For any $x \in [0,1]$, $\Phi_d(x)$ and $\Phi_d^{\mathcal{M}}(x)$ may be viewed as vectors that satisfy $\Phi_d^{\mathcal{M}}(x) = S^{1/2}\Phi_d(x)$, where S is a $d \times d$ diagonal matrix with $S_{ii} = \lambda_i(\mathcal{K})$.

Existing results regarding kernel PCA [52, 24] state that if K (or M) is sufficiently large, then we are able to reconstruct $\Phi_d(X)$ (or $\Phi_d^{\mathcal{M}}(X)$) for the observed datapoints. Specifically, let $X = \{x_1, \ldots, x_n\} \subseteq [0,1]$ be the latent positions of the observed datapoints and let $\Phi_d(X)$ and $\Phi_d^{\mathcal{M}}(X)$ be the $n \times d$ matrices, where the i^{th} row of $\Phi_d(X)$ and $\Phi_d^{\mathcal{M}}(X)$ are $\Phi_d(x_i)$ and $\Phi_d^{\mathcal{M}}(x_i)$, respectively.

We have with probability $\geq 1 - \epsilon$,

$$\|U_K S_K^{1/2} W - \Phi_d(X)\|_F \leq 2\sqrt{2}\frac{\sqrt{\log \epsilon^{-1}}}{\lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})}, \tag{21}$$

where W is an orthogonal matrix. Similar results also hold for M (see *e.g.,* [24]). Furthermore, S_K (and S_M) are approximations of the eigenvalues of $\mathcal{K}$ (and $\mathcal{M}$), *i.e.,* $(S_K)_{i,i}/n \to \lambda_i(\mathcal{K})$ ($(S_M)_{i,i}/n \to \lambda_i(\mathcal{M})$). The specific convergence rate is stated in Theorem F.1.

C.2 Step 1. From M to $\mathcal{M}$ and $\mathcal{K}$.

Using the above facts, we know that (these are hand-waving arguments to deliver intuitions; formal treatment will be presented below) (1) U_K and U_M "approximate" the eigenfunctions of $\mathcal{K}$ and $\mathcal{M}$ respectively; but eigenfunctions of $\mathcal{K}$ and $\mathcal{M}$ are the same so U_K and U_M are close. (2) $\lambda_i(K)/n \approx \lambda_i(\mathcal{K})$, $\lambda_i(M)/n \approx \lambda_i(\mathcal{M})$, and $\lambda_i(\mathcal{M}) = \lambda_i^2(\mathcal{K})$. Thus, we roughly have $\sqrt{\lambda_i(M)/n} = \lambda_i(K)/n$. These two observations imply we may have $S_K/n \approx \sqrt{S_M/n}$ and thus, $\sqrt{n} \cdot U_M S_M^{1/2} U_M^\mathsf{T} \approx U_K S_K U_K^\mathsf{T}$.

We now formalize the intuition. Our main goal is to prove the following proposition.

Proposition C.3. *Let K and M be the matrices defined above. Let $d \in \mathbb{N}$ and $\tilde{\delta}_d \in \mathbb{R}^+$ be such that if $(\lambda_i(K))_{i=1}^n$ and $(\lambda_i(M))_{i=1}^n$ are the eigenvalues of K and M, respectively, then $\lambda_i(K) - \lambda_{i+1}(K) \geq \tilde{\delta}_d$ and $\lambda_i(M) - \lambda_{i+1}(M) \geq \tilde{\delta}_d$ for $i = 1, \ldots, d-1$. Let $\mathcal{P}_K = U_K U_K^\mathsf{T}$ and $\mathcal{P}_M = U_M U_M^\mathsf{T}$ be projection operators onto the linear subspaces spanned by the eigenvectors corresponding to the d largest eigenvalues of K and M, respectively. Then with probability at least $1 - 4\epsilon$,*

$$\left\| \frac{\mathcal{P}_K K^2}{n^2} - \frac{\mathcal{P}_M M}{n} \right\| = O\left(\left(\frac{d^2 \log(1/\epsilon)}{n} \right)^{1/4} + \left(\frac{d \log(1/\epsilon)}{\tilde{\delta}_d^2 n} \right)^{1/2} \right) \tag{22}$$

Our analysis consists of two parts: (1) Show that $\frac{U_K S_K W_K}{n}$ and $\frac{U_M S_M^{1/2} W_M}{\sqrt{n}}$ are "close" , where W_K and W_M are orthogonal matrices, and (2) Show that if two matrices X and Y are close, then XX^T and YY^T are also close .

Part 1 of proof of Proposition C.3. We shall show the following lemma.

Lemma C.4. *Using the notation defined above, we have with probability at least $1 - 4\epsilon$,*

$$\left\| \frac{U_K S_K W_K}{n} - \frac{\Phi_d(X) S^{1/2}}{\sqrt{n}} \right\|_F \leq C \left(\left(\frac{d^2 \log(1/\epsilon)}{n} \right)^{1/4} + \left(\frac{d \log(1/\epsilon)}{\tilde{\delta}_d^2 n} \right)^{1/2} \right) \tag{23}$$

$$\left\| \frac{U_M S_M^{1/2} W_M}{\sqrt{n}} - \frac{\Phi_d(X) S^{1/2}}{\sqrt{n}} \right\|_F \leq \frac{4}{\tilde{\delta}_d} \sqrt{\frac{2 \log(1/\epsilon)}{n}}. \tag{24}$$

Proof of Lemma C.4. Let $\mathcal{H}$ be the Hilbert space corresponding to the kernel $\kappa(\cdot,\cdot)$. Then, we define the following two positive symmetric linear operators that act on $\mathcal{H}$.

$$\mathcal{K}_{\mathcal{H}}\eta = \int \langle \eta, \kappa(\cdot,x)\rangle_{\mathcal{H}}\kappa(\cdot,x)dF(x) \;\&\; \mathcal{K}_n\eta = \frac{1}{n}\sum_{i=1}^{n}\langle \eta, \kappa(\cdot,x_i)\rangle_{\mathcal{H}}\kappa(\cdot,x_i) \tag{25}$$

We note that the operators $\mathcal{K}_{\mathcal{H}}$ and $\mathcal{K}_n$ are closely related to $\mathcal{K}$ and K respectively, however while $\mathcal{K}_{\mathcal{H}}$ and $\mathcal{K}_n$ both act on $\mathcal{H}$, $\mathcal{K}$ acts on $L^2(\mathcal{X},F)$ and K acts on $\mathbb{R}^n$. The eigenvalues of $\mathcal{K}_{\mathcal{H}}$ and $\mathcal{K}$ are the same, and the eigenvalues of $\mathcal{K}_n$ and K/n are the same. Furthermore, if ψ is an eigenfunction of $\mathcal{K}$ with eigenvalue λ, then $\sqrt{\lambda}\psi$ is an eigenfunction of $\mathcal{K}_{\mathcal{H}}$ with eigenvalue λ; the $\sqrt{\lambda}$ factor is required to ensure that the norm in $\mathcal{H}$ of the eigenfunction is 1. Similarly if $\widehat{u}\in\mathbb{R}^n$ is an eigenvector of K/n with eigenvalue $\widehat{\lambda}$, then $\widehat{v}(\cdot)=\frac{1}{\sqrt{\widehat{\lambda}n}}\sum_{i=1}^{n}\kappa(\cdot,x_i)\widehat{u}_i$ is an eigenfunction of $\mathcal{K}_n$ with eigenvalue $\widehat{\lambda}$. See also [52].

For some $r\le d$, let λ_r and $\widehat{\lambda}_r$ be the r^{th} largest eigenvalues of $\mathcal{K}_{\mathcal{H}}$ and $\mathcal{K}_n$, respectively. Denote by $\mathcal{P}_r$ and $\widehat{\mathcal{P}}_r$ the projection operators on to the corresponding eigenfunctions. We will use the following Theorem, which generalizes Davis-Kahan sin theorem to linear operators.

Theorem C.5 (Thm. 2 [59])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be symmetric positive Hilbert-Schmidt operators on some Hilbert space. Let $\tilde{\delta}_d>0$ and $d\in\mathbb{N}$ such that*

1. *For all $i<d$, $\lambda_i(\mathcal{A})-\lambda_{i+1}(\mathcal{A})\ge\tilde{\delta}_d$.*
2. *$\|\mathcal{A}-\mathcal{B}\|_{\mathsf{HS}}\le\frac{\tilde{\delta}_d}{4}$.*

Let $\mathcal{P}_r^{\mathcal{A}}$ and $\mathcal{P}_r^{\mathcal{B}}$ be projection operators that project onto the eigenfunctions corresponding to the r^{th} largest eigenvalue of $\mathcal{A}$ and $\mathcal{B}$, respectively. Then,

$$\|\mathcal{P}_r^{\mathcal{A}}-\mathcal{P}_r^{\mathcal{B}}\|_{\mathsf{HS}}\le\frac{2\|\mathcal{A}-\mathcal{B}\|_{\mathsf{HS}}}{\tilde{\delta}_d} \tag{26}$$

Applying the theorem with $\mathcal{K}_{\mathcal{H}}$ and $\mathcal{K}_n$ taking the role of $\mathcal{A}$ and $\mathcal{B}$ and recalling that $\mathcal{P}_r$ and $\widehat{\mathcal{P}}_r$ are the corresponding projection operators, we have, $\|\mathcal{P}_r-\widehat{\mathcal{P}}_r\|_{\mathsf{HS}}\le 2\left(\|\mathcal{K}_{\mathcal{H}}-\mathcal{K}_n\|_{\mathsf{HS}}\right)/\tilde{\delta}_d$. It can be shown that with probability (over the random draw of $\{x_1,\ldots,x_n\}$) at least $1-2\epsilon$, $\|\mathcal{K}_{\mathcal{H}}-\mathcal{K}_n\|_{\mathsf{HS}}\le 2\sqrt{\frac{2\log(1/\epsilon)}{n}}$; the proof of this claim appears in Theorem B.2 [24]. Thus, we get $\|\mathcal{P}_r-\widehat{\mathcal{P}}_r\|_{\mathsf{HS}}\le\frac{4}{\tilde{\delta}_d}\sqrt{\frac{2\log(1/\epsilon)}{n}}$. Thus, for any $x\in\mathcal{X}$, we have

$$\|\mathcal{P}_r\kappa(\cdot,x)-\widehat{\mathcal{P}}_r\kappa(\cdot,x)\|_{\mathcal{H}}\le\|\mathcal{P}_r-\widehat{\mathcal{P}}_r\|_{\mathsf{HS}}\|\kappa(\cdot,x)\|_{\mathcal{H}}\le\frac{4}{\tilde{\delta}_d}\sqrt{\frac{2\log(1/\epsilon)}{n}}.$$

Recall that $\mathcal{P}_K K=U_K S_K U_K^{\mathsf{T}}$ and let $(U_K)_{:,r}$ the r^{th} column of U_K be the eigenvector of K corresponding to the r^{th} largest eigenvalue. Note that the corresponding eigenvalue $(S_K)_{rr}=n\widehat{\lambda}_r$ (the factor n appears because $\widehat{\lambda}_r$ is the eigenvalue of K/n). Then $K^{(r)}:=(U_K)_{:,r}(U_K)_{:,r}^{\mathsf{T}}K=n\widehat{\lambda}_r(U_K)_{:,r}(U_K)_{:,r}^{\mathsf{T}}$ denotes the projection of K on to the space corresponding to the r^{th} eigenvector. We have:

Lemma C.6. $K_{i,j}^{(\ell)}=\langle\widehat{\mathcal{P}}_\ell\kappa(\cdot,x_i),\widehat{\mathcal{P}}_\ell\kappa(\cdot,x_j)\rangle_{\mathcal{H}}$.

The proof can be found in, *e.g.,* Lemma 3.4 in [24]. For completeness, we repeat the arguments here.

Proof of Lemma C.6. Let $\Psi_{r,n}\in R^n$ be the vector whose entries are $\sqrt{\lambda_r}\psi_r(x_i)$ for $i\in[n]$. We have $K=\sum_{r\ge1}\Psi_{r,n}\Psi_{r,n}^{\mathsf{T}}$. Recall that $\widehat{u}^{(1)},\ldots,\widehat{u}^{(d)}$ are eigenvectors associated with the d largest

eigenvalues of K/n, we have

$$K^{(s)} = \sum_{r\geq 1} \widehat{u}^{(s)} \big(\widehat{u}^{(s)}\big)^{\mathsf{T}} \Psi_{r,n} \Psi_{r,n}^{\mathsf{T}} \widehat{u}^{(s)} \big(\widehat{u}^{(s)}\big)^{\mathsf{T}} \tag{27}$$

Thus, we have

$$K_{i,j}^{(s)} = \widehat{u}_i^{(s)} \big(\widehat{u}^{(s)}\big)^{\mathsf{T}} \Psi_{r,n} \Psi_{r,n}^{\mathsf{T}} \widehat{u}^{(s)} \big(\widehat{u}_j^{(s)}\big)^{\mathsf{T}}.$$

Recall that

$$\widehat{v}^{(i)}(\cdot) = \frac{1}{\sqrt{\widehat{\lambda}_i n}} \sum_{i=1}^{n} \kappa(\cdot, x_i) \widehat{u}_i.$$

We have for any $s \in [d]$:

$$\begin{aligned}
\langle \widehat{v}^{(s)}, \sqrt{\lambda_r}\psi_r \rangle_{\mathcal{H}} &= \left\langle \frac{1}{\sqrt{\widehat{\lambda}_s n}} \sum_{i=1}^{n} \kappa(\cdot, x_i) \widehat{u}_i^{(s)}, \sqrt{\lambda_r}\psi_r \right\rangle_{\mathcal{H}} \\
&= \left\langle \frac{1}{\sqrt{\widehat{\lambda}_s n}} \sum_{i=1}^{n} \sum_{r'\geq 1} \sqrt{\lambda_{r'}}\psi_{r'}(x_i) \sqrt{\lambda_{r'}}\psi_{r'} \widehat{u}^{(s)}, \sqrt{\lambda_r}\psi_r \right\rangle_{\mathcal{H}} \\
&= \frac{1}{\sqrt{\widehat{\lambda}_s n}} \sum_{i=1}^{n} \psi_r(x_i) \sqrt{\lambda_r} \widehat{u}_i^{(s)} \\
&= \frac{1}{\sqrt{\widehat{\lambda}_s n}} \langle \widehat{u}^{(s)}, \Psi_{r,n} \rangle_{R^n}.
\end{aligned}$$

This implies

$$\widehat{u}_i^{(s)} \big(\widehat{u}^{(s)}\big)^{\mathsf{T}} \Psi_{r,n} = \widehat{u}_i^{(s)} \langle \widehat{u}^{(s)}, \Psi_{r,n} \rangle_{R^n} = \widehat{v}^{(s)}(x_i) \langle \widehat{v}^{(s)}, \sqrt{\lambda_r}\psi_r \rangle_{\mathcal{H}}. \tag{28}$$

Next, let $\xi^{(s)}(x) = \sum_{r\geq 1} \langle \widehat{v}^{(s)}, \psi_r \sqrt{\lambda_r} \rangle_{\mathcal{H}} \widehat{v}^{(s)}(x) \sqrt{\lambda_r}\psi_r \in \mathcal{H}$. We have

$$K_{i,j}^{(\ell)} = \sum_{r\geq 1} \widehat{u}_i^{(\ell)} \big(\widehat{u}^{(\ell)}\big)^{\mathsf{T}} \Psi_{r,n} \Psi_{r,n}^{\mathsf{T}} \widehat{u}^{(\ell)} \widehat{u}_j^{(\ell)} = \langle \xi^{(\ell)}(x_i), \xi^{(\ell)}(x_j) \rangle_{\mathcal{H}} \tag{29}$$

We then use the reproducing kernel property of $\kappa(., x)$:

$$\begin{aligned}
\xi^{(\ell)}(s) &= \sum_{r=1}^{\infty} \langle \widehat{v}^{(s)}, \psi_r \sqrt{\lambda_r} \rangle_{\mathcal{H}} \widehat{v}^{(s)}(x) \sqrt{\lambda_r}\psi_r \\
&= \sum_{r=1}^{\infty} \langle \widehat{v}^{(s)}, \psi_r \sqrt{\lambda_r} \rangle_{\mathcal{H}} \langle \widehat{v}^{(s)}, \kappa(\cdot, x) \rangle_{\mathcal{H}} \sqrt{\lambda_r}\psi_r \\
&= \langle \widehat{v}^{(s)}, \kappa(\cdot, x) \rangle_{\mathcal{H}} \sum_{r=1}^{\infty} \langle \widehat{v}^{(s)}, \psi_r \sqrt{\lambda_r} \rangle_{\mathcal{H}} \sqrt{\lambda_r}\psi_r \\
&= \langle \widehat{v}^{(s)}, \kappa(\cdot, x) \rangle_{\mathcal{H}} \widehat{v}^{(s)}.
\end{aligned}$$

Finally, we have

$$\begin{aligned}
K_{i,j}^{(\ell)} &= \langle \xi^{(\ell)}(x_i), \xi^{(s)}(x_j) \rangle_{\mathcal{H}} \\
&= \langle \widehat{v}^{(\ell)}, \kappa(\cdot, x_i) \rangle_{\mathcal{H}} \langle \widehat{v}^{(\ell)}, \widehat{v}^{(\ell)} \rangle_{\mathcal{H}} \langle \widehat{v}^{(\ell)}, \kappa(\cdot, x_j) \rangle_{\mathcal{H}} \\
&= \left\langle \langle \widehat{v}^{(\ell)}, \kappa(\cdot, x_i) \rangle_{\mathcal{H}} \widehat{v}^{(\ell)}, \langle \widehat{v}^{(s)}, \kappa(\cdot, x_j) \rangle_{\mathcal{H}} \widehat{v}^{(\ell)} \right\rangle_{\mathcal{H}} \\
&= \langle \widehat{\mathcal{P}}_\ell \kappa(\cdot, x_i), \widehat{\mathcal{P}}_\ell \kappa(\cdot, x_j) \rangle_{\mathcal{H}}.
\end{aligned}$$

□

It then follows that, there exists a $w_r \in \{-1, 1\}$, such that $\sqrt{\widehat{\lambda}_r n}(U_K)_{:,r} w_r$ corresponds to an isometric isomorphism from the one dimensional subspace of the Hilbert space $\mathcal{H}$ under $\widehat{\mathcal{P}}_r$ to $\mathbb{R}$, for the datapoints $x_1, \ldots, x_n$. Similarly, the projection under operator $\mathcal{P}_r$ corresponds to the vector $\Phi_{:,r}(X) := \sqrt{\lambda}_r[\psi_r(x_1), \ldots, \psi_r(x_n)]^\mathsf{T}$. Thus, we have that: $\|(U_K)_{:,r}\sqrt{\widehat{\lambda}_r n} \cdot w_r - \Phi_{:,r}(X)\| \leq \frac{4}{\tilde{\delta}_d}\sqrt{2\log(1/\epsilon)}$ Applying the above to all $r \leq d$ and writing succinctly, we get:

$$\left\| U_K S_K^{1/2} W_K - \Phi_d(X) \right\|_F \leq \frac{4}{\tilde{\delta}_d}\sqrt{2d\log(1/\epsilon)}. \tag{30}$$

Above W_K is a diagonal orthogonal matrix, *i.e.,* every diagonal element is ± 1.

Using the fact that $S_K^{1/2}$, $S^{1/2}$ and W_K are all diagonal and hence commute and that $\|AB\|_F \leq \min\{\|A\|_F\|B\|, \|A\|\|B\|_F\}$, we get

$$\begin{aligned}\left\| \frac{U_K S_K W_K}{n} - \frac{\Phi_d(X) S^{1/2}}{\sqrt{n}} \right\|_F \leq & \left\| \frac{U_K S_K^{1/2} W_K}{\sqrt{n}} \right\|_F \cdot \left\| \frac{S_K^{1/2}}{\sqrt{n}} - S^{1/2} \right\| \\ & + \left\| \frac{U_K S_K^{1/2} W_K}{\sqrt{n}} - \frac{\Phi_d(X)}{\sqrt{n}} \right\|_F \cdot \left\| S^{1/2} \right\|.\end{aligned}$$

Next, we note that $\left\| \frac{U_K S_K^{1/2} W_K}{\sqrt{n}} \right\|_F \leq \|U_K\|_F \|S_K^{1/2}/\sqrt{n}\| \|W_K\| \leq \sqrt{d}$; as $\widehat{\lambda}_i = (S_K)_{ii} \leq 1$ for all i and W_K is an orthogonal matrix. Also, $\left\| \frac{S_K^{1/2}}{\sqrt{n}} - S^{1/2} \right\| \leq \max_{i \leq d} |\sqrt{\lambda_i} - \sqrt{\widehat{\lambda}_i}| \leq \max_{i \leq d} \sqrt{|\lambda_i - \widehat{\lambda}_i|}$, where λ_i and $\widehat{\lambda}_i$ are the i^{th} largest eigenvalues of the operators $\mathcal{K}_\mathcal{H}$ and $\mathcal{K}_n$, respectively. (We also use that for $a, b > 0$, $|\sqrt{a} - \sqrt{b}| \leq \sqrt{|a-b|}$). We know using Theorem B.2 from [24] that $\max_i |\lambda_i - \widehat{\lambda}_i| \leq 2\sqrt{\frac{2\log(1/\epsilon)}{n}}$ with probability at least $1 - 2\epsilon$. For the second term, we use (30) and the fact that $\|S^{1/2}\| \leq \max_{i \leq d} \lambda_i \leq 1$. Putting everything together and simplifying, we get that for some constant C,

$$\left\| \frac{U_K S_K W_K}{n} - \frac{\Phi_d(X) S^{1/2}}{\sqrt{n}} \right\|_F \leq C \left(\left(\frac{d^2 \log(1/\epsilon)}{n} \right)^{1/4} + \left(\frac{d\log(1/\epsilon)}{\tilde{\delta}_d^2 n} \right)^{1/2} \right) \tag{31}$$

We can prove the statement regarding M, by obtaining the equivalent of (30) for M. This completes the proof of the lemma. □

Part 2 of Proposition C.3. For part 2, we need the following lemma.

Lemma C.7. *Let $X, Y \in R^{m \times d}$, $\|X\|$ and $\|Y\|$ are bounded by c_1, and $\|X - Y\|_F \leq \epsilon$, we have*

$$\|XX^\mathsf{T} - YY^\mathsf{T}\|_F \leq 2c_1\epsilon. \tag{32}$$

Proof. We have

$$\begin{aligned}\|XX^\mathsf{T} - YY^\mathsf{T}\|_F &\leq \|XX^\mathsf{T} - XY^\mathsf{T} + XY^\mathsf{T} - YY^\mathsf{T}\|_F \\ &\leq \|X\|_2 \|X^\mathsf{T} - Y^\mathsf{T}\|_F + \|Y\|_2 \|X - Y\|_F \\ &\leq 2c_1\epsilon.\end{aligned}$$

□

Finally, using Lemmas C.4 and C.7 together finishes the proof of Prop. C.3.

C.3 Step 2. $A \propto M$.

We next show that the spectral norm of the difference between $\frac{n}{m}A$ and $\frac{1}{n}M$ is small by making use of suitable matrix tail inequalities.

Before we proceed, we comment on the reason we decided to bound the difference between A and M, instead of the difference between B and K. One commonly used approach to analyze directed graphs (*i.e.,* the B matrix) is to use standard Chernoff-type inequalities for matrices to bound $\|B - \mathbb{E}B\|$ via the "symmetrization" trick, *i.e.,* by considering the symmetric matrix $\begin{pmatrix} 0 & B \\ B^\mathsf{T} & 0 \end{pmatrix}$ [43, 27]. One drawback of this trick is that it requires both the (average) in- and out-degree to be in the order of $\Omega(\log n)$. This requirement is not satisfied in our model, because the average out-degree of followers is a constant. In fact, this is not the problem of the quality of Chernoff bound. Instead, $\|B - \mathbb{E}B\|$ could be large when B is tall and thin.

Example. $\|B - \mathbb{E}B\|$ **can be large.** Let us consider a simplified example where $B \in R^{m\times 1}$ and entries are independent r.v. from $\{-1, 1\}$. Each value appears with 0.5 probability. Note that $\mathbb{E}B = 0$. Thus, $\|B - \mathbb{E}B\|_2 = \|B\|_F = \sqrt{n}$ (which is considered to be large). Our product trick can address the issue directly: if we take the product of B, we notice that $\|B^\mathsf{T}B - \mathbb{E}[B^\mathsf{T}B]\| = 0$ (note that $B^\mathsf{T}B$ degenerates to a scalar) so the spectral gap between BB^T and $\mathbb{E}[B^\mathsf{T}B]$ is significantly reduced.

Example. Singular vectors of B do not converge to eigenvectors of M. Observe that the right singular vectors of B are the same as the eigenvectors of $B^\mathsf{T}B$. If we fix n and let $m \to \infty$, we have $\frac{1}{m}B^\mathsf{T}B \to \mathbb{E}B^\mathsf{T}B$. Because the diagonal of $B^\mathsf{T}B$ is not proportional to that of M, the eigenvectors of $\mathbb{E}B^\mathsf{T}B$ are different from those of M unless $\mathrm{diag}(\frac{n^2}{m}B^\mathsf{T}B) - \mathrm{diag}(M) \propto I$, which often is not true. Thus, we cannot use singular vectors of B to find $U_M S_M^{1/4}$, which is used to approximate $\Phi_d(\mathbf{X})$.

Our goal is to prove the following lemma.

Lemma C.8. *Using the notation defined above, if $m = \Omega(n \log^c n)$ for a suitably large constant c, we have with high probability*

$$\left\| \frac{n}{m}A - \frac{1}{n}M \right\| = O\left(\left(\frac{n}{m}\right)^{1/4} \right). \tag{33}$$

Proof. We let $Z_j = \sqrt{n}B_j$ (viewed as a column vector in $\mathbb{R}^n$), where B_j refers to the j^{th} row of B (*i.e.,* this vector encodes the connectivity of y_j). Note that Z_j are i.i.d. conditioned on knowing the latent variables $(x_i)_{i=1}^n$.

There are two sources of randomness for each Z_j: (1) Random selection of latent position of y_j drawn according to F, and (2) Random realizations of edges given x_i and y_j.

As the Z_j are identically distributed, we can calculate the expected behavior of $Z_1 Z_1^\mathsf{T}$. We have already seen that for $i \neq j$, $\mathbb{E}[(Z_1 Z_1^\mathsf{T})_{i,j}] = \frac{1}{n}\mu(x_i, x_j)$. We can also see that $\mathbb{E}[(Z_1 Z_1^\mathsf{T})_{i,i}] = \int \mathbb{E}[(Z_1)_i^2 \mid x_i] dF(y) = \int \kappa(y, x_i) dF(y) \neq \frac{1}{n}\mu(x_i, x_i)$.

We first a use triangle inequality to "decouple" diagonal entries from the off-diagonal entries. Let $\tilde{M} := M - \mathrm{diag}(M)$ be the matrix with M with diagonal entries set to 0. We have,

$$\left\| \frac{n}{m}A - \frac{1}{n}M \right\| \leq \left\| \frac{n}{m}A - \frac{1}{n}\tilde{M} \right\| + \frac{1}{n}\|\tilde{M} - M\| \tag{34}$$

The second term is straightforward to bound as $\frac{1}{n}(\tilde{M} - M)$ is a diagonal matrix, with each diagonal element of order $\frac{1}{n}$. Thus, we have $\frac{1}{n}\|\tilde{M} - M\| = O(1/n)$.

In order to bound the first term of (34), note that $\frac{n}{m}A = \frac{n}{m}B^\mathsf{T}B - \frac{n}{m}\mathrm{diag}(B^\mathsf{T}B)$ and $\frac{1}{n}\tilde{M} = \mathbb{E}[Z_1 Z_1^\mathsf{T}] - \mathrm{diag}(\mathbb{E}[Z_1 Z_1^\mathsf{T}])$. We have

$$\begin{aligned}\|(n/m)A - (1/n)\bar{M}\| \leq & \left\|(n/m)B^\mathsf{T}B - \mathbb{E}[Z_1 Z_1^\mathsf{T}]\right\| \\ & + \left\|(n/m)\mathrm{diag}(B^\mathsf{T}B) - \mathrm{diag}(\mathbb{E}[Z_1 Z_1^\mathsf{T}])\right\| \end{aligned} \tag{35}$$

We use a matrix inequality for sum of low rank matrices to bound the first term and a standard Chernoff bound for the second term.

First, let us bound the (easier) second term $\|(n/m)\mathrm{diag}(B^\mathsf{T}B) - \mathrm{diag}(\mathbb{E}[Z_1 Z_1^\mathsf{T}])\|$. Our crucial observation here is that $(B^\mathsf{T}B)_{i,i} = \sum_j B_{j,i}^2 = \sum_j B_{j,i}$. Thus each entry on the $\mathrm{diag}(B^\mathsf{T}B)$ is a sum of i.i.d. Bernoulli random variables. We note that $\mathbb{E}[B_{j,i}] = \Theta(1/n)$. We may thus apply a Chernoff bound on each element of the diagonal and a union bound over all n diagonal elements to get that there exists a constant c_2, such that:

$$\Pr\left[\left\|\mathrm{diag}(B^\mathsf{T}B)\frac{n}{m} - \mathrm{diag}\mathbb{E}[Z_1 Z_1^\mathsf{T}]\right\| \geq c_2\sqrt{\log(1/\epsilon)\cdot\frac{n}{m}}\right] \leq \epsilon \tag{36}$$

We use the following matrix inequality in Lemma C.9 [60] to bound $\left\|\frac{n}{m}B^\mathsf{T}B - \mathbb{E}[Z_1 Z_1^\mathsf{T}]\right\|$.

Lemma C.9. *[Lemma 1 in [61]] Let $Z_1, ..., Z_m$ be i.i.d. random column vectors in $\mathbb{R}^d$, with $|Z_i| \leq \alpha$ a.s. and $\|\mathbb{E}Z_i Z_i^\mathsf{T}\| \leq \beta$. Then we have for any $t \geq 0$:*

$$\Pr\left[\left\|\frac{1}{m}\sum_{i=1}^m Z_i Z_i^\mathsf{T} - \mathbb{E}[Z_1 Z_1^\mathsf{T}]\right\| \geq t\right] \leq (2m^2)\exp\left(-\frac{mt^2}{16\beta\alpha^2 + 8\alpha^2 t}\right) \tag{37}$$

We will apply the above lemma to obtain the required result. Observe that $\|Z_i\|^2 = n\sum_{j=1}^n B_{i,j}$, where $B_{j,i}$ are i.i.d. Bernoulli random variables and $\mathbb{E}[B_{j,i}] = \Theta(1/n)$. One can see that $|Z_i| \leq \log^2 n\sqrt{n}$ a.s. But as $n \to \infty$ both n and m grow simultaneous so a more careful analysis is needed.

Here, we do not directly work with Z_i. Instead, we couple Z_i with another group of bounded random variables so that we may use the matrix trail inequality directly. Specifically, we define the coupled process as follows.

1. Sample C_i from the distribution that's identical to $\sum_{j=1}^n B_{i,j}$. Then let $\tilde{C}_i = \min\{C_i, \zeta(n)\}$ and $H_i = I(C_i \geq \zeta(n))$.
2. Sample $\tilde{B}_i$ from the distribution $B_i|(|B_i|_1 = \tilde{C}_i)$ (interpreted as "sample B_i conditioned on knowing $|B_i|_1$"). and sample B_i from the distribution $B_i||B_i|_1 = C_i$.
3. Set $\tilde{Z}_i = \sqrt{n}\tilde{B}_i$ and $Z_i = \sqrt{n}B_i$.

Let us also set $R_i = Z_i - \tilde{Z}_i$ and $S = \mathbb{E}[Z_1 Z_1^\mathsf{T}] - \mathbb{E}[\tilde{Z}_1 \tilde{Z}_1^\mathsf{T}]$. One can see that $\{\tilde{Z}_i\}_{i\leq n}$ are independent and the statistical difference between $\tilde{Z}_i$ and Z_i is $O(\zeta(n))$ because $\Pr[\sum_{j=1}^n B_{i,j} > \zeta(n)] = O(\zeta(n))$. This also implies $S = n^{-\omega(1)}$.

Let t be a parameter to be decided later. We have

$$\begin{aligned} & \Pr\left[\left\|\frac{1}{m}\sum_{i=1}^m Z_i Z_i^\mathsf{T} - \mathbb{E}[Z_1 Z_1^\mathsf{T}]\right\| \geq t\right] \\ \leq \; & \Pr\left[\left\|\frac{1}{m}\sum_{i=1}^m \tilde{Z}_i \tilde{Z}_i^\mathsf{T} - \mathbb{E}[\tilde{Z}_1 \tilde{Z}_1^\mathsf{T}]\right\| \geq \frac{t}{2}\right] + \Pr\left[\left\|\frac{1}{m}\sum_{i=1}^m R_i - S\right\| \geq \frac{t}{2}\right] \end{aligned}$$

Using the fact that $S = n^{-\omega(1)}$, it is simple to bound the second term:

$$\begin{aligned}\Pr\left[\left\|\frac{1}{m}\sum_{i=1}^{m} R_i - S\right\| \geq \frac{t}{2}\right] &\leq \Pr\left[\left\|\frac{1}{m}\sum_{i=1}^{m} R_i\right\| \geq \frac{t}{4}\right] \\ &\leq \sum_{i=1}^{m}\Pr[R_i \neq 0] = \sum_{i=1}^{n}\mathbb{E}H_i = n^{-\omega(1)}.\end{aligned}$$

We next use Lemma C.9 to bound the first term. We have $|\tilde{Z}_i| \leq \sqrt{n\zeta(n)}$. Next, we observe that since for $i \neq j$, $(\mathbb{E}Z_1Z_1^{\mathsf{T}})_{i,j} = \frac{1}{n}\mu(x_i, x_j)$ and $(\mathbb{E}Z_1Z_1^{\mathsf{T}})_{i,i} = \int \kappa(y, x_i)dF(y) \leq \sup_y\{\kappa(y,x)\}$, $\|\mathbb{E}Z_1Z_1^{\mathsf{T}}\| = O\left(\sup_{x,x'}\mu(x,x') + \sup_{x,x'}\kappa(x,x')\right) = O(1)$. Thus, $\|\mathbb{E}\tilde{Z}_1\tilde{Z}_1^{\mathsf{T}}\| = O(1)$.

We set $t = \frac{c_1}{\left(\frac{m}{n}\right)^{1/4}}$. Assuming $m/n = \omega(\log^c n)$ and for c and c_1 chosen suitably, Lemma C.9 gives us,

$$\Pr\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\tilde{Z}_i\tilde{Z}_i^{\mathsf{T}} - \mathbb{E}[\tilde{Z}_1\tilde{Z}_1^{\mathsf{T}}]\right\| \geq \frac{t}{2}\right] = n^{-\omega(1)}.$$

Together with (36), this completes the proof of the lemma, we complete the proof of Lemma C.8.

□

Remark. Eigengaps for different operators. From $A \propto M$ and how we decide d in our algorithm, we can bound the eigengaps between different matrices/operators by using Theorem F.1 and Theorem C.5:

$$\delta_j\left((n/m)A\right) = \Theta\left(\delta_j\left(M/n\right)\right) = \Theta(\delta_j(\mathcal{M}) = \Theta(\delta_j(\mathcal{K})) = \Theta\left(\delta_j\left(K/n\right)\right). \tag{38}$$

The argument here is similar to the one presented in Lemma B.6. for all $j < d$, where $\delta_j(\cdot)$ is the eigengap between j-th and $j+1$-th eigenvalue of the matrix of interest. Furthermore, all the above gaps are $\Omega(\delta_d(\mathcal{K}))$ and $\bar{A}$ is positive definite.

We can also show that $\bar{A} \propto \bar{M}$. Specifically, let $\mathcal{P}_A = U_A U_A^{\mathsf{T}}$ and $\mathcal{P}_M = U_M U_M^{\mathsf{T}}$ be the corresponding projection operators. Note that $\bar{A} = \mathcal{P}_A A$ and $\bar{M} = \mathcal{P}_M M$. We have

Lemma C.10. *Let $\mathcal{P}_A$ and $\mathcal{P}_M$ be defined above. Then $\mathcal{P}_A A$ is positive definite. Furthermore, with high probability (over the randomness in matrix A),*

$$\left\|\frac{n}{m}\mathcal{P}_A A - \frac{1}{n}\mathcal{P}_M M\right\| = O\left(\frac{\left\|\frac{n}{m}A - \frac{1}{m}M\right\|}{\delta_d(M/n)}\right), \tag{39}$$

where $\delta_d(M/n) = \lambda_d(M/n) - \lambda_{d+1}(M/n)$.

Proof. First, we will show that $\mathcal{P}_A A$ is positive definite. Lemma C.8 gives a bound on $\left\|\frac{n}{m}A - \frac{1}{n}M\right\|$. Using Theorem II of [58], we know that if $\widehat{\lambda}_i$ and λ_i denote the i^{th} largest eigenvalues of $\frac{n}{m}A$ and $\frac{1}{n}M$, respectively, then

$$\max_i\left|\widehat{\lambda}_i - \lambda_i\right| \leq \left\|\frac{n}{m}A - \frac{1}{n}M\right\|. \tag{40}$$

The algorithm chooses d so that $\widehat{\lambda}_d - \widehat{\lambda}_{d+1} = \Omega\left(\left(\frac{n}{m}\right)^{c_2}\right)$ for some $c_2 \ll 1/4$. This together with the bound on $\left\|\frac{n}{m}A - \frac{1}{n}M\right\|$ given by Lemma C.8 and the fact that M is positive definite shows that the d largest eigenvalues of $\frac{n}{m}A$ are all positive and hence $\mathcal{P}_A A$ is positive definite.

Next, we show that $\|\mathcal{P}_A - \mathcal{P}_M\|$ can be suitably bounded by using the Davis-Kahan sin Θ theorem [38]. In particular, let $\eta = \left\|\frac{n}{m}A - \frac{1}{n}M\right\|$, $S_1 = \{\lambda \mid \lambda \geq \lambda_d - \eta\}$ and $S_2 = \{\lambda \mid \lambda < \lambda_{d+1} + \eta\}$. Let $\delta_d := \lambda_d - \lambda_{d+1}$. We know that $0 < \mathrm{dist}(S_1, S_2) \leq \delta_d - 2\eta$. Let $\mathcal{P}_A(S_1)$

and $\mathcal{P}_M(S_1)$ be the projection operations defined using eigenvectors with eigenvalues in S_1 of the matrices $\frac{n}{m}A$ and $\frac{1}{n}M$, respectively. By our choice of parameters, we see that $\mathcal{P}_A(S_1) = \mathcal{P}_A$ and $\mathcal{P}_M(S_1) = \mathcal{P}_M$. Thus, by using the Davis-Kahan sin Θ theorem, we get

$$\|\mathcal{P}_A - \mathcal{P}_M\| \leq \frac{\left\|\frac{n}{m}A - \frac{1}{n}M\right\|}{\mathrm{dist}(S_1, S_2)} = O\left(\frac{\eta}{\delta_d}\right) \tag{41}$$

Finally, we observe that,

$$\begin{aligned}\left\|\mathcal{P}_A\frac{n}{m}A - \mathcal{P}_M\frac{1}{n}M\right\| &\leq \left\|\mathcal{P}_A\left(\frac{n}{m}A - \frac{1}{n}M\right)\right\| + \left\|\left(\mathcal{P}_A - \mathcal{P}_M\right)\frac{1}{n}M\right\| \\ &\leq \left\|\frac{n}{m}A - \frac{1}{n}M\right\| + O\left(\frac{\eta}{\delta_d}\right).\end{aligned}$$

Above, we used the fact that $\|\mathcal{P}_A\| \leq 1$ and that $\left\|\frac{1}{n}M\right\| = O(1)$. □

Step 3: $\bar{A}^{1/2} \propto \bar{K}$. Let $\eta = \|(n/m)A - M/n\|$. From (38), (39), and Proposition C.3, we can get $\|(n/m)\bar{A} - \bar{K}^2/n^2\| = O(\eta/\delta_d(\mathcal{M}))$. Next, we need to show that the square root of $\bar{A}$ and those of $\bar{K}^2$ will be close, *i.e.,*

Lemma C.11. *Let $\eta = \left\|\frac{n}{m}A - \frac{1}{n}M\right\|$. Let $\bar{A}^{1/2}$ and $\bar{K}$ be defined as above. We have*

$$\|\sqrt{\frac{n}{m}}\bar{A}^{1/2} - \frac{1}{n}\bar{K}\| \leq O(\sqrt{\eta/\delta_d(\mathcal{M})} + \sqrt{d}\eta/\delta_d^2(\mathcal{M})) \tag{42}$$

Here, $\bar{A}^{1/2} = U_A S_A^{1/2} U_A^{\mathsf{T}}$. Our techniques for proving the above lemma are similar to those used in Step 1. Specifically, we show that the *pairwise* eigenvectors of $\bar{A}$ and that of $\bar{K}$ are close. Thus, after linearly scaling these two set of vectors by using (approximation of) $S^{-1/4}$ will result in two set of vectors that are still close.

Proof. We need to argue that each eigenvector of $\bar{A}$ is close to that of K^2 (up to a sign difference). But this time we need to handle matrices, rather than linear operators, so we can use the original Davis-Kahan theorem.

We shall also set that $\eta > 10\delta_d^2$ (η grows with m and it is in $\mathrm{polylog}n$ scale; 10 is an arbitrarily large constant). Let $(U_A)_{:,i}$ and $(U_K)_{:,i}$ be the i-th column of U_A and U_K, respectively. By the Davis-Kahan theorem [38] and (38), we have

$$|\sin\Theta((U_A)_{,i}, (U_K)_{,i})| \leq \frac{\|\frac{n}{m}\bar{A} - \frac{1}{n^2}\bar{K}^2\|}{\Theta(\delta_d(\mathcal{M}))} = \Theta(\eta/\delta_d^2(\mathcal{M})). \tag{43}$$

Thus, there exists an $w \in \{\pm 1\}$ such that

$$\|(U_A)_{:,i}w - (U_K)_{:,i}\| \leq |\sin((U_A)_{:,i}, (U_K)_{:,i})| = O(\eta/\delta_d^2(\mathcal{M})). \tag{44}$$

Therefore,

$$\|U_A W - U_K\|_F = O(\sqrt{d}\eta/\delta_d^2(\mathcal{M})), \tag{45}$$

where W is a diagonal matrix so that each diagonal entry is in $\{\pm 1\}$, and recall that η is an upper bound of $\|(n/m)A - (1/n)M\|$.

Now we can move to bound $\|\sqrt{\frac{n}{m}}\bar{A}^{1/2} - \frac{1}{n}\bar{K}\|$. Let $E = U_A W - U_K$, *i.e.,* $U_A = (U_K + E)W^\mathsf{T}$. We have

$$\begin{aligned}
& \left\|\sqrt{\frac{n}{m}}\bar{A}^{1/2} - \frac{1}{n}\bar{K}\right\| \\
= & \left\|(U_K+E)W^\mathsf{T}\left(\sqrt{\frac{n}{m}}S_A^{1/2}\right)(U_K+E)^\mathsf{T} - U_K\left(\frac{S_K}{n}\right)U_K^\mathsf{T}\right\| \\
\leq & \left\|U_K\left(W^\mathsf{T}\sqrt{\frac{n}{m}}S_A^{1/2}W - \frac{S_K}{n}\right)U_K^\mathsf{T}\right\| + \left\|EW^\mathsf{T}\sqrt{\frac{n}{m}}S_A^{1/2}W\right\| \\
& \quad + \|W\left(\sqrt{\frac{n}{m}}S_A^{1/2}\right)E^\mathsf{T}\| + \left\|EW^\mathsf{T}\sqrt{\frac{n}{m}}S_A^{1/2}WE^\mathsf{T}\right\| \\
= & \left\|U_K\left(\sqrt{\frac{n}{m}}S_A^{1/2} - \frac{S_K}{n}\right)U_K^\mathsf{T}\right\| + O(\|E\|) \\
\leq & \left\|\left(\sqrt{\frac{n}{m}}S_A^{1/2} - \frac{S_K}{n}\right)\right\| + O(\|E\|).
\end{aligned}$$

Note first for the j-th eigenvalue of $(n/m)A$, namely $\widehat{\lambda}_j$ and the j-th eigenvalue of $\frac{1}{n^2}K$, namely λ_j, we have $\max_j|\lambda_j - \widehat{\lambda}_j| \leq \left\|\frac{n}{m}\bar{A} - \frac{1}{n^2}\bar{K}\right\|$. See Theorem F.1. Using a similar trick developed in Step 1 in Section C, we can bound $\max_j|\lambda_j^{1/2} - \widehat{\lambda}_j^{1/2}| \leq \left\|\frac{n}{m}\bar{A} - \frac{1}{n^2}\bar{K}\right\|^{1/2}$. Thus, we have

$$\|\sqrt{\frac{n}{m}}\bar{A}^{1/2} - \frac{1}{n}\bar{K}\| \leq O(\sqrt{\eta/\delta_d(\mathcal{M})} + \sqrt{d}\eta/\delta_d^2(\mathcal{M})).$$

□

Finally, putting together Lemma C.11 and $U_K S_K^{1/2} W \approx \Phi_d$ (From Eq.(21)), one can see that $U_A S_A^{1/4} \propto \Phi_d$, *i.e.,*

Proposition C.12. *Let $\widehat{\Phi}_d = \frac{n^{3/4}}{m^{1/4}}U_A S_A^{1/4}$. Then there exists some orthogonal matrix $\tilde{W}$ such that whp*

$$\left\|\widehat{\Phi}_d - \Phi_d\right\|_F \leq O\left(\frac{\sqrt{dn}(\sqrt{\eta/\delta_d(\mathcal{M})} + \sqrt{dn}\eta/\delta_d^2(\mathcal{M}))}{\delta_d(\mathcal{M})}\right), \tag{46}$$

where $\eta = \|(n/m)A - M/n\| \leq c\log^{1/2}(1/\epsilon)\left(\frac{n}{m}\right)^{1/4}$

Proof. We prove the proposition via using a triangle inequality through K, *i.e.,* we need to show that $U_A S_A^{1/4} \propto U_K S_K^{1/2}$ and $U_K S_K^{1/2} \propto \Phi_d$. As discussed before, the latter part is a known result in kernel PCA [24], *i.e.,* there exists a rotation matrix so that with probability $\geq 1 - 2\epsilon$:

$$\|U_K S_K^{1/2} W - \Phi_d\|_F \leq 2\sqrt{2}\frac{\sqrt{\log(1/\epsilon)}}{\delta_d(\mathcal{K})}. \tag{47}$$

Thus, we need only understand the relationship between $U_A S^{1/4}$ and $U_K S_K^{1/2}$. Recall from Lemma C.11 that

$$\left\|\sqrt{\frac{n}{m}}\bar{A}^{1/2} - \frac{\bar{K}}{n}\right\| = O\left(\sqrt{\frac{\eta}{\delta_d(\mathcal{M})}} + \frac{\sqrt{d}\eta}{\delta_d^2(\mathcal{M})}\right). \tag{48}$$

Next, we need to show that the "square root" of $\bar{A}$ is close to the "square root" of $\bar{K}$. We leverage Lemma B.7 appeared before (Lemma A.1 from [24]).

Matrices $\sqrt{n/m}\bar{A}^{1/2}$ and $\bar{K}/n$ are the matrices A and B for Lemma B.7. Note that both of our matrices have constant operator norm so there exists a rotational matrix such that

$$\left\|\left(\frac{n}{m}\right)^{1/4}U_A S_A^{1/4}W - \frac{1}{\sqrt{n}}U_K S_K^{1/2}\right\|_F = O\left(\frac{\sqrt{d}(\sqrt{\eta/\delta_d(\mathcal{M})} + \sqrt{d}\eta/\delta_d(\mathcal{M}))}{\delta_d(\mathcal{M})}\right). \tag{49}$$

By properly scaling up the above inequality (multiplying both sides by a factor of $\sqrt{n}$) and using (47), we know that
$\left\| \frac{n^{3/4}}{m^{1/4}} U_A S_A^{1/4} W - \frac{1}{\sqrt{n}} U_K S_K^{1/2} \right\|_F$ is the dominating term, and this completes the proof of the proposition.

□

Step 4. Truncation error $\|\Phi_d - \Phi\|_F$. We shall use the same argument presented in Section B to bound $\|\widehat{\Phi}_d - \Phi\|_F$, *i.e.,* if the decay condition holds, then $\|\Phi_d - \Phi\|_F = \sqrt{n}\delta_d^{1/6}$ and $d = \delta_d^{1/4}$. Thus, we may set $\delta_d = (n/m)^{2/43}$. Then we have whp $\|\widehat{\Phi} - \Phi\|_F = O(\sqrt{n}(n/m)^{2/43} \log n)$, which completes our proof for Proposition C.1. Here we also make an additional assumption that the eigengaps $\delta_d = \lambda_d(\mathcal{K}) - \lambda_{d+1}$ monotonically decreases whenever the number of non-zero eigenvalues is infinite. Recall that Section D presents an analysis without the assumption.

D More Refined Truncation Error Analysis

Let $\lambda_1, ..., \lambda_{N_\mathcal{H}}$ be the eigenvalues of $\mathcal{K}$. This section analyzes the truncation error without the assumption the gap $\delta_d = \lambda_d(\mathcal{K}) - \lambda_{d+1}(\mathcal{K})$ is monotonically decreasing. Specifically, the following proposition suffices to prove Theorem 2.2 (the constants in the theorem will become worse).

Proposition D.1. *Let $\mathcal{K}$ be a linear operator such that $\lambda_i(\mathcal{K}) = O(1/i^c)$ for some constant $c > 2.5$. Let $\delta = \lambda_d - \lambda_{d+1}$ be given. Then we can express $\sum_{i \geq d+1} \lambda_i$ and d in terms of δ. Specifically,*

$$\begin{array}{rl} \sum_{i \geq d+1} \lambda_i & = \mathrm{poly}(\delta) \\ d & = \mathrm{poly}(1/\delta) \end{array} \tag{50}$$

We need the following lemma.

Lemma D.2. *Let d be a sufficiently large number. There exists an i^* such that:*

1. $\sum_{i \geq i^*} \lambda_i \leq c_1/(2d^{0.1})$.
2. $\delta_{i^*} \geq c_2/d^{2.1}$.

Here, c_1 and c_2 are constants that are independent of d.

The constants 0.1 and 2.1 are chosen arbitrarily. We do not attempt to optimize them.

Proof of Lemma D.2. Since $\lambda_d = O(d^{-c})$, there exists a constant c_0 such that $\lambda_d \leq c_0/d^c$ for all d. We let $\mathrm{tail}(\ell) \triangleq \sum_{i \geq \ell} c_0/i^c = c_1/\ell^{c-1}$ for some constant c_1. Next, we define i_1 and i_2:

$$i_1 = \max_{i_1} \left\{ \sum_{i \leq i_1} \lambda_i \leq 1 - \mathrm{tail}(d^{\frac{0.1}{c-1}}) \right\}. \tag{51}$$

$$i_2 = \max_{i_2} \left\{ \sum_{i \leq i_2} \lambda_i \leq 1 - 0.5 \times \mathrm{tail}(d^{\frac{0.1}{c-1}}) \right\}. \tag{52}$$

We know that $i_2 \leq i_1 \leq d$. Furthermore,

$$\sum_{i \leq i_2 + 1} \lambda_i \geq 1 - \frac{c_1}{2} d^{-\frac{0.1}{c-1} \cdot (c-1)} \geq 1 - \frac{c_1}{2} d^{-0.1}.$$

$$\sum_{i_2 < i \leq i_1} \lambda_i \geq 0.5 \cdot \mathrm{tail}(d^{\frac{0.1}{c-1}}) = 0.5 c_1 / d^{0.1}.$$

By using an averaging argument, there exists an $i_3 \in [i_2 + 1, i_1]$ such that $\lambda_{i_3} \geq c_1/(2d^{1.1})$. On the other hand, we have $\lambda_d \leq c_0/d^{c-1}$. We have that there exists an $i^* \in [i_3, d]$ such that

$\delta_{i^*} = \Theta(1/d^{2.1})$. On the other hand,

$$\sum_{i \geq i^*} \lambda_i \leq \sum_{i \geq i_2} \lambda_i \leq c_1/(2d^{0.1}).$$

This completes the proof. □

Proof of Proposition D.1. Let $\bar{d}$ be that $\delta = c_2/(\bar{d})^{2.1}$, *i.e.,* $\bar{d} = (c_2/\delta)^{1/2.1}$. Using Lemma D.2, we can find an i^* such that

1. $\delta_{i^*} \geq c_2/(\bar{d})^{2.1} = \delta$
2. $\sum_{i \geq i^*} \lambda_i \leq c_1/(2\bar{d}^{0.1}) = \Theta(\delta^{1/21})$.

Because $\delta_{i^*} > \delta$, $d \geq i^*$. Thus, $\sum_{i \geq d} \lambda_i \leq \sum_{i \geq i^*} \lambda_i = O(\delta^{1/21})$. Next, we need to show that d is $\mathrm{poly}(1/\delta)$. Note that $d \geq i^*$ and $\lambda_d \geq \delta$, we have

$$\Theta(\delta^{1/21}) \geq \sum_{i^* \leq i \leq d} \lambda_i \geq d\delta.$$

Thus, $d \leq \delta^{-20/21} = \mathrm{poly}(1/\delta)$. □

E Analysis for the isomap-based algorithm

This section analyzes the isomap-based algorithm.

Recall of the notations. $x_1, x_2, ..., x_n$ are the latent variables. Also, $z_i = \Phi(x_i)$ and $\widehat{z}_i = \widehat{\Phi}(x_i)$. For any $z \in R^{N_{\mathcal{H}}}$ and $r > 0$, we let $\mathbf{Ball}(z, r) = \{z' : \|z' - z\| \leq r\}$. Define projection $\mathrm{Proj}(z) = \arg\min_{z' \in \mathcal{C}} \|z' - z\|$. Finally, for any point $z \in \mathcal{C}$, define $\Phi^{-1}(z)$ be that $\Phi(\Phi^{-1}(z)) = z$ (*i.e.,* z's original latent position). For points that are outside $\mathcal{C}$, define $\Phi^{-1}(z) = \Phi^{-1}(\mathrm{Proj}(z))$.

Outline. We first describe the fundamental building block for our analysis. Then we analyze the performance of the denoising procedure 3. Finally, we give an analysis for the full isomap algorithm.

E.1 Fundamental building blocks

Lemma E.1. *Let $x, x' \in [0,1]$. Let also $g(n)$ and $h(n)$ be two diminishing functions (*i.e., *$g(n), h(n) = o(1)$). We have*

(1) If $|x - x'| = h(n)$, then $\|\Phi(x) - \Phi(x')\| = \sqrt{\frac{2}{c}} h^{\Delta/2}(n) + o(h^{\Delta/2}(n))$.

(2) If $\|\Phi(x) - \Phi(x')\| = g(n)$, then $|x - x'| = \left(\frac{c}{2}\right)^{1/\Delta} g^{2/\Delta}(n)$.

Proof of Lemma E.1. For part (1), we have

$$\begin{aligned}
\|\Phi(x) - \Phi(x')\|^2 &= \kappa(x,x) + \kappa(x',x') - 2\kappa(x,x') \\
&= 2 - 2\frac{c}{c(1 + |x - x'|^{\Delta}/c)} \\
&= 2 - 2(1 - |x - x'|^{\Delta}/c) + o(|x - x'|^{\Delta}) \\
&= \frac{2}{c}|x - x'|^{\Delta} + o(|x - x'|^{\Delta}).
\end{aligned}$$

Thus, $\|\Phi(x) - \Phi(x')\| = \sqrt{\frac{2}{c}} h^{\Delta/2}(n) + o(h^{\Delta/2}(n))$.

We may then use part 1 to prove part 2 in a straightforward manner. □

Lemma E.2. *Let $x \in [0,1]$ and $z = \Phi(x)$. Let $z' \in R^{N_{\mathcal{H}}}$ be a point such that $\|z' - z\| \leq g(n)$, then $\|\mathrm{Proj}(z') - z\| \leq 2g(n)$ and $|x - \Phi^{-1}(z')| \leq \left(\frac{c}{2}\right)^{1/\Delta} (2g(n))^{2/\Delta} = (2c)^{1/\Delta} g^{2/\Delta}(n)$.*

Proof of Lemma E.2. Since $\|z'-z\| \le g(n)$, we have $\|\mathrm{Proj}(z')-z'\| \le g(n)$. Then using a triangle inequality, we have $\|\mathrm{Proj}(z') - z\| \le \|\mathrm{Proj}(z') - z'\| + \|z' - z\| \le 2g(n)$. Then by Lemma E.1, we may also prove the second part of the lemma. □

E.2 Analysis of the denoising procedure

Also, recall that we classify a point i into three groups: **1. Good:** when $\|\widehat{z}_i - \mathrm{Proj}(\widehat{z}_i)\| \le 1/\sqrt{f(n)}$. We may further partition the set of good points into two parts. **Good-I:** those points so that $\|\widehat{z}_i - z_i\| \le 1/\sqrt{f(n)}$. **Good-II:** those points that are good but not in Good-I. **2. Bad:** when $\|z_i - \mathrm{Proj}(z_i)\| > 4/\sqrt{f(n)}$. **3. Unclear:** otherwise. We have (see Appendix E for a proof)

We use the following decision rule to

$$\textsc{Denoise}(\widehat{z}_i) : \text{If } |\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})| < n/f(n), \text{ remove } \widehat{z}_i. \tag{53}$$

We want to prove the following lemma.

Lemma E.3. *[Repeat of Lemma 3.2] After running* Denoise*, Using the counting-based decision rule, all the good points are kept, all the bad points are eliminated, and the unclear points have no performance guarantee. The total number of eliminated nodes is* $\le n/f(n)$.

Proof of Lemma 3.2. We have the follow three facts.

Fact E.1. *Let* $\widehat{z}_i$ *be a good point. For any good point* j *such that* $\|\mathrm{Proj}(\widehat{z}_i)-\mathrm{Proj}(\widehat{z}_j)\| \le 1/\sqrt{f(n)}$*, we have* $\|\widehat{z}_i - \widehat{z}_j\| \le 3/\sqrt{f(n)}$.

This can be shown via a simple triangle inequality:

$$\|\widehat{z}_j - \widehat{z}_i\| \le \|\widehat{z}_j - \mathrm{Proj}(\widehat{z}_j)\| + \|\mathrm{Proj}(\widehat{z}_j) - \mathrm{Proj}(\widehat{z}_i)\| + \|\mathrm{Proj}(\widehat{z}_i) - \widehat{z}_i\| \le \frac{3}{\sqrt{f(n)}}. \tag{54}$$

Fact E.2. *Let* $\widehat{z}_i$ *be a good point and consider* $\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})$*. The total number of points that are within the ball is at least* $n(c_0/3f(n))^{1/\Delta}$ *for some constant* c_0.

Proof. We need to show that (1) there are a sufficient number of Good-I nodes that are near $\mathrm{Proj}(\widehat{z}_i)$, and (2) these points are in $\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})$. Note that when we have a Good-I z_j such that $\|z_j - \mathrm{Proj}(\widehat{z}_i)\| \le 1/\sqrt{f(n)}$, we have $|x_j - \Phi^{-1}(\widehat{z}_i)| \le (c/2)^{1/\Delta}(1/\sqrt{f(n)})^{2/\Delta} = (c/(2f(n)))^{1/\Delta}$. By the near-uniform density assumption, we have the total number of nodes x_j that within the distance of $(c/(2f(n)))^{1/\Delta}$ is at least $n(c_0/(2f(n)))^{1/\Delta}$ for some constant c_0. Note that the number of non-Good-I nodes is at most $n/f(n)$. Thus, the total number of Good-I nodes here is

$$n\left(\left(\frac{c_0}{2f(n)}\right)^{1/\Delta} - \frac{1}{f(n)}\right) \le n\left(c_0/(3f(n))\right)^{1/\Delta}.$$

Finally, we have

$$\|\widehat{z}_j - \widehat{z}_i\| \le \|\widehat{z}_j - z_j\| + \|z_j - \mathrm{Proj}(\widehat{z}_i)\| + \|\mathrm{Proj}(\widehat{z}_i) - \widehat{z}_i\| \le 3/\sqrt{f(n)}.$$

Therefore, all these nodes are in $\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})$. □

Fact E.3. *For all the bad points* $\widehat{z}_i$*, the ball* $\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})$ *does not cover more than* $n/f(n)$ *nodes.*

Proof. For any node $\widehat{z}_j$ in $\mathbf{Ball}(\widehat{z}_i, 3/\sqrt{f(n)})$, we have $\|\widehat{z}_j - \mathrm{Proj}(\widehat{z}_j)\| > 1/\sqrt{f(n)}$. Otherwise,

$$\|\widehat{z}_i - \mathrm{Proj}(\widehat{z}_i)\| \le \|\widehat{z}_i - \mathrm{Proj}(\widehat{z}_j)\| \le \|\widehat{z}_i - \widehat{z}_j\| + \|\widehat{z}_j - \mathrm{Proj}(\widehat{z}_j)\| \le 4/\sqrt{f(n)}.$$

This contradicts to that $\widehat{z}_i$ is bad. But $\|\widehat{z}_j - \mathrm{Proj}(\widehat{z}_j)\| > 1/\sqrt{f(n)}$ implies $\|\widehat{z}_j - z_j\| > 1/\sqrt{f(n)}$ so the number of such j is upper bounded by $n/f(n)$. □

□

E.3 The performance of isomap algorithm

This section proves the following proposition .

Proposition E.4. *The length of the path connecting between i and j has the following bounds:*

$$(d-1)\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta} \leq |x_i - x_j| \leq d\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell+8}{\sqrt{f(n)}}\right)^{2/\Delta} \tag{55}$$

Before proceeding, we remark that our final distance estimate should be $(d-1)(c/2)^{1/\Delta}((\ell-3)/\sqrt{f(n)})^{2/\Delta}$ when the distance between two nodes is d on the graph. The multiplicative error ratio is $((\ell+8)/(\ell+3))^{2/\Delta}$ and the additive error is $\left(\frac{c}{2}\right)^{1/\Delta}\left((\ell+8)(\sqrt{f(n)})\right)^{2/\Delta}$. This implies Theorem 2.2.

Proof. The analysis consists of two parts. First we give a lower bound on d, *i.e.,* d is not too small. Then we give an upper bound.

1. d **is not too small.** We want to bound the latent distance. For an arbitrary consecutive pair of nodes i_j and i_{j+1}, they are not bad nodes so we have

$$\begin{aligned}\|\mathrm{Proj}(\widehat{z}_{i_j}) - \mathrm{Proj}(\widehat{z}_{i_{j+1}})\| &\leq \|\mathrm{Proj}(\widehat{z}_{i_j}) - \widehat{z}_{i_j}\| + \|\mathrm{Proj}(\widehat{z}_{i_{j+1}}) - \widehat{z}_{i_{j+1}}\| + \|\widehat{z}_{i_j} - \widehat{z}_{i_{j+1}}\| \\ &\leq \frac{\ell + 2\times 4}{\sqrt{f(n)}}.\end{aligned}$$

Then by Lemma E.2, we have

$$\|\Phi^{-1}(\widehat{z}_{i_j}) - \Phi^{-1}(\widehat{z}_{i_{j+1}})\| \leq \left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell+2\times 4}{\sqrt{f(n)}}\right)^{2/\Delta}.$$

Thus, we have

Lemma E.5. *Let d be the shortest path distance between i and j on the graph built form isomap. The latent distance between x_i and x_j is at most $d\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell+8}{\sqrt{f(n)}}\right)^{2/\Delta}$.*

2. d **is not too large.** We next give a constructive proof for an upper bound of d.

Lemma E.6. *Using the notations above, we can find a path $i, i_1, ..., i_{d-1}, j$ such that:*

1. *All these nodes are Good-I nodes.*
2. *The corresponding latent variables are monotonically increasing or decreasing, and the distance (in feature space) between two consecutive nodes is at least $(\frac{c}{2})^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta}$.*
3. *The distance between $\widehat{z}_{i_j}$ and $\widehat{z}_{i_{j+1}}$ (for any j) is within $\ell/\sqrt{f(n)}$.*

If all the above claims were true, then we know that the path "makes good progress" in every step, *i.e.,* when we move from i_j to i_{j-1}, in the latent space, we are $(\frac{c}{2})^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta}$ closer to the destination j. This implies:

Corollary E.7. *The length of the path connecting between i and j has the following upper bound:*

$$(d-1)(c/2)^{1/\Delta}((\ell-3)/\sqrt{f(n)})^{2/\Delta} \leq |x_i - x_j| \tag{56}$$

Corollary E.7 and Lemma E.5 implies Proposition E.4. □

We now proceed to prove Lemma E.5.

Proof of Lemma E.5. Let us start with considering an arbitrary interval $I \in [0,1]$ in the latent space of size $\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta} - \left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-2}{\sqrt{f(n)}}\right)^{2/\Delta} = \left(\frac{c}{2f(n)}\right)^{1/\Delta}((\ell-3)^{2/\Delta} - (\ell-2)^{2/\Delta})$.

The expected number of nodes in this interval is $(c_0/2)^{1/\Delta} n/f^{1/\Delta}(n)$ for some constant c_0, and we have a concentration bound, *i.e.,* with exponentially small probability that the number of nodes is $\geq \frac{1}{2}\left(\frac{c}{+}02\right)^{1/\Delta} n/f^{1/\Delta}(n)$. On the other hand, out of these nodes only $n/f(n)$ are not Good-I nodes, so there are $\Theta((n/2)^{1/\Delta} n/f^{1/\Delta}(n))$ Good-I nodes in I.

Then we may construct the sequence $i_1, ..., i_{d-1}$ using this property. Wlog, let $x_i < x_j$. We let

$$I_1 = \left[x_i + \left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta}, x_i + \left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-2}{\sqrt{f(n)}}\right)^{2/\Delta}\right].$$

Let i_1 be an arbitrary Good-I node in I_1. We can also recursively define

$$I_j = \left[x_{i_j} + \left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-3}{\sqrt{f(n)}}\right)^{2/\Delta}, x_{i_j} + \left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-2}{\sqrt{f(n)}}\right)^{2/\Delta}\right].$$

Then we can find a Good-I i_j in I_j.

By construction, property 1 and 2 hold. Now we need only verify property (3). Since all the nodes are Good-I, $\|\widehat{z}_{i_j} - z_{i_j}\| \leq \frac{1}{\sqrt{f(n)}}$. Using Lemma E.1, we also know that

$$\|z_{i_j} - z_{i_{j+1}}\| \leq \sqrt{\frac{2}{c}}\left(\left(\frac{c}{2}\right)^{1/\Delta}\left(\frac{\ell-2}{\sqrt{f(n)}}\right)^{2/\Delta}\right)^{\Delta/2} \leq \frac{\ell-2}{\sqrt{f(n)}}. \tag{57}$$

By using a triangle inequality, $\|\widehat{z}_{i_j} - \widehat{z}_{i_{j+1}}\| \leq \ell/\sqrt{f(n)}$. □

F Spectral properties of linear operators and graphs

This section presents prior spectral results on linear operators or graphs that are used in our analysis.

Theorem F.1 (Simplified from [58])**.** *Let A and B be self-adjoint operators in H such that $B = A + C$, where C is a compact self-adjoint operator. Let $\{\gamma_k\}$ be an enumeration of the non-zero eigenvalues of C. Then there exist extended enumerations $\{\alpha_j\}, \{\beta_j\}$ of discrete eigenvalues for A, B, respectively, such that the following inequality holds:*

$$\left(\sum_{j\geq 1}|\alpha_i - \beta_i|^p\right)^{1/p} \leq \left(\sum_j |\gamma_k|^p\right)^{1/p}, \quad 1 \leq p \leq \infty. \tag{58}$$

F.1 Distribution of the eigenvalues

Theorem F.2 (Weyl [62], or see [63])**.** *Let κ, F, and $\mathcal{K}$ be defined as above. If F is uniform, $\kappa(x,y) = \kappa(y,x)$, and $\partial^v(x,y)/\partial^v x$ exists and is continuous, then $\lambda_n(\mathcal{K}) = o(n^{-v-1/2})$.*

When $\mathcal{D}$ is not uniform, we also have the following Proposition [46].

Proposition F.3. *Let $2 \leq p < \infty$. $\mathcal{X}$ be a Banach space and $\mathcal{T} \in L(\mathcal{X})$ be $(p,2)$-summing (*i.e., *$\left(\sum_{i\geq 1}\lambda_i^p(\mathcal{T})\right)^{1/2}$ exists), then $\lambda_n(\mathcal{T}) = O(n^{1/p})$.*

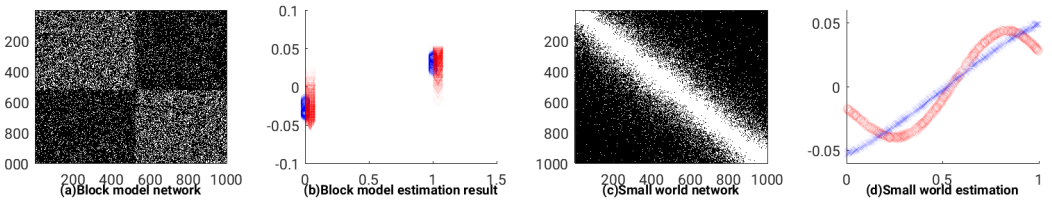


Figure 9: (a) and (b): recover the latent structure of a graph from SBM. (a) is the heat map of the observed graph. (b) is a scatter plot between estimated and true distance. An algorithm that exactly recovers latent variables will output two dots, each of which corresponds to a community; a low quality algorithm will output two intervals that significantly overlap with each other. The blue curve corresponds to our algorithm. The red curve corresponds to a simplified small-world algorithm from [16]. (c) and (d) recover the latent structure of a small-world graph. (c) is the heat map of the observed data. (d) is also a scatter plot between estimated and true distances. High quality algorithms output straight lines. Our algorithm corresponds to the blue curve while Newman's spectral algorithm corresponds to the red curve [55].

One can see that if $\mathcal{A}$ and $\mathcal{B}$ are, for example, $(4, 2)$-summing, then $(\mathcal{A} + \mathcal{B})$ is also $(4, 2)$-summing. Together with the fact that when $\mathcal{D}$ is uniform, $\mathcal{K}$ is $(4, 2)$-summing, we see that when $\mathcal{D}$ is a mixture of uniform distribution (*i.e.,* $\mathcal{D}$ is piecewise constant), our decay assumption holds.

G Experiments

In this setion, we describe the experiments used to validate the necessity of using specral and isomap techniques, and the efficacy of the new regularization technique on the product $B^\mathsf{T}B$. We compare our algorithms against baselines while noting that a comprehensive evaluation is beyond the scope of this paper.

G.1 Synthetic data

We use synthetic data to carry out a "sanity check", *i.e.,* an SBM inference algorithm does not perform well in a SWM graph and vice versa.

We evaluate our algorithm against two algorithms optimized for SBM and SWM, respectively: (simplified) Abraham et al.'s algorithm [16] for small-world and Newman's spectral-based modularity algorithm [55]. Abraham et al.'s algorithm is the only known algorithm with provable guarantee for SWM. Modularity algorithm is a most widely used community detection algorithm. Both baselines perform well when the input comes from the right model. Here we present the result when the model is mis-specified. See Figure 9. In Figure 9b and 9d, we plot the scatter plot between the true pairwise distances and estimated distances by our algorithm and a baseline. For the block model, note that the true distances can be only either 0 or 1 (after proper rescaling). We observe that our algorithm can automatically detect SBM and SWM; meanwhile, baselines have reduced performance on models for which they are not optimized.

G.2 Real data

This section evaluates our algorithm on a Twitter dataset related to the US presidential election in 2016. We evaluate the algorithm (1) against binary classification problems for a fair comparison against prior related algorithm (Section G.2.2), (2) against ground-truth of Senate and House members' political leanings (Section G.2.3), and (3) against state level political leanings (Section G.2.4).

Data collection. From October 1 to November 30, 2016, we used the Twitter streaming API to track tweets that contain the keywords "trump," ''clinton," "kaine," "pence," and "election2016" as text, hashtags (#trump) or mentions (trump). Keyword matching is case-insensitive, and #election2016 is Twitter's recommended hashtag for the US elections of 2016. We collected a total of 176 million tweets posted by 12 million distinct users.

We build a directed graph of the users so that node u connects to node v (the edge (u, v) exists) if and only if u retweets/replies to v. Our graph is unweighted because we observe insignificant performance differences between weighted and unweighted graphs. Then we choose 3000 nodes

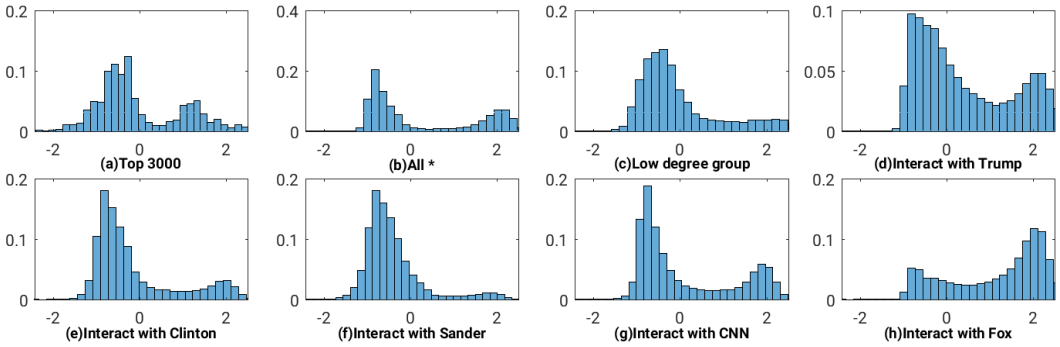


Figure 10: Latent distribution of different groups of users. Negative latent variables correspond to liberal users. Group a. Distribution of all influencers (top 3000). Group b. Distribution of users with 30 or more edges. Group c. Distribution of users with only one edge. Group d. Distribution of users who ever interacted with Trump. Group e. Distribution of users who ever interacted with Clinton. Group f. Distribution of users who ever interacted with Sanders. Group g. Distribution of users who ever interacted with CNN Politics. Group h. Distribution of users who ever interacted with Fox News.

with the largest in-degrees as our *influencers* and construct B. the mean out-degree of the followers is 4.6. The influencers also appear at the left-hand side of B because they can also follow other influencers.

Sparsity of the data. Our dataset is very sparse. For example, only 11 out of the 447 legislators with ground-truth scores are in the influencer set (a considerable portion of the influencers are often not highly visible in traditional media). The median degree (in + out) of these legislators is 19. It appears that very few legislators actively use Twitter to discuss election-related topics. The sparsity issue causes the performance degradation of many baselines, many of which have made dense graph assumptions.

Regularization and choice of θ. In our algorithm (BIPARTITE-EST(B) in Fig. 8), a regularization parameter θ needs to be decided. While our result suggests that any $\theta < 0.75$ works (Proposition C.1), the degrees in the real-world graph are more skewed than the graphs specified by our model so θ impacts the performance of our algorithm. θ is chosen by using a small portion of classification data as in-sample, *i.e.,* find θ so that the classification error is minimized for the in-sample data. The skewed degrees also require us to use the standard regularization techniques (introduced and studied by [21, 64]) to push the singular values to be better positioned at the cost of reducing the gaps between two consecutive singular values. *i.e.,* let D be a diagonal matrix such that $D_{i,i}$ is the row sum of A, and compute $A \leftarrow D^{-1/2}AD^{-1/2}$.

Estimation of followers. When influencers' latent variables x_i's are known/estimated, one may run a simple grid-search to find the maximum-likelihood estimator (MLE) for each follower, *i.e.,* examine $\{0, \epsilon, 2\epsilon, ..., 1\}$ and find the value that maximizes the likelihood: $\Pr[B_i \mid \widehat{y}_i] = \prod_{j \le n} \left(\frac{\kappa(\widehat{y}_i, x_j)}{n}\right)^{I(B_{i,j}=1)} \left(1 - \frac{\kappa(\widehat{y}_i, x_j)}{n}\right)^{I(B_{i,j}=0)}$. The exponent Δ in $\kappa(\cdot,\cdot)$ can also be estimated from the data.

The (approximate) MLE possesses the following property: 1. *Constant additive error:* for any constants δ and ϵ, there exists a constant C so that if a follower's expected degree is larger than C, then with probability $\ge 1-\delta$, the additive error is ϵ. 2. *Optimality:* standard lower bound arguments using statistical diference[2] gives us that with at least constant probability the estimation error is $\Omega(\epsilon)$ so the (approximate) MLE is asymptotically optimal.

Speeding up the estimation for the followers. The grid-search above finds approximate MLE in polynomial time but in practice they are too slow so we use a simpler heuristics to estimate x_i's: we take the mean of x_i's neighbors as the estimate of x_i. Our heuristics preserves the following property: if $x_i < x_j$, then with probability $1-\delta$, $\widehat{x}_i < \widehat{x}_j$ so long as the expected degrees of x_i and x_j are larger than a suitable constants.

[2]The statistical difference between $B_{i,:}$ and $B_{j,:}$ is a constant for any y_i and y_j so the probability of having an estimation error is also a constant.

As most of our experiments assess the quality of order statistics of the users, such heuristics has adequate performance.

Baseline algorithms. Our focus is graph-based algorithms. Algorithms that use the content of tweets to forecast users' political leanings are beyond the scope of this project.

Newman's spectral algorithm for modularity. A spectral-based algorithm that maximizes modularity [55] of the interaction graph. In this algorithm, PCA is applied to the properly normalized interaction graph. We use the first singular vector to decide the membership.

Correspondence analysis. Another spectral algorithm that was frequently used by political scientists, see *e.g.,* [18] and discussions therein. The algorithm first normalized the graph with its column and row sums. Then it uses the first left singular vector, which is the result of SVD, to estimate the influencers' latent positions. Standard Kernel PCA techniques are then used to "generalize" the model and predict the followers.

Label propagation: Label propagation (LP) algorithms are local algorithms so that each user updates his/her latent variable based on estimates of his/her neighbors' latent variables. We consider two versions of LP, namely majority [56] and random walks [54]. For the majority algorithm, we use output of modularity algorithm as the initial weights. For the random walks one, we set two presidential candidates to be 0 and 1, respectively.

Multidimensional scaling. We use a standard MDS algorithm [49] on the raw social network and low-dimension approximation of the social network.

G.2.1 Qualitative summaries

We first present several key qualitative findings, which serve as a sanity check to ensure that the outcomes of our model are consistent with common sense.

Distribution of the users. Figure 10 shows the latent variables for different groups of users. The output is *standardized* (so that the standard deviation is 1). 1% of the outliers are removed in the visualization. We consider the following groups: (a) influencers, (b) all users with at least 30 outgoing edges, (c) users with only one edge, (d - f) users that referred to Trump, Clinton, and Sanders, respectively, and (h & g) users that referred to CNN Politics and Fox News, respectively.

We observe a bimodal distribution in both groups and more left-leaning populations. Most of the latent estimates in group (c) are negative (liberal). This suggests that the first account a Twitter user refers to usually is a left-leaning media. Many left-leaning users refer to Trump (mainly to bash him), which is consistent with his media coverage. Users referring to Sanders skew to the left. CNN Politics attracts more left-followers while Fox News attracts more right-followers.

Distribution of the edges: We can also use heatmap to visualize the interactions between users. Figure 4 compares the inferred kernel against SWM and SBM. Specifically, Figure 4b represents a small-world interaction and Figure 4c represents the stochastic block model.

Figure 4a is a visualization of our model. We construct the image as follows. We first sort the influencers according to their latent scores and partition them into 30 groups of equal size. Each group corresponds to one row in the image (*e.g.,* first row is the leftmost group). For each group, we compute the "average" histogram of users that refers to a member in the group. Here, we use 20-bins. Finally, we color code the histogram, *e.g.,* a white pixel corresponds to a large bin.

Thus, Figure 4a approximates the inferred kernel function for the influencers. We observe that the diagonals are brighter (resembling small world) but at the same time there is some "blurred" block structure. This observation confirms that the real dataset "sits" between the small-world and block models, and highlights the need to design a unified algorithm that disentangles these two models.

Distribution of the presidential candidates. As mentioned, we perform a sanity check on our estimates of all candidates' latent scores. These (unnormalized) scores are: Sanders (-0.272) $<$ Clinton (-0.014) $<$ Kasich (0.01) $<$ Cruz (0.013) $<$ Trump(0.037).

Table 1: Performance of the algorithms on classifying user's political leaning. Column 2: all 752 labeled users, Column 3: 270 influencers in the top 3000 list, Column 4: 571 users with 30 or more edges (exclude those influencers), Column 5: the rest users. MDS 5 = MDS with 5 eigenvalues; MDS = MDS with all eigenvalues.

Algo.	All (752)	Top 3000 (270)	30+ (571)	Others (123)
Ours	**89.9%**	89.6%	**91.9%**	**87.8%**
Modularity	87.7%	89.6%	90.9%	82.1%
Correspond Analysis	57.3%	45.2%	58.1%	57.7%
Majority	88.4%	**90.7%**	90.7%	79.7%
Random Walk	65.2%	57.4%	64.6%	65.9%
MDS 5	64.8%	59.6%	65.7%	62.6%
MDS	57.3 %	66.3%	57.1%	55.3%

G.2.2 Classification result

We sample a subset of users and label them as conservative or liberal according to four categories. Category 1. *Top 2000*, *i.e.,* one of the 2000 users with the largest in-degree. Category 2. Top 2000 to 3000. *i.e.,* one of the 1000 users with an in-degree rank between 2000 and 3000. Category 3: 30+. *i.e.,* users with at least 30 out-going edges. Category 4. Everyone, *i.e.,* all users.

Labeling. For each selected user, we ask two human judges, who are either the authors of this paper or workers from Amazon Mechanical Turk, to label the user as likely to vote Clinton or Trump in the final election. If it is not clear, *e.g.,* the user criticizes both candidates, the judges can label the user "unclear." The information we provide to the judge includes user's self-reported name, screen name, Twitter-verified account status (usually indicates a celebrity), self-description, URL, and a random sample of at most 20 tweets. When the labeling is complete, we discard all users except for those unanimously labeled as likely to vote for one of the candidates, for a total of 45% of the original data.

We need to decide a threshold to turn the output of our algorithms (and some of the baseline algorithms) into binary forecasts. We use a subset of classification tasks as in-sample data. We note that the threshold is robust for both in-sample and out-of-sample data (the performance difference is inconsequential).

Results. See Table 1. The classes are balanced. We report classification accuracy on (1) all 752 labeled users, (2) 270 influencers in the top 3000 list, (3) 571 large-degree users with 30 or more edges (but not influencers), and (4) the rest users.

Our algorithm has the best overall performance and is the best or near-best for each subgroup. Only the performance of the modularity algorithm, which is optimized for classification applications is close to ours. But as we shall see in Section G.2.3 and G.2.4, the algorithm performs poorly for tasks that requires understanding users' latent structure in finer granularity. For the predictions of average users (the last column), the accuracies of most baselines are below 80%, which is consistent with prior experiments [65].

G.2.3 Correlation with ground-truth

We compare the latent estimates of politicians (members of the 114th Congress) and the ground-truth. The ground-truth of these politicians is estimated by various third parties using data sources such as voting record and co-sponsorship [57].

Standard error Beyond correlation, we also need to estimate the statistical significance of our estimates. We use bootstrapping to compute the standard error of our estimator, and then use the standard error to estimate the p-value of our estimator. Specifically, our goal is to understand the explanatory power of our latent estimates $\widehat{x}_l(B)$ (how we write it to highlight the statistics we compute depends on the bipartite graph B) to response y_i representing the ground-truth of the politicians. Thus, we set up a linear regression: $y \sim \beta_1 \widehat{x} + \beta_0$ In the bootstrapping procedure, we repeat the following process for k times: Sample 80% of the edges from B and compute the latent estimates as well as β_1 (by running an OLS linear regression). We mark the estimate at the i-th repetition $\beta_{1,i}$. The standard error of β_1 is the empirical standard deviation of $\beta_{1,i}$. The t-statistics can also be estimated as $(\sqrt{k\widehat{\beta_1}})/s.e(\widehat{\beta_1})$, which can be used to construct p-value. Here we set k to be 50. We also do not standardize the estimates for all the estimation algorithms. The decision of whether to

Table 2: Explanatory power of the estimates of the latent variables against ground-truth of politicians' ideology scores. S.E. stands for standard error

Algo.	ρ	Slope of β	S.E.	p-value
Ours	**0.53**	9.54	0.28	< 0.001
Modularity	0.16	1.14	0.02	< 0.001
Correspond Analysis	0.20	0.11	7e-4	< 0.001
Majority	0.13	0.09	0.02	< 0.001
Random Walk	0.01	1.92	0.65	< 0.001
MDS 5	0.05	30.91	120.9	0.09
MDS	0.31	101.3	14.88	< 0.001

standardize latent estimates is inconsequential as the ratio between the slope and standard deviation is more important.

Table 2 shows the result. Except for MDS 5, all the models' forecast power is statistically significant. Our algorithms are again the best here, and are significantly better than the rest algorithms.

G.2.4 By State analyses

Next, we aggregate users' latent variables by inferring their locations (see below) and grouping users by state. We sort the states by the mean of the latent scores of users in that state. If we assume that voters in the same state come from the same underlying distribution and different states have different means, then the empirical mean provides a min-variance unbiased estimate of the mean of the distribution of a state.

Around 10% of the users' locations can be found and no particular state is over- or under-represented. See also Figure 11. We construct the graph using two datasets: data prior to November 8, 2016, and all data until November 30. Since there is little difference, we decide to use the full dataset, *i.e.,* all experiments use the same data.

Samples from the population. We observe that the correlation between the number of users in a state and the population in the state is 0.968, which is very high. Figure 11 shows the corresponding scatter plot. This serves as a sanity check of our location extraction algorithm. We also observe that while Twitter users are biased samples of voters, no particular state is over- or under- represented.

Result Our goal is not to predict the election outcome since we know that Twitter users are biased samples of the voter population. On the other hand, we observe that the order statistics of the states' latent variables have the strongest or near-strongest explanatory power against the metrics below. See Figure 3.

Binary outcome of 2016 US presidential election. Similar to Section G.2.2, we find an optimal threshold that turns the ranking into a binary forecast and maximize the forecasting accuracy for both our algorithm and other baselines.

Order statistics vs. winning margins. We can also turn the election results into scalars. For each state, we compute the ratio between the number of votes for Trump and the number of votes for Clinton and order the states according to the ratio. A state with a small ratio corresponds to a left-leaning state. Our goal is to understand the correlation between the order from our estimation and the order by the margin of winning in the election.

Long-term ideology. We also compare our order statistics against the estimated liberal ideology in [66], which is based on the survey and sociodemographic data. This dataset can be considered as a long-term ideology score while the election result is a short-term one.

Table 3 shows the result. The following states are mis-classified in our algorithm: NV, NH, NJ, UT, and WI. Except for NJ, all other states are swing states. NJ has a Republican governor (Christie) who ran in the 2016 presidential campaign before dropping out. Except for the Kendall correlation for the election data, our algorithm continues to have the best performance. The winner (label propagation/majority) here also suffers from poor performance on other metrics.

Table 3: Quality of the order statistics by states. Miss = misclassified state; E = correlation with election winning margin; L = represents correlation with long-term ideology. K = Kendall's τ; S = Spearman's ρ. For example, KE refers to Kendall's correlation with election winning margin.

Algo.	Miss	KE	SE	KL	SL
Ours	**5**	0.613	**0.816**	**0.636**	**0.800**
Modularity	8	0.462	0.649	0.468	0.641
Correspond Analysis	21	0.410	0.567	0.374	0.547
Majority	6	**0.619**	0.808	0.605	0.772
Random Walk	14	0.242	0.340	0.331	0.447
MDS 5	19	0.087	0.110	0.112	0.150
MDS	21	0.470	0.632	0.387	0.550

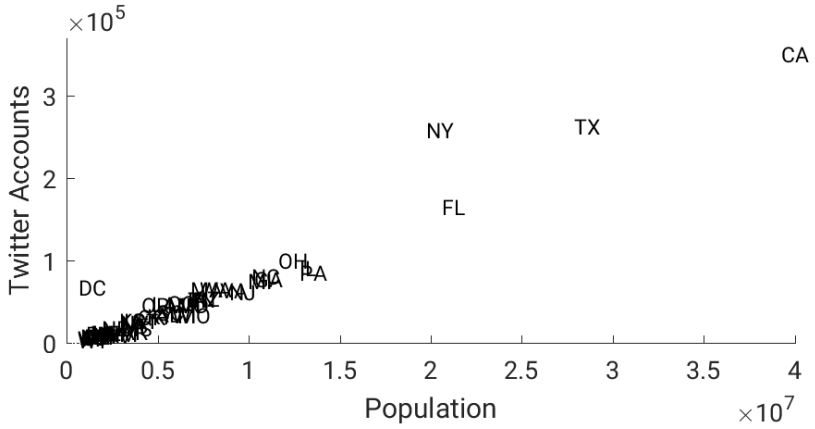


Figure 11: Scatter plot between state population and sampled accounts per state. No particular state is over- or under-represented

H Summary of notations

- $A \in R^{n \times n}$: the adjacent matrix of an undirected graph. In the simplified graph model, A is the input graph. In the bipartite graph model, A is the regularized matrix over $B^{\mathsf{T}}B$.
- $B \in R^{m \times n}$: the bipartite graph matrix, *i.e.,* $B_{i,j} = 1$ if and only if follower i is connected to influencer j.
- $\mathbf{Ball}(z, r) = \{z' : \|z' - z\| \leq r\}$
- $C(n)$: the normalization constant in the undirected graph model, *i.e.,* $\Pr[\{x_i, x_j\} \in E] = \kappa(x_i, x_j)/C(n)$.
- $\mathcal{C}$: a curve in $R^{N_{\mathcal{H}}}$, defined as $\mathcal{C} = \{\Phi(x)\}_{x \in [0,1]}$
- $\mathcal{D}$: the distribution in which x_i and y_I come from.
- D: the distance estimate, *e.g.,* $D_{i,j}$ is the estimate of distance between x_i and x_j in our algorithm.
- d: the number of eigenvalues to keep; in the section for isomap technique, it sometimes is used to refer to the length of the shortest path.
- $\mathcal{H}$: the reproducing kernel Hilbert space of κ.
- $K \in R^{n \times n}$: the kernel/Gram matrix associated with κ, *i.e.,* $K_{i,j} = \kappa(x_i, x_j)$.
- $\mathcal{K}$: an integral operator defined as $\mathcal{K}f(x) = \int \kappa(x, x')f(x')dF(x')$.
- $F(n)$: the cdf of $\mathcal{D}$.
- $\mathcal{M}$: an integral operator defined as $\mathcal{M}f(x) = \int \mu(x, y)f(y)dF(y)$.
- $M \in R^{n \times n}$: the kernel/Gram matrix associated with μ, *i.e.,* $M_{i,j} = \mu(x_i, x_j)$.
- m: the number of followers in the bipartite graph model.
- $N_{\mathcal{H}}$: the number of eigenvalues in $\mathcal{K}$, which could be countably infinite.

- n: the number of nodes in the simplified model and the number of influencers in the bipartite grpah model.
- $\mathcal{P}$: projection operators.
- $[\tilde{U}_X, \tilde{S}_X, \tilde{V}_S]$ where $X \in \{A, K, M\}$: the SVD of X.
- $[U_X, S_X, V_S]$, where $X \in \{A, K, M\}$: the first d singular vectors/values of X. Note here they implicitly depend on d.
- $\mathbf{X} = \{x_1, ..., x_n\}$: the set of nodes in the simplified model and the set of influencers in the bipartite graph model.
- $\widehat{x}_i$: our algorithm's estimate of x_i.
- $\mathbf{Y} = \{y_1, ..., y_n\}$: the set of followers in the bipartite graph model.
- $\widehat{y}_i$: our algorithm's estimate of y_i.
- z_i: the feature of x_i, *i.e.,* $\Phi(z_i)$.
- $\widehat{z}_i$: our algorithm's estimate of z_i.
- δ_i: the eigengap, defined as $\delta_i = \lambda_i - \lambda_{i+1}$.
- λ_i: the eigenvalues of $\mathcal{K}$ unless otherwise specified.
- $\rho(n)$: we also re-parametrize $C(n) = n/\rho(n)$, *i.e.,* $\rho(n) = n/C(n)$.
- $\kappa : [0, 1] \times [0, 1] \to (0, 1]$: the kernel function.
- Δ: the exponent in the small-world kernel, *i.e.,* $\kappa(x_i, x_j) = c_0/(|x_i - x_j|^{\Delta} + c_0$.
- $\Phi : [0, 1] \to R^{N_{\mathcal{H}}}$ the feature map associated with $\mathcal{K}$.
- $\widehat{\Phi}$: our algorithm's estimate of Φ.
- $\Phi^{\mathcal{M}}$: the feature map associated with $\mathcal{M}$.
- Φ_d: the first d coordinates of the feature map. It is also overloaded to be in $R^{N_{\mathcal{H}}}$ by padding 0's after the d-th coordinate.
- $\widehat{\Phi}_d$: our algorithm's estimate of Φ_d.
- $\Phi_d^{\mathcal{M}}$: the first d coordinates of $\Phi^{\mathcal{M}}$.
- ψ_i: the i-th eigenfunction of $\mathcal{K}$.
- μ: a kernel used for the analysis of the bipartite graph model, *i.e.,* $\mu(x, x') = \int \kappa(x, z)\kappa(z, x')dF(z)$ (see Appendix C).
- Norms: $\|A\|$ is the opereator/spectral norm of A. $\|A\|_F$ is the Frobenius norm. $\|A\|_{\mathrm{HS}}$ is the Hilbert-Schmidt norm (see Definition B.4).