Semiparametric Nonlinear Bipartite Graph Representation Learning with Provable Guarantees

Sen Na [1] Yuwei Luo [1] Zhuoran Yang [2] Zhaoran Wang [3] Mladen Kolar [4]

Abstract

Graph representation learning is a ubiquitous task in machine learning where the goal is to embed each vertex into a low-dimensional vector space. We consider the bipartite graph and formalize its representation learning problem as a statistical estimation problem of parameters in a semiparametric exponential family distribution: the bipartite graph is assumed to be generated by a semiparametric exponential family distribution, whose parametric component is given by the proximity of outputs of two one-layer neural networks that take high-dimensional features as inputs, while nonparametric (nuisance) component is the base measure. In this setting, the representation learning problem is equivalent to recovering the weight matrices, and the main challenges of estimation arise from the nonlinearity of activation functions and the nonparametric nuisance component of the distribution. To overcome these challenges, we propose a pseudo-likelihood objective based on the rank-order decomposition technique and show that the proposed objective is strongly convex in a neighborhood around the ground truth, so that a gradient descent-based method achieves linear convergence rate. Moreover, we prove that the sample complexity of the problem is linear in dimensions (up to logarithmic factors), which is consistent with parametric Gaussian models. However, our estimator is robust to any model misspecification within the exponential family, which is validated in extensive experiments.

1. Introduction

Graphs naturally arise as models in a variety of applications, ranging from social networks (Scott, 1988) and molecular biology (Higham et al., 2008) to recommendation systems (Ma et al., 2018) and transportation (Bell & Iida, 1997). In a variety of problems, graphs tend to be high-dimensional and highly entangled, and hence difficult to directly learn from. As a prominent remedy, graph representation learning aims to learn a mapping that represents each vertex as low-dimensional vector such that structural properties of the original graph are preserved. Those learned low-dimensional representations, also called embeddings, are further used as the input features in downstream machine learning tasks, such as link prediction (Taskar et al., 2004; Al Hasan & Zaki, 2011), node classification (Bhagat et al., 2011), and community detection (Fortunato, 2010).

There are three major approaches to graph embedding: matrix factorization-based algorithms (Belkin & Niyogi, 2002; Ahmed et al., 2013), random walk algorithms (Perozzi et al., 2014; Grover & Leskovec, 2016), and graph neural networks (Scarselli et al., 2008; Zhou et al., 2018; Wu et al., 2019). These approaches can be unified via the encoder-decoder framework proposed in Hamilton et al. (2017b). In this framework, the encoder is a mapping that projects each vertex or a subgraph to a low-dimensional vector, whereas the decoder is a probability model that infers the *structural information* of the graph from the embeddings generated by the encoder. The structural information here depends on the specific downstream tasks of interest, which also determines the loss function of the decoder. The desired graph representations are hence obtained by minimizing the loss function as a function of embedding vectors. For example, in the link prediction task, the decoder predicts whether an edge between two vertices exists or not using a Bernoulli model and logistic loss function, and the model parameter is a function of embeddings (Baldin & Berthet, 2018).

Such an encoder-decoder architecture motivates the study of graph representation learning through the lens of statistical estimation for generative models. In particular, suppose the observed graph is generated by a statistical model specified by the decoder with *true* graph representations as its inputs. We can then assess the performance of a graph embedding algorithm by examining the difference between the learned representation and the ground truth. Baldin & Berthet (2018)

[1] Department of Statistics, University of Chicago, Chicago IL, USA [2] Department of Operations Research and Financial Engineering, Princeton University, Princeton NJ, USA [3] Department of Industrial Engineering and Management Sciences, Northwestern University, Chicago IL, USA [4] Booth School of Business, University of Chicago, Chicago IL, USA. Correspondence to: Sen Na <senna@uchicago.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

adopted this perspective to study the performance of a linear embedding method for the link prediction problem. The validity of their results hinges on the condition that both the linear model of the encoder and the Bernoulli model of the decoder are correctly specified. When either of these assumptions are violated, they would incur large estimation error. Recent advances in graph representation learning are attributed to more flexible decoders (Cho et al., 2014; Goodfellow et al., 2016; Badrinarayanan et al., 2017), which are based on deep neural networks and can handle graphs with edge attributes that can be categorical. These approaches are poorly understood from a theoretical point of view.

In the present paper, we focus on *bipartite graphs*, where there are two distinct sets of vertices, U and V, and only edges between two vertices in different sets are allowed. We study the semiparametric nonlinear bipartite graph representation learning problem under the encoder-decoder framework. We assume that each vertex $u \in U$ is associated with a high-dimensional Gaussian vector $\mathbf{x}_u \in \mathbb{R}^{d_1}$. Similarly, each vertex $v \in V$ is associated with a high-dimensional Gaussian vector $\mathbf{z}_v \in \mathbb{R}^{d_2}$. The encoder maps them via one-layer neural networks to low-dimensional vectors $\phi_1(\mathbf{U}^{\star T}\mathbf{x}_u), \phi_2(\mathbf{V}^{\star T}\mathbf{z}_v) \in \mathbb{R}^r$, where $\mathbf{U}^\star \in \mathbb{R}^{d_1 \times r}$, $\mathbf{V}^\star \in \mathbb{R}^{d_2 \times r}$ are weight matrices, $\{\phi_i\}_{i=1,2}$ are activation functions evaluated entrywise, and $r \ll (d_1 \wedge d_2)$. Furthermore, in the decoder, we consider the link prediction task under a semiparametric model. In particular, we assume that the attribute of an edge follows a natural exponential family distribution parameterized by the proximity between two vertices, which is defined as the inner product $\langle \phi_1(\mathbf{U}^{\star T}\mathbf{x}_u), \phi_2(\mathbf{V}^{\star T}\mathbf{z}_v)\rangle$ between the embedding vectors. Here, $\phi_1(\mathbf{U}^{\star T}\mathbf{x}_u)$ is the embedding vector of u, while $\phi_2(\mathbf{V}^{\star T}\mathbf{z}_v)$ is the embedding vector of v. We do not specify the base measure of the exponential family distribution, but, instead, treat it as a nuisance parameter. This gives us a semiparametric model for the decoder and robustness to model misspecification within the exponential family.

In the above described semiparametric nonlinear model, our goal is to recover weight matrices $\mathbf{U}^\star$ and $\mathbf{V}^\star$. Based on these weight matrices, we can then compute embeddings for all vertices. There are two main obstacles that make the estimation problem challenging. First, while the activation functions $\{\phi_i\}_{i=1,2}$ make the encoder model more flexible, their nonlinearity leads to a loss function that is nonconvex and nonsmooth. Second, while the unknown nonparametric nuisance component of the decoder model makes the graph representation learning robust to the model misspecification, it also makes the likelihood function not available. To overcome these obstacles, we propose a pseudo-likelihood objective, which is minimized at $(\mathbf{U}^\star, \mathbf{V}^\star)$ locally. We analyze the landscape of the empirical objective and show that, in a neighborhood around the ground truth, the objective is strongly convex. Therefore, the vanilla gradient descent (GD) achieves linear convergence rate. Moreover, we prove that the sample complexity is linear in dimensions $d_1 \vee d_2$, up to logarithmic factors, which matches the best known result under the parametric model (Zhong et al., 2018). Experiments on synthetic and real data corroborate our theoretical results and illustrate flexibility of the proposed representation learning model.

Notations. For any positive integer n, $[n] = \{1, 2, \ldots, n\}$ denotes the index set, and Unif($[n]$) is a uniform sampling over the indices. We write $a \lesssim b$ if $a \leq c \cdot b$ for some constant c, and $a \asymp b$ if $a \lesssim b$ and $b \lesssim a$. We define $\delta_{ij} = \mathbf{1}_{i=j}$, which equals to 1 if $i = j$ and 0 otherwise. For any matrix $\mathbf{U}$, vec($\mathbf{U}$) denotes the column vector obtained by vectorizing $\mathbf{U}$ and $\|\mathbf{U}\|_{p,q} = (\sum_j(\sum_i |\mathbf{U}_{ij}|^p)^{q/p})^{1/q}$. As usual, $\|\mathbf{U}\|_F$, $\|\mathbf{U}\|_2$ refer to the Frobenius and operator norm, respectively, and $\sigma_p(\mathbf{U})$ denotes the p-th singular value of $\mathbf{U}$. For a square matrix $\mathbf{U}$, $\operatorname{diag}(\mathbf{U}) = (\mathbf{U}_{11}; \mathbf{U}_{22}; \ldots)$ is a vector including all diagonal entries of $\mathbf{U}$; when $\mathbf{U}$ is symmetric, $\lambda_{\max}(\mathbf{U})$ ($\lambda_{\min}(\mathbf{U})$) denotes its maximum (minimum) eigenvalue. We write $\mathbf{A} \succeq \mathbf{B}$ if $\mathbf{A} - \mathbf{B}$ is positive semidefinite and $\mathbf{A} \succ \mathbf{B}$ if it is positive definite. For any vector $\boldsymbol{a}$, $\|\boldsymbol{a}\|_{\min} = \min_i |\boldsymbol{a}_i|$ is the minimal absolute value of its entries.

Structure of the paper. In Section 2, we formalize the semiparametric graph representation learning problem and introduce related work. In Section 3, we present our estimation method by proposing a pseudo-likelihood objective, and the theoretical analysis of such objective is provided in Section 4. In Section 5 we show experimental results and conclusions are summarized in Section 6. Proofs and auxiliary experiments are referred to the supplement.

2. Preliminaries and related work

We describe the setup of our problem and introduce the applications and related work. We particularly focus on the statistical literature on theory of semiparametric estimation and matrix completion, although bipartite graph representation learning has been routinely applied to varied deep neural networks (Nassar, 2018; Wu et al., 2018). We point reader to Zha et al. (2001) for a survey on bipartite graph.

2.1. Problem formulation

Let $G = (U, V, E)$ be a bipartite graph where U and V are two sets of vertices and E denotes the set of edges between two vertex sets. For each vertex $u \in U$, we assume it is associated with a Gaussian vector $\mathbf{x}_u \in \mathbb{R}^{d_1}$, while for each $v \in V$ we have a Gaussian vector $\mathbf{z}_v \in \mathbb{R}^{d_2}$. An edge between u and v has an attribute $y_{(u,v)}$ that follows the following semiparametric exponential family model

$$\begin{aligned} &P(y_{(u,v)} \mid \mathbf{\Theta}^\star_{(u,v)}, f) \\ &\quad = \exp(y_{(u,v)} \cdot \mathbf{\Theta}^\star_{(u,v)} - b(\mathbf{\Theta}^\star_{(u,v)}, f) + \log f(y_{(u,v)})), \quad (1) \end{aligned}$$

which is parameterized by the base measure function $f : \mathbb{R} \to \mathbb{R}$ and a scalar

$$\mathbf{\Theta}^{\star}_{(u,v)} = \langle \phi_1(\mathbf{U}^{\star T}\mathbf{x}_u), \phi_2(\mathbf{V}^{\star T}\mathbf{z}_v)\rangle.$$

In model (1), $b(\cdot,\cdot)$ is the log-partition function (or normalizing function) that makes the density have unit integral. The parametric component of the exponential family, $\mathbf{\Theta}^{\star}_{(u,v)} = \mathbf{\Theta}^{\star}(\mathbf{x}_u, \mathbf{z}_v)$, depends on the covariate $\mathbf{x}_u$ coming from the set U and the covariate $\mathbf{z}_v$ coming from the set V. The nonparametric component f is treated as a nuisance parameter, which gives us flexibility in modeling the edge attributes. To make notation concise, we will drop the subscript of $\mathbf{x}_u$ and $\mathbf{z}_v$ hereinafter, and use $\mathbf{x}$ and $\mathbf{z}$ to denote covariates from set U and V, respectively. In our analysis, the activation functions $\{\phi_i\}_{i=1,2}$ have one of the following three forms: Sigmoid: $\phi(x) = \exp(x)/(1+\exp(x))$; Tanh: $\phi(x) = (\exp(x)-\exp(-x))/(\exp(x)+\exp(-x))$; ReLU: $\phi(x) = \max(0,x)$.

We formalize the bipartite graph representation learning as a statistical parameter estimation problem of a generative model. In particular, suppose the graph is generated by the exponential family model (1) with some unknown base measure f, and we observe part of edge attributes, y, and associated covariates on two ends, $\mathbf{x}$ and $\mathbf{z}$. Thus, we obtain data set $\{(y_{ij}, \mathbf{x}_i, \mathbf{z}_j)\}_{i,j}$ where i, j index the vertices of two sets. The graph representation learning in our setup is then equivalent to recovering $\mathbf{U}^{\star} \in \mathbb{R}^{d_1\times r}$ and $\mathbf{V}^{\star} \in \mathbb{R}^{d_2\times r}$, which can be used to compute parametric component of the decoder model and estimate embedding vectors, $\phi_1(\hat{\mathbf{U}}^T\mathbf{x})$ and $\phi_2(\hat{\mathbf{V}}^T\mathbf{z})$, for all vertices in two sets, since activation functions are user-chosen and known.

2.2. Applications and related works

Graph representation learning underlies a number of real world problems, including object recognition in image analysis (Bunke & Messmer, 1995; Fiorio, 1996), community detection in social science (Perozzi et al., 2014; Cavallari et al., 2017), and recommendation systems in machine learning (Kang et al., 2016; Jannach et al., 2016). See Bengio et al. (2013), Hamilton et al. (2017a), Hamilton et al. (2017b) for recent surveys and other applications. The bipartite graph is of particular interest since it classifies vertices into two types, which extensively appears in modern applications.

For concreteness, in user-item recommendation systems, the attribute of an edge between a user node and an item node represents the rating, which is modeled by the proximity of projected features onto the latent space. Specifically, each user is represented by a high-dimensional feature vector $\mathbf{x}$ and each item is represented by a high-dimensional feature vector $\mathbf{z}$. A simple generative model for the rating y that a user gives to an item is $y = \langle \mathbf{U}^{\star T}\mathbf{x}, \mathbf{V}^{\star T}\mathbf{z}\rangle + \epsilon$ with $\epsilon \sim \mathcal{N}(0,1)$ independent from $\mathbf{x}, \mathbf{z}$. Such a model is studied in the inductive matrix completion (IMC) literature (Abernethy et al., 2006; Jain & Dhillon, 2013; Si et al., 2016; Berg et al., 2017). Zhong et al. (2018) studied nonlinear IMC problem, where a generalized model for the rating is $y = \langle \phi(\mathbf{U}^{\star T}\mathbf{x}), \phi(\mathbf{V}^{\star T}\mathbf{z})\rangle + \epsilon$, with $\phi(\cdot)$ being a common activation function. In this generalized nonlinear model, one-layer neural network compresses the high-dimensional features into low-dimensional embeddings. Zhong et al. (2018) proposed to minimize the squared loss to recover weight matrices $\mathbf{U}^{\star}$ and $\mathbf{V}^{\star}$, and established consistency for their minimizer, with linear sample complexity in dimension $d_1 \vee d_2$, up to logarithmic factors.

Our work contributes to this line of research by enhancing the IMC model from two aspects. First, we allow for two separate neural networks to embed user and item covariates. Although this modification may seem minor, it makes theoretical analysis more challenging when two networks mismatch: one network has a smooth activation function while the other does not. Second, we consider an exponential family model with unknown base measure, which extends the applicability of the model and allows for model misspecification within the exponential family. In particular, the semiparametric setup makes our estimator independent of the specific form of f. For example, the model in Zhong et al. (2018) is a special case of (1) with $f(y) = \exp(-y^2/2)$, while the link prediction problem in Liben-Nowell & Kleinberg (2007) and Menon & Elkan (2011) is a special case with $f(y) = 1$.

Furthermore, our work contributes to the literature on graph embedding (Qiu et al., 2018; Goyal & Ferrara, 2018). Our paper studies the bipartite graph and casts the graph representation learning as the problem of parameter estimation in a generative model. This setup allows us to analyze statistical properties, such as consistency and convergence rate, of the learned embedding features. To the best of our knowledge, statistical view of representation learning is missing although it was successfully used in real experiments (see, e.g., Graepel et al., 2001; Yang et al., 2015). In addition, our work also contributes to a growing literature on semiparametric modeling (Fengler, 2005; Li & Liang, 2008; Fan et al., 2017), where the parametric component in (1) is given by $\mathbf{\Theta}^{\star} = \boldsymbol{\beta}^{\star T}\mathbf{x}$ and the goal is to estimate $\boldsymbol{\beta}^{\star}$ by regressing y on $\mathbf{x}$, without knowing f. Fosdick & Hoff (2015) formalized the representation learning as a latent space network model, where the parameter $\mathbf{\Theta}^{\star}$ is given by the inner product of two latent vectors and $f(y) = \exp(-y^2/2)$, that is under a Gaussian noise setup, and proposed methodology for testing the dependence between nodal attributes and latent factors. Ma et al. (2019) studied a similar model with $f(y) = 1$ and proposed both convex and nonconvex approaches to recover latent factors. However, our work is more challenging due to the nonlinearity of activation functions and the missing knowledge of f.

Lastly, several estimation methods for pairwise measurements have been studied in related, but simpler, models (Chen & Goldsmith, 2014; Chen & Suh, 2015; Chen et al., 2016; Pananjady et al., 2017; Negahban et al., 2018; Chen & Candès, 2018; Chen et al., 2019). Chen et al. (2018) studied model (1) by assuming the parameter matrix of the graph to be low-rank, and estimated $\mathbf{\Theta}^\star$ as a whole. As a comparison, our model is more complicated since each entry of $\mathbf{\Theta}^\star$ in our setup is given by the inner product of two embedding vectors, which measures the proximity of two vertices. Our task is to recover two underlying weight matrices $\mathbf{U}^\star$, $\mathbf{V}^\star$ that are convolved by activation functions to formalize $\mathbf{\Theta}^\star$.

3. Methodology

We propose a pseudo-likelihood objective function to estimate the unknown weight matrices and discuss identifiability of the parameters. The objective function is minimized by the gradient descent with a constant step size. Theoretical analysis of the iterates is provided in Section 4.

The likelihood for the model in (1) is not available due to the presence of the infinite-dimensional nuisance parameter f. Using the rank-order decomposition technique (Ning et al., 2017), we focus on the pairwise differences and develop a pseudo-likelihood objective. Importantly, the differential pseudo-likelihood does not depend on f and, as a result, our estimator is valid for a wide range of distributions, without having to explicitly specify them in advance.

We follow the setup described in Section 2.1. To simplify the presentation, suppose we have $2n_1$ vertices in U and $2n_2$ vertices in V, denoted by $U = \{u_1, \ldots, u_{n_1}, u_1', \ldots, u_{n_1}'\}$ and $V = \{v_1, \ldots, v_{n_2}, v_1', \ldots, v_{n_2}'\}$, respectively. For $i \in [n_1]$ and $j \in [n_2]$, we let $\mathbf{x}_i = \mathbf{x}_{u_i}, \mathbf{x}_i' = \mathbf{x}_{u_i'}, \mathbf{z}_j = \mathbf{z}_{v_j}$, $\mathbf{z}_j' = \mathbf{z}_{v_j'}$, and suppose that $\mathbf{x}_i, \mathbf{x}_i' \overset{i.i.d}{\sim} \mathcal{N}_{d_1}(0, I)$ and $\mathbf{z}_j, \mathbf{z}_j' \overset{i.i.d}{\sim} \mathcal{N}_{d_2}(0, I)$, independent of each other. Further, we assume to observe m edge attributes, y, between vertices $\{u_1, \ldots, u_{n_1}\}$ and $\{v_1, \ldots, v_{n_2}\}$, and another m edge attributes, y', between $\{u_1', \ldots, u_{n_1}'\}$ and $\{v_1', \ldots, v_{n_2}'\}$, both of which follow the distribution in (1) and are sampled with replacement from the set of all possible $n_1 n_2$ edges. We note that the sampling setup is commonly adopted in the literature on partially observed graphs and matrix completion problems (Zhong et al., 2018; Chen et al., 2018), which is equivalent to assuming edges are missing at random.

Denote sample sets $\Omega = \{(y_{u(k),v(k)}, \mathbf{x}_{u(k)}, \mathbf{z}_{v(k)})\}_{k=1}^m$ and $\Omega' = \{(y'_{u'(l),v'(l)}, \mathbf{x}'_{u'(l)}, \mathbf{z}'_{v'(l)}\}_{l=1}^m$, where $u(k), u'(l)$ $=$ Unif$([n_1])$ and $v(k), v'(l) =$ Unif$([n_2])$. While the observations within Ω or Ω' are not independent, as they may have common features $\mathbf{x}$ or $\mathbf{z}$, the observations between Ω and Ω' are independent. Such two independent sets of samples are obtained by sample splitting in practice. We stress that the splitting is used only to make the analysis concise without enhancing the order of sample complexity. In particular, it does not help us avoid the main difficulties of the problem.

Based on samples Ω and Ω', we consider m^2 pairwise differences and construct an empirical loss function. For $k \in [m]$, let $k_1 = u(k)$, $k_2 = v(k)$, and $\mathbf{\Theta}^\star_{k_1 k_2} = \langle \phi_1(\mathbf{U}^{\star T} \mathbf{x}_{k_1}), \phi_2(\mathbf{V}^{\star T} \mathbf{z}_{k_2}) \rangle$ denote the true parameter associated with the k-th sample (similarly for $\mathbf{\Theta}^{\star\prime}_{l_1 l_2}$). Note that $\mathbf{\Theta}^\star_{k_1 k_2}$ is the underlying parametric component of the model that generates $y_k = y_{k_1, k_2}$. The key idea in constructing the pseudo-likelihood objective is to use rank-order decomposition to extract a factor, that is independently from the base measure. Given a pair of independent samples, y_k and y_l', we denote their order statistics as $y_{(\cdot)}$ and rank statistics as R. Then we know $y_{(\cdot)} = (y_k, y_l')$ or $y_{(\cdot)} = (y_l', y_k)$, and $R = (1, 2)$ or $R = (2, 1)$. Thus, $(y_{(\cdot)}, R)$ fully characterizes the pair (y_k, y_l'), and is hence a sufficient statistics. Note that

$$
\begin{aligned}
&P(y_k, y_l' \mid \mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) = P(y_{(\cdot)}, R \mid \mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) \\
&= P(R \mid y_{(\cdot)}, \mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) \cdot P(y_{(\cdot)} \mid \mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) \\
&= \frac{P(y_k|\mathbf{\Theta}^\star_{k_1 k_2}, f) P(y_l'|\mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) \cdot P(y_{(\cdot)}|\mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f)}{P(y_k|\mathbf{\Theta}^\star_{k_1 k_2}, f) P(y_l'|\mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) + P(y_k|\mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f) P(y_l'|\mathbf{\Theta}^\star_{k_1 k_2}, f)} \\
&\overset{(1)}{=} \frac{\exp(y_k \mathbf{\Theta}^\star_{k_1 k_2} + y_l' \mathbf{\Theta}^{\star\prime}_{l_1 l_2}) \cdot P(y_{(\cdot)} \mid \mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f)}{\exp(y_k \mathbf{\Theta}^\star_{k_1 k_2} + y_l' \mathbf{\Theta}^{\star\prime}_{l_1 l_2}) + \exp(y_l' \mathbf{\Theta}^\star_{k_1 k_2} + y_k \mathbf{\Theta}^{\star\prime}_{l_1 l_2})} \\
&= \underbrace{\frac{1}{1 + \exp\left(-(y_l - y_l')(\mathbf{\Theta}^\star_{k_1 k_2} - \mathbf{\Theta}^{\star\prime}_{l_1 l_2})\right)}}_{\text{local differential pseudo-likelihood}} \cdot P(y_{(\cdot)} \mid \mathbf{\Theta}^\star_{k_1 k_2}, \mathbf{\Theta}^{\star\prime}_{l_1 l_2}, f).
\end{aligned}
\tag{2}
$$

The first term is the density of the rank statistics given order statistics, which is only a function of unknown weight matrices $\mathbf{U}^\star$ and $\mathbf{V}^\star$. The second term is the density of order statistics, which relies on the specific base measure f. Thus, we omit the second term and sum over all m^2 paired samples for the first term to arrive at the following objective

$$
\begin{aligned}
&\mathcal{L}(\mathbf{U}, \mathbf{V}) = \\
&\frac{1}{m^2} \sum_{k,l=1}^m \log\left(1 + \exp\left(-(y_k - y_l')(\mathbf{\Theta}_{k_1 k_2} - \mathbf{\Theta}'_{l_1 l_2})\right)\right).
\end{aligned}
\tag{3}
$$

The above loss function is similar to the logistic loss for the pairwise measurements. However, it is nonconvex in both components even for identity activation functions. When feature vectors $\mathbf{x}, \mathbf{z}$ follow the multinomial distribution and activations $\{\phi_i\}_{i=1}^2$ are not present, Chen et al. (2018) estimated the rank-r matrix $\mathbf{U}^\star \mathbf{V}^{\star T}$ as a whole by minimizing (3) with an additional nuclear norm penalty. Our goal is to recover both components $\mathbf{U}^\star$, $\mathbf{V}^\star$, in the presence of nonlinear activation functions, resulting in a challenging nonconvex optimization problem.

We propose to minimize (3) using the gradient descent with constant step size. The iteration is given by

$$\begin{pmatrix}\mathbf{U}^{t+1}\\ \mathbf{V}^{t+1}\end{pmatrix} = \begin{pmatrix}\mathbf{U}^{t}\\ \mathbf{V}^{t}\end{pmatrix} - \eta \begin{pmatrix}\frac{\partial \mathcal{L}(\mathbf{U}^t, \mathbf{V}^t)}{\partial \mathbf{U}}\\ \frac{\partial \mathcal{L}(\mathbf{U}^t, \mathbf{V}^t)}{\partial \mathbf{V}}\end{pmatrix}, \tag{4}$$

with explicit formulas for $\frac{\partial \mathcal{L}(\mathbf{U}^t, \mathbf{V}^t)}{\partial \mathbf{U}}$ and $\frac{\partial \mathcal{L}(\mathbf{U}^t, \mathbf{V}^t)}{\partial \mathbf{V}}$ given in the supplement.

Identifiability. In general, the weight matrices in loss function (3) are not identifiable as the function is bilinear in $\mathbf{U}$, $\mathbf{V}$. For example, when both activation functions are identity, $\mathcal{L}(\mathbf{UQ}, \mathbf{V}(\mathbf{Q}^T)^{-1})$ and $\mathcal{L}(\mathbf{U}, \mathbf{V})$ have the same value for any invertible matrix $\mathbf{Q} \in \mathbb{R}^{r\times r}$, which makes the Hessian at $(\mathbf{U}^\star, \mathbf{V}^\star)$ indefinite. Similarly, for ReLU activation, this phenomenon reappears by letting $\mathbf{Q}$ be any diagonal matrix with positive entries. To resolve this issue, one can use a penalty function $\|\mathbf{U}^T\mathbf{U} - \mathbf{V}^T\mathbf{V}\|_F^2$ to balance two components $\mathbf{U}$ and $\mathbf{V}$ (Yi et al., 2016; Park et al., 2018; Na et al., 2019). Fortunately, in our problem, the identifiability issue disappears when a smooth nonlinear activation is used, such as sigmoid or tanh, although their nonconvexity brings other challenges.

Different from over-parameterized problems in neural networks (Sagun et al., 2017; Li & Liang, 2018; Allen-Zhu et al., 2018), the identifiability issue comes from the redundancy of parameters, which is also observed in inductive matrix completion problem (Zhong et al., 2018). Zhong et al. (2018) showed that by fixing the first row of $\mathbf{U}^\star$, both components are recoverable from the square loss even with ReLU activation. In our problem, when either one of activation functions is ReLU, we use a similar restriction on $\mathbf{U}^\star$ and show that the loss in (3) has positive definite Hessian at $(\mathbf{U}^\star, \mathbf{V}^\star)$, without adding any penalties.

4. Theoretical analysis

In this section, we will show that the ground truth $(\mathbf{U}^\star, \mathbf{V}^\star)$ is a stationary point of the loss (3) and then show that this objective is strongly convex in its neighborhood. Using these two observations, we further establish the local linear convergence rate for iterates in (4). Since the radius of the neighborhood is fixed in terms of $(\mathbf{U}^\star, \mathbf{V}^\star)$, a wart-start initialization can be obtained by a third-order tensor method (see, e.g., Zhong et al., 2017; 2018). In our simulations, due to high computational cost of a tensor method, we recommend a random initialization (Du et al., 2017; Cao & Gu, 2019).

We require two assumptions to establish our main results. The first assumption fixes the scale of weight matrices, while the second one imposes a mild regularity condition.

Assumption 1. *The weight matrices $\mathbf{U}^\star$, $\mathbf{V}^\star$ have rank r and satisfy $\sigma_r(\mathbf{U}^\star) = \sigma_r(\mathbf{V}^\star) = 1$.*

Assumption 2. *Let $\mathcal{D} = \{(y_{ij}, \mathbf{x}_i, \mathbf{z}_j)\}_{i\in[n_1], j\in[n_2]}$ (similarly for $\mathcal{D}'$) be the complete subgraph. We assume*

(a) (boundedness): There exist $\alpha, \beta > 0$ such that, for any $(y, \mathbf{x}, \mathbf{z}) \in \mathcal{D} \cup \mathcal{D}'$, we have $|\mathbf{\Theta}^\star| = |\langle \phi_1(\mathbf{U}^{\star T}\mathbf{x}), \phi_2(\mathbf{V}^{\star T}\mathbf{z})\rangle| \le \alpha$ and $|y| \le \beta$;

(b) (regularity condition): Suppose $(y, \mathbf{x}, \mathbf{z}) \in \mathcal{D}$ and $(y', \mathbf{x}', \mathbf{z}') \in \mathcal{D}'$, we let

$$M_\alpha(\mathbf{\Theta}^\star, \mathbf{\Theta}^{\star\prime}) = \mathbb{E}\left[(y-y')^2 \cdot \psi(2\alpha|y-y'|) \mid \substack{\mathbf{x},\mathbf{z},\\ \mathbf{x}',\mathbf{z}'}\right],$$

where $\psi(x) = \exp(x)/(1+\exp(x))^2$. We assume $M_\alpha(\mathbf{\Theta}^\star, \mathbf{\Theta}^{\star\prime})$ is a continuous, positive two-dimensional function.

Assumption 2 is widely assumed in the analysis of logistic loss function (Chen et al., 2018). In particular, Assumption 2(a) restricts the parametric component $\mathbf{\Theta}^\star$ into a compact set, which controls the range of proximity between two connected nodes. Intuitively, larger α implies a harder estimation problem. We also add boundedness condition on the response y for simplicity. It can be replaced by assuming y to be subexponential (Ning et al., 2017). Boundedness holds deterministically for some distribution in exponential family, such as Bernoulli and Beta, and holds with high probability for a wide range of exponential family distributions, though β may depend on the sample size n_1 and n_2. Assumption 2(b) is the regularity condition, which plays the key role when showing the strong convexity of the population loss at the ground truth. It can be shown to hold for all exponential family distributions with bounded support, and for some unbounded distributions, such as Gaussian and Poisson.

With the above assumptions, our first result shows that the gradient of the population loss at $(\mathbf{U}^\star, \mathbf{V}^\star)$ is zero.

Theorem 3. *The loss* (3) *satisfies* $\mathbb{E}\left[\nabla\mathcal{L}(\mathbf{U}^\star, \mathbf{V}^\star)\right] = \mathbf{0}$.

The next theorem lower bounds the population Hessian at the ground truth. We separate results into two cases: (1) ϕ_1, $\phi_2 \in \{\text{sigmoid}, \tanh\}$; (2) either ϕ_1 or ϕ_2 is ReLU.

Theorem 4. *Suppose Assumptions 1 and 2 hold. We let $\bar{\kappa}(\mathbf{U}^\star) = \prod_{p=1}^r \frac{\sigma_p(\mathbf{U}^\star)}{\sigma_r(\mathbf{U}^\star)}$ and similarly for $\mathbf{V}^\star$. There exist a constant $C > 0$, independent of $\mathbf{U}^\star$ and $\mathbf{V}^\star$, and a constant $\gamma_\alpha > 0$, depending on α only, such that*

(Case 1) If ϕ_1, $\phi_2 \in \{$sigmoid, tanh$\}$, then

$$\lambda_{\min}\left(\mathbb{E}[\nabla^2\mathcal{L}(\mathbf{U}^\star, \mathbf{V}^\star)]\right) \ge \frac{C\gamma_\alpha}{\bar{\kappa}(\mathbf{U}^\star)\bar{\kappa}(\mathbf{V}^\star)\max(\|\mathbf{U}^\star\|_2^2, \|\mathbf{V}^\star\|_2^2)};$$

(Case 2) If either ϕ_1 or ϕ_2 is ReLU, then by fixing the first row of $\mathbf{U}^\star$ (i.e., treating it as known)

$$\lambda_{\min}\left(\mathbb{E}[\nabla^2\mathcal{L}(\mathbf{U}^\star, \mathbf{V}^\star)]\right) \ge \frac{C\gamma_\alpha\|\mathbf{e}_1^T\mathbf{U}^\star\|_{\min}^2}{\bar{\kappa}(\mathbf{U}^\star)\bar{\kappa}(\mathbf{V}^\star)\max(\|\mathbf{U}^\star\|_2^2, \|\mathbf{V}^\star\|_2^2)(1+\|\mathbf{e}_1^T\mathbf{U}^\star\|_2)^2},$$

where $\mathbf{e}_1 = (1, 0, \ldots, 0) \in \mathbb{R}^{d_1}$.

By symmetry one can alternatively fix the first row of $\mathbf{V}^\star$ in the second case. . We realize that the lower bound of population Hessian in Case 2 is smaller than the bound in Case 1. This is due to nonsmoothness and unboundedness of ReLU activation function. In later analysis we will see the sample complexity when using ReLU for either networks will have larger logarithmic factor, while is linear in $d_1 \vee d_2$ in both cases.

Combining Theorem 3 and 4, we obtain that $(\mathbf{U}^\star, \mathbf{V}^\star)$ is a local minimizer of the population loss. In order to characterize how the empirical loss behaves near the ground truth, we study its local geometry via the concentration of the Hessian matrix. We summarize the concentration result next.

Theorem 5 (Concentration of the Hessian matrix)**.** *Suppose Assumptions 1 and 2 hold. For any $s \geq 1$, if $m \wedge n_1 \wedge n_2 \gtrsim s(d_1 + d_2) \left\{\log\left(r(d_1 + d_2)\right)\right\}^{1+2q}$, where $q = 0$ for Case 1 and $q = 1$ for Case 2, then with probability at least $1 - 1/(d_1 + d_2)^s$,*

$$\begin{aligned}
&\|\nabla^2 \mathcal{L}(\mathbf{U}, \mathbf{V}) - \mathbb{E}\left[\nabla^2 \mathcal{L}(\mathbf{U}^\star, \mathbf{V}^\star)\right]\|_2 \\
&\lesssim \beta^3 r^{\frac{3(1-q)}{2}} \left(\|\mathbf{V}^\star\|_F^{3q} + \|\mathbf{U}^\star\|_F^{3q}\right) \\
&\quad \cdot \left(\sqrt{\frac{s \log(d_1 + d_2) \log\left(r(d_1 + d_2)\right)}{m \wedge n_1 \wedge n_2}} + \right. \\
&\qquad \left. \left(\|\mathbf{U} - \mathbf{U}^\star\|_F^2 + \|\mathbf{V} - \mathbf{V}^\star\|_F^2\right)^{\frac{2-q}{4}}\right).
\end{aligned}$$

Combining Theorem 5 and 4, we know that the empirical loss function has positive curvature in a neighborhood of $(\mathbf{U}^\star, \mathbf{V}^\star)$. This local geometry guarantees that GD has the local linear convergence rate, as stated in the next theorem. For notation simplicity, we let $\lambda^\star_{\min} := \lambda_{\min}\left(\mathbb{E}[\nabla^2 \mathcal{L}(\mathbf{U}^\star, \mathbf{V}^\star)]\right)$ be the minimum eigenvalue of the population Hessian and $\lambda^\star_{\max} := \lambda_{\max}\left(\mathbb{E}[\nabla^2 \mathcal{L}(\mathbf{U}^\star, \mathbf{V}^\star)]\right)$ be the maximum eigenvalue. The explicit formula for $\lambda^\star_{\min}$ is given by Theorem 4 and $\lambda^\star_{\max} = \beta^2 r^{1-q}\left(\|\mathbf{V}^\star\|_F^2 + \|\mathbf{U}^\star\|_F^2\right)^q$, as proven in Lemma 11.

Theorem 6 (Local linear convergence rate)**.** *Suppose the conditions of Theorem 5 hold. For any $s \geq 1$ and any initial point $(\mathbf{U}^0, \mathbf{V}^0)$ in the neighborhood*

$$\begin{aligned}
\mathcal{B}(\mathbf{U}^\star, \mathbf{V}^\star) = \Bigg\{&(\mathbf{U}, \mathbf{V}) : \|\mathbf{U} - \mathbf{U}^\star\|_F^2 + \|\mathbf{V} - \mathbf{V}^\star\|_F^2 \\
&\leq \left(\frac{\lambda^\star_{\min}}{C\beta^3 r^{3(1-q)/2}\left(\|\mathbf{U}^\star\|_F^{3q} + \|\mathbf{V}^\star\|_F^{3q}\right)}\right)^{\frac{4}{2-q}}\Bigg\},
\end{aligned}$$

with constant C sufficiently large, the iterates in (4) *with $\eta = 1/\lambda^\star_{\max}$ satisfy*

$$\begin{aligned}
&\|\mathbf{U}^T - \mathbf{U}^\star\|_F^2 + \|\mathbf{V}^T - \mathbf{V}^\star\|_F^2 \\
&\qquad \leq \rho^T\left(\|\mathbf{U}^0 - \mathbf{U}^\star\|_F^2 + \|\mathbf{V}^0 - \mathbf{V}^\star\|_F^2\right),
\end{aligned}$$

with probability at least $1 - T/(d_1 + d_2)^s$, where contraction rate $\rho = 1 - \lambda^\star_{\min}/(7\lambda^\star_{\max})$.

Comparing the above sample complexity with the one for inductive matrix completion problem (Zhong et al., 2018), our rate improves from $d(\log d)^3$ to $d \log d$, when ϕ_1, ϕ_2 are sigmoid or tanh. Moreover, we allow a semiparametric model with two different activation functions, which results in a more involved analysis.

5. Experiments

We show experimental results on synthetic and real-world data. In the following, we call our model nonlinear semiparametric matrix completion (NSMC). We compare NSMC with the baseline nonlinear inductive matrix completion (NIMC) proposed by Zhong et al. (2018), where they assumed the generative model to be Gaussian and minimized the squared loss. The models obtained by removing nonlinear activation functions in NSMC and NIMC are called SMC and IMC, respectively. The local linear convergence result in Theorem 6 is verified in the supplement.

5.1. Robustness to Model Misspecification

We generate synthetic data with model misspecification and compare the performance of estimators given by NSMC, SMC, NIMC and IMC. We fix $d = d_1 = d_2 = 50$, $r = 3$, $n_1 = n_2 = 400$, and use ReLU as the activation function for NSMC and NIMC. For NSMC and SMC, we randomly generate two independent sample sets with $m = 1000$ observations, which are denoted as Ω and Ω'. The observed sample set for NIMC and IMC are set to be the union $\Omega \cup \Omega'$. For NSMC and SMC, we minimize the proposed pseudo-likelihood objective. For NIMC and IMC, we minimize the square loss as suggested by Zhong et al. (2018). We apply gradient descent starting from a random initialization near the ground truth $(\mathbf{U}^\star, \mathbf{V}^\star)$, in order to guarantee convergence of all methods. We evaluate the estimated matrix $\hat{\mathbf{U}}$ using the relative approximation error $\mathcal{E}_{\hat{\mathbf{U}}} = ||\hat{\mathbf{U}} - \mathbf{U}^\star||_F / ||\mathbf{U}^\star||_F$, with $\mathcal{E}_{\hat{\mathbf{V}}}$ defined similarly. We also evaluate the performance of a solution $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ on recovering the parametric component $\hat{\mathbf{\Theta}}$ using relative test error $\mathcal{E}_{\hat{\mathbf{\Theta}}} = \sqrt{\sum_{(\mathbf{x},\mathbf{z}) \in \Omega_t} (\hat{\mathbf{\Theta}} - \mathbf{\Theta}^\star)^2 / \sum_{(\mathbf{x},\mathbf{z}) \in \Omega_t} \mathbf{\Theta}^{\star 2}}$, where $\hat{\mathbf{\Theta}} = \langle \phi(\hat{\mathbf{U}}^T \mathbf{x}), \phi(\hat{\mathbf{V}}^T \mathbf{z}) \rangle$, $\mathbf{\Theta}^\star = \langle \phi(\mathbf{U}^{\star T} \mathbf{x}), \phi(\mathbf{V}^{\star T} \mathbf{z}) \rangle$, and Ω_t is a newly sampled test data set. For each setting below, we report results averaged over 10 runs.

Gaussian model. We introduce model misspecification by sampling y from $y \sim \mathcal{N}\left((1-\tau)^2 \cdot \mathbf{\Theta}, (1-\tau)^2\right)$. Parameter τ is introduced to modify the impact of model misspecification. We summarize the relative errors in Table 1 and Figure 1. When $\tau = 0$, there is no model misspecification and NSMC and NIMC achieve comparable relative approximation errors. As τ increases, the relative approximation errors of NIMC grow rapidly due to the increase of model misspecification. However, NSMC gives robust estimations.

SMC and IMC serve as bilinear modeling baselines that fail to learn in the nonlinear embedding setting.

Table 1: Relative error in the Gaussian model.

Method	$\tau = 0$			$\tau = 0.2$			$\tau = 0.4$		
	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$
NSMC	**0.0291**	**0.0299**	**0.0383**	**0.0322**	**0.0326**	**0.0427**	**0.0365**	**0.0365**	**0.0468**
SMC	0.8163	0.8603	0.9954	0.8139	0.8557	0.9938	0.7970	0.8338	0.9849
NIMC	0.0425	0.0410	0.0527	0.2140	0.1935	0.3633	0.4315	0.3638	0.6377
IMC	0.6209	0.6191	1.0899	0.6495	0.6349	1.0503	0.6981	0.6681	1.0289

Binomial model. We sample $y \sim B(N_B, \frac{\exp(\mathbf{\Theta})}{1+\exp(\mathbf{\Theta})})$ and apply NSMC and SMC with original attributes y. For NIMC and IMC, we first do variance-stabilizing transformation $\tilde{y} = \arcsin\left(\frac{y}{N_B}\right)$ as the data preprocessing step, inspired by what people might do for non-Gaussian data in practical applications. From Table 2, NSMC achieves the best estimating result in each setting, while other methods fail to learn the embeddings with a binomial model.

Table 2: Relative error in the Binomial model.

Method	$N_B = 100$			$N_B = 200$			$N_B = 500$		
	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$
NSMC	**0.0354**	**0.0352**	**0.0464**	**0.0329**	**0.0327**	**0.0441**	**0.0301**	**0.0297**	**0.0381**
SMC	0.8629	0.8896	0.9956	0.9402	0.9493	0.9988	0.9843	0.9873	0.9998
NIMC	0.8221	0.6151	0.9364	0.8212	0.6201	0.9248	0.8236	0.6138	0.9259
IMC	0.8137	0.7934	1.0044	0.8302	0.7781	1.0038	0.8205	0.7891	1.0078

Poisson model. We generate $y \sim Pois(\exp(\mathbf{\Theta}))$, where the activation functions are $\phi_1 = \text{ReLU}$ and $\phi_2 \in \{\text{ReLU}, \text{sigmoid}, \text{tanh}\}$. Due to model misspecification, we apply transformation $\tilde{y} = \sqrt{y}$ for NIMC and IMC. The activation function of NIMC is set to be the same as ϕ_2. We see from Table 3 that NSMC achieves the best estimating result, while other methods fail to recover the parameters.

Table 3: Relative error in the Poison model.

Method	ReLU+ReLU			ReLU+sigmoid			ReLU+tanh		
	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$	$\mathcal{E}_{\hat{\mathbf{U}}}$	$\mathcal{E}_{\hat{\mathbf{V}}}$	$\mathcal{E}_{\hat{\mathbf{\Theta}}}$
NSMC	**0.0691**	**0.0718**	**0.0975**	**0.0661**	**0.0617**	**0.0631**	**0.0442**	**0.0457**	**0.0727**
SMC	0.3696	0.3852	0.5559	0.7855	0.8229	0.9757	0.2812	0.3019	0.4500
NIMC	2.2479	2.3282	10.7018	1.3024	0.4078	1.4877	0.5203	0.2522	0.5595
IMC	1.5717	1.5889	5.7169	0.5745	0.6368	1.0922	0.3604	0.3847	1.1643

5.2. Clustering of Embeddings

We generate synthetic data with clustered embeddings and compare the performance of NSMC and NIMC on learning the true embedding clustering. We fix $d = d_1 = d_2 = 30$, $r = 2$, $n_1 = n_2 = 400$, and choose tanh as the activation function. We generate features $\mathbf{x}$ and $\mathbf{z}$ independently from a Gaussian mixture model with four components, resulting in the ground-truth embedding clustering with four components. We sample y from a binomial model with $N_B = 20$. We fix observed sample size $m = 1000$ and apply NSMC and NIMC to get the estimated $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, respectively. We plot the top 2 left singular vectors $(\hat{\iota}_1, \hat{\iota}_2)$ of $\phi_1(\hat{\mathbf{U}}^T\mathbf{x})$ for NSMC and NIMC, respectively, where the points are colored according to the ground-truth clustering. We also plot the top 2 left singular vectors $(\iota_1^\star, \iota_2^\star)$ of the ground-truth embeddings $\phi_1(\mathbf{U}^{\star T}\mathbf{x})$. Similar plots for feature $\mathbf{z}$ are shown as well. We see from Figure 2 that NIMC fails to find the ground-truth embeddings due to model misspecification, while NSMC gives robust estimation and recovers the ground-truth embeddings.

To quantitatively evaluate the performance, we apply the k-means clustering to the left singular vectors. We define the clustering error following Zhong et al. (2018) as

$$\frac{2}{n(n-1)}\left(\sum_{(i,j):\aleph_i^\star=\aleph_j^\star} 1_{\aleph_i\neq\aleph_j} + \sum_{(i,j):\aleph_i^\star\neq\aleph_j^\star} 1_{\aleph_i=\aleph_j}\right),$$

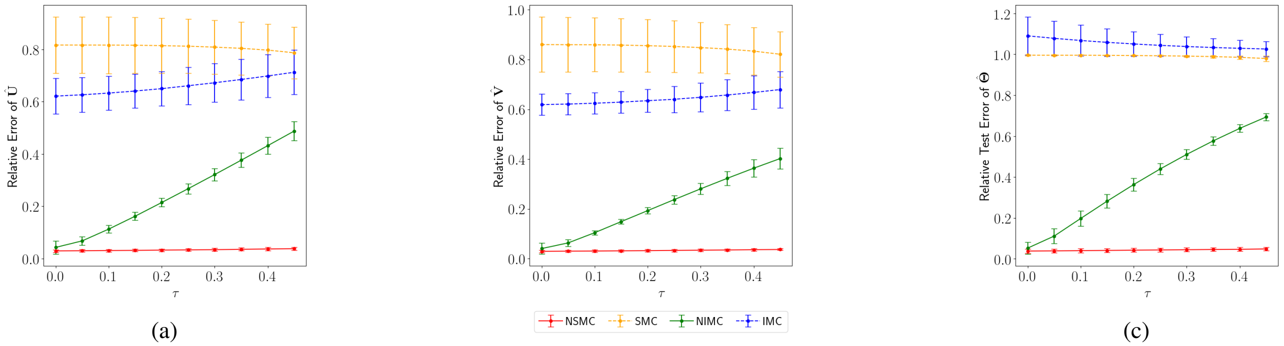
where $\aleph^\star$ is the ground-truth clustering and $\aleph$ is the predicted clustering. As a result, NIMC attains clustering error 0.0596 and 0.1725 for $\mathbf{x}$ and $\mathbf{z}$ respectively. NSMC achieves a better performance with clustering error 0.0196 and 0.0147 for $\mathbf{x}$ and $\mathbf{z}$ respectively.

5.3. Semi-supervised Clustering

We further illustrate the superior performance of NSMC over NIMC with real-world data. Following the experimental setting in Zhong et al. (2018), we apply NSMC and NIMC to a semi-supervised clustering problem, where we only have one kind of features, $\mathbf{x} \in \mathbb{R}^{d_1}$, on a set of items. The edge attribute $y_{ij} = 1$, if the i-th item and j-th item are similar, and $y_{ij} = 0$, if they are dissimilar. To apply NSMC and NIMC, we set $\mathbf{x} = \mathbf{z}$, $\phi_1 = \phi_2 = \phi$, and assume $\mathbf{U}^\star = \mathbf{V}^\star$. We initialize $\mathbf{U}^0 = \mathbf{V}^0$ as the same random Gaussian matrix and apply gradient descent to ensure $\mathbf{U}^t = \mathbf{V}^t$ during training. After training, we apply k-means clustering to the top r left singular vectors of $\phi(\hat{\mathbf{U}}^T\mathbf{x})$. We follow Zhong et al. (2018) and again use the clustering error defined by (5.2). We set the activation function ϕ to be tanh for all data sets. For NSMC, we first uniformly sample two independent sets of items with $n_1 = n_2 = 1000$. Then we generate independent observation sets Ω and Ω' with size $m = 5000$. For NIMC, the observed dataset is set to be the union $\Omega \cup \Omega'$. We consider three datasets: *Mushroom*, *Segment* and *Covtype* (Dua & Graff, 2017), and regard items with the same label as similar ($y_{ij} = 1$). *Covtype* dataset is subsampled first to balance the size of each cluster. As shown in Table 4, for linear separable dataset *Mushroom*, both NSMC and NIMC achieve perfect clustering. For the other two datasets, NSMC achieves better clustering results than NIMC.

Table 4: Relative error in the Poison model.

Dataset	d	r	NIMC	NSMC
Mushroom	112	2	**0**	**0**
Segment	19	7	0.0971	**0.0427**
Covtype	54	7	0.1931	**0.1373**

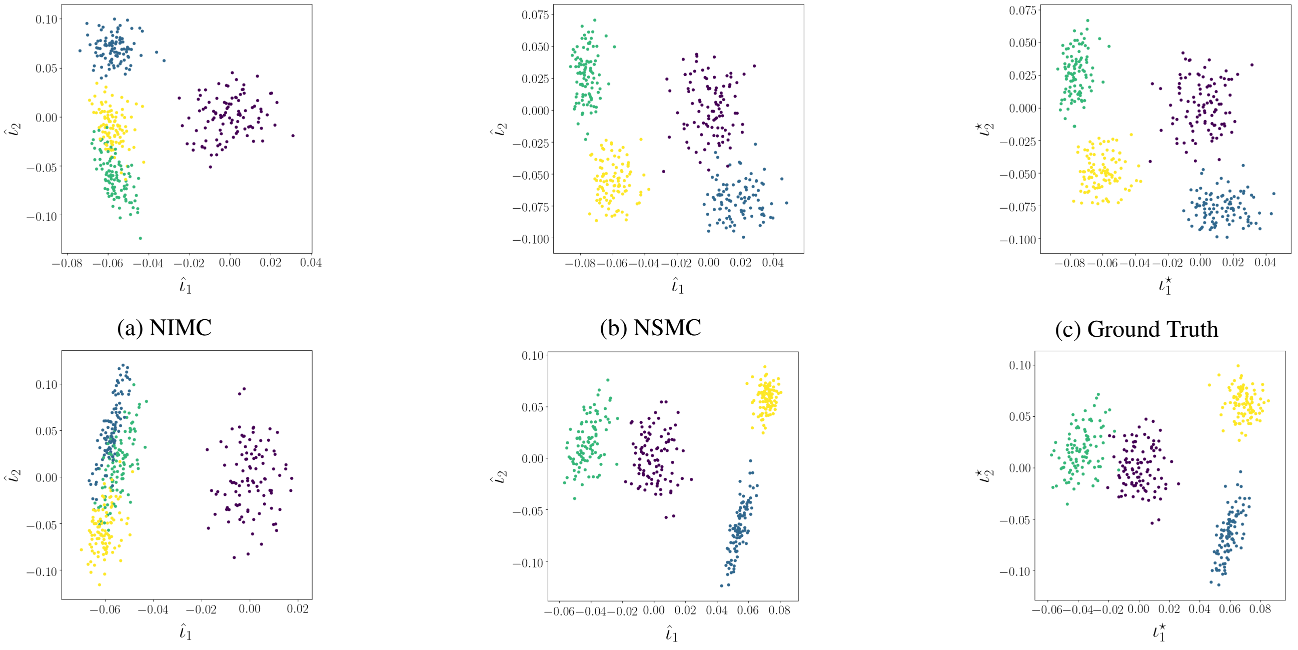


(a)

(b)

(c)

Figure 1: Relative Error of NSMC, SMC, NIMC and IMC. The plot shows how relative error of estimations given by each method varies with parameter τ, which introduces model misspecification in the Gaussian model. We see that NSMC gives accurate and robust estimation, while NIMC suffers from model misspecification. SMC, IMC fail to learn the non-linear embeddings and give unsatisfactory estimations for all τ.



(a) NIMC

(b) NSMC

(c) Ground Truth

(d) NIMC

(e) NSMC

(f) Ground Truth

Figure 2: The comparison of learned embeddings based on NIMC and NSMC, with the ground truth embeddings. The first row shows embeddings of $\mathbf{x}$, while the second row shows embeddings of $\mathbf{z}$. The points are colored according to the ground-truth clustering.

6. Conclusion

We studied the nonlinear bipartite graph representation learning problem. We formalized the representation learning problem as a statistical parameter estimation problem in a semiparametric model. In particular, the edge attributes, given node features, are assumed to follow an exponential family distribution with unknown base measure. The parametric component of the model is assumed to be the proximity of outputs of one-layer neural network, whose inputs are node representations. In this setting, learning embedding vectors is equivalent to estimating two low-rank weight matrices $(\mathbf{U}^\star, \mathbf{V}^\star)$. Using the rank-order decomposition technique, we proposed a pseudo-likelihood function, and proved that GD with constant step size achieves local linear convergence rate. The sample complexity is linear in dimensions up to a logarithmic factor, which matches existing results in matrix completion. However, our estimator is robust to model misspecification within exponential family due to the adaptivity to the base measure. We also provided numerical simulations and real experiments to corroborate the main theoretical results, which demonstrated superior performance of our method over existing approaches.

One potential extension is to consider a more general distribution for node representations. For example, when node representations follow a heavy-tailed distribution, it is not clear whether we can still recover $(\mathbf{U}^{\star}, \mathbf{V}^{\star})$ with the same convergence rate. In addition, using two-layer or even deep neural networks for encoders in our semiparametric model, while still providing theoretical guarantee is another interesting extension.

Acknowledgements

This work is partially supported by the William S. Fishman Faculty Research Fund at the University of Chicago Booth School of Business. This work was completed in part with resources supported by the University of Chicago Research Computing Center. Sen Na is supported by Harper Dissertation Fellowship.

References

Abernethy, J., Bach, F., Evgeniou, T., and Vert, J.-P. Low-rank matrix factorization with attributes. *arXiv preprint cs/0611124*, 2006.

Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., and Smola, A. J. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 37–48. ACM, 2013.

Al Hasan, M. and Zaki, M. J. A survey of link prediction in social networks. In *Social network data analytics*, pp. 243–275. Springer, 2011.

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*, 2018.

Badrinarayanan, V., Kendall, A., and Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Baldin, N. and Berthet, Q. Optimal link prediction with matrix logistic regression. *arXiv preprint arXiv:1803.07054*, 2018.

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591, 2002.

Bell, M. G. and Iida, Y. *Transportation network analysis*. 1997.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Berg, R. v. d., Kipf, T. N., and Welling, M. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

Bhagat, S., Cormode, G., and Muthukrishnan, S. Node classification in social networks. In *Social network data analytics*, pp. 115–148. Springer, 2011.

Bunke, H. and Messmer, B. T. Efficient attributed graph matching and its application to image analysis. In *International Conference on Image Analysis and Processing*, pp. 44–55. Springer, 1995.

Cao, Y. and Gu, Q. Tight sample complexity of learning one-hidden-layer convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 10611–10621, 2019.

Cavallari, S., Zheng, V. W., Cai, H., Chang, K. C.-C., and Cambria, E. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 377–386. ACM, 2017.

Chen, Y. and Candès, E. J. The projected power method: An efficient algorithm for joint alignment from pairwise differences. *Communications on Pure and Applied Mathematics*, 71(8):1648–1714, 2018.

Chen, Y. and Goldsmith, A. J. Information recovery from pairwise measurements. In *2014 IEEE International Symposium on Information Theory*, pp. 2012–2016. IEEE, 2014.

Chen, Y. and Suh, C. Spectral mle: Top-k rank aggregation from pairwise comparisons. In *International Conference on Machine Learning*, pp. 371–380, 2015.

Chen, Y., Kamath, G., Suh, C., and Tse, D. Community recovery in graphs with locality. In *International Conference on Machine Learning*, pp. 689–698, 2016.

Chen, Y., Yang, Z., Xie, Y., and Wang, Z. Contrastive learning from pairwise measurements. In *Advances in Neural Information Processing Systems*, pp. 10909–10918, 2018.

Chen, Y., Fan, J., Ma, C., Wang, K., et al. Spectral method and regularized mle are both optimal for top-k ranking. *The Annals of Statistics*, 47(4):2204–2235, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Fan, J., Liu, H., Ning, Y., and Zou, H. High dimensional semiparametric latent graphical model for mixed data. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 79(2):405–421, 2017. ISSN 1369-7412. doi: 10.1111/rssb.12168. URL `https://doi.org/10.1111/rssb.12168`.

Fengler, M. R. *Semiparametric modeling of implied volatility*. Springer Finance. Springer-Verlag, Berlin, 2005. ISBN 978-3-540-26234-3; 3-540-26234-2.

Fiorio, C. A topologically consistent representation for image analysis: the frontiers topological graph. In *International Conference on Discrete Geometry for Computer Imagery*, pp. 151–162. Springer, 1996.

Fortunato, S. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

Fosdick, B. K. and Hoff, P. D. Testing and modeling dependencies between a network and nodal attributes. *J. Amer. Statist. Assoc.*, 110(511):1047–1056, 2015. ISSN 0162-1459. doi: 10.1080/01621459.2015.1008697. URL `https://doi.org/10.1080/01621459.2015.1008697`.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Goyal, P. and Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

Graepel, T., Goutrie, M., Krüger, M., and Herbrich, R. Learning on graphs in the game of go. In *International Conference on Artificial Neural Networks*, pp. 347–352. Springer, 2001.

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

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017a.

Hamilton, W. L., Ying, R., and Leskovec, J. Representation learning on graphs: Methods and applications. *arxiv: 1709.05584*, 2017b.

Higham, D. J., Rašajski, M., and Pržulj, N. Fitting a geometric graph to a protein–protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.

Hsu, D., Kakade, S., and Zhang, T. A tail inequality for quadratic forms of subgaussian random vectors. *Electron. Commun. Probab.*, 17(52):1–6, 2012. ISSN 1083-589X. doi: 10.1214/ECP.v17-2079. URL `http://ecp.ejpecp.org/article/view/2079`.

Jain, P. and Dhillon, I. S. Provable inductive matrix completion. *arXiv preprint arXiv:1306.0626*, 2013.

Jannach, D., Resnick, P., Tuzhilin, A., and Zanker, M. Recommender systems—beyond matrix completion. *Communications of the ACM*, 59(11):94–102, 2016.

Kang, Z., Peng, C., and Cheng, Q. Top-n recommender system via matrix completion. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Li, R. and Liang, H. Variable selection in semiparametric regression modeling. *Ann. Stat.*, 36(1): 261–286, 2008. ISSN 0090-5364. doi: 10.1214/009053607000000604. URL `http://dx.doi.org/10.1214/009053607000000604`.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pp. 8157–8166, 2018.

Liben-Nowell, D. and Kleinberg, J. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7): 1019–1031, 2007.

Ma, M., Na, S., Xu, C., and Fan, X. The graph-based broad behavior-aware recommendation system for interactive news. *arXiv preprint arXiv:1812.00002*, 2018.

Ma, Z., Ma, Z., and Yuan, H. Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research (to appear)*, 2019.

Magnus, J. R. The moments of products of quadratic forms in normal variables. *Statist. Neerlandica*, 32(4):201–210, 1978. ISSN 0039-0402. doi: 10.1111/j.1467-9574.1978.tb01399.x. URL `https://doi.org/10.1111/j.1467-9574.1978.tb01399.x`.

Menon, A. K. and Elkan, C. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 437–452. Springer, 2011.

Na, S., Kolar, M., and Koyejo, O. Estimating differential latent variable graphical models with applications to brain connectivity. *arXiv preprint arXiv:1909.05892*, 2019.

Nassar, M. Hierarchical bipartite graph convolution networks. *arXiv preprint arXiv:1812.03813*, 2018.

Negahban, S., Oh, S., Thekumparampil, K. K., and Xu, J. Learning from comparisons and choices. *The Journal of Machine Learning Research*, 19(1):1478–1572, 2018.

Ning, Y., Zhao, T., and Liu, H. A likelihood ratio framework for high-dimensional semiparametric regression. *Ann. Statist.*, 45(6):2299–2327, 2017. ISSN 0090-5364. doi: 10.1214/16-AOS1483. URL `https://doi.org/10.1214/16-AOS1483`.

Pananjady, A., Mao, C., Muthukumar, V., Wainwright, M. J., and Courtade, T. A. Worst-case vs average-case design for estimation from fixed pairwise comparisons. *arXiv preprint arXiv:1707.06217*, 2017.

Park, D., Kyrillidis, A., Caramanis, C., and Sanghavi, S. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM Journal on Imaging Sciences*, 11(4):2165–2204, 2018.

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

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467. ACM, 2018.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Scott, J. Social network analysis. *Sociology*, 22(1):109–127, 1988.

Si, S., Chiang, K.-Y., Hsieh, C.-J., Rao, N., and Dhillon, I. S. Goal-directed inductive matrix completion. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1165–1174. ACM, 2016.

Stein, C. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. pp. 583–602, 1972.

Taskar, B., Wong, M.-F., Abbeel, P., and Koller, D. Link prediction in relational data. In *Advances in neural information processing systems*, pp. 659–666, 2004.

Tropp, J. A. et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Vershynin, R. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018. ISBN 978-1-108-41519-4. doi: 10.1017/9781108231596. URL `https://doi.org/10.1017/9781108231596`. An introduction with applications in data science, With a foreword by Sara van de Geer.

Weyl, H. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Math. Ann.*, 71(4):441–479, 1912. ISSN 0025-5831. doi: 10.1007/BF01456804. URL `https://doi.org/10.1007/BF01456804`.

Wu, Y., Liu, H., and Yang, Y. Graph convolutional matrix completion for bipartite edge prediction. In *KDIR*, pp. 49–58, 2018.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

Yang, C., Liu, Z., Zhao, D., Sun, M., and Chang, E. Network representation learning with rich text information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Yi, X., Park, D., Chen, Y., and Caramanis, C. Fast algorithms for robust PCA via gradient descent. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4152–4160. Curran Associates, Inc., 2016.

Zha, H., He, X., Ding, C., Simon, H., and Gu, M. Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pp. 25–32, 2001.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 4140–4149. JMLR. org, 2017.

Zhong, K., Song, Z., Jain, P., and Dhillon, I. S. Nonlinear inductive matrix completion based on one-layer neural networks. *arXiv preprint arXiv:1805.10477*, 2018.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., and Sun, M. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.