Learning Embedding Representations in High Dimensions

Golara Ahmadi Azar*, Melika Emami ‡*, Alyson Fletcher*, and Sundeep Rangan†
Email: golazar@g.ucla.edu, emami@ucla.edu, akfletcher@g.ucla.edu, srangan@nyu.edu
* Electrical and Computer Engineering Department, UCLA, Los Angeles, CA
† Electrical and Computer Engineering Department, NYU, Brooklyn, NY

2024 58th Annual Conference on Information Sciences and Systems (CISS) | 979-8-3503-6929-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/CISS59072.2024.10480173

***Abstract*—Embeddings are a basic initial feature extraction step in many machine learning models, particularly in natural language processing. An embedding attempts to map data tokens to a low-dimensional space where similar tokens are mapped to vectors that are close to one another by some metric in the embedding space. A basic question is how well can such embedding be learned? To study this problem, we consider a simple probability model for discrete data where there is some "true" but unknown embedding where the correlation of random variables is related to the similarity of the embeddings. Under this model, it is shown that the embeddings can be learned by a variant of low-rank approximate message passing (AMP) method. The AMP approach enables precise predictions of the accuracy of the estimation in certain high-dimensional limits. In particular, the methodology provides insight on the relations of key parameters such as the number of samples per value, the frequency of the terms, and the strength of the embedding correlation on the probability distribution. Our theoretical findings are validated by simulations on both synthetic data and real text data.**

***Index Terms*—AMP, Poisson channel, State Evolution, Embedding learning.**

I. Introduction

Embeddings are widely-used in machine learning tasks, particularly text processing [1]. In this work, we study embedding of pairs of discrete random variables, (X_1, X_2), where $X_1 \in [m] := \{1, \ldots, m\}$ and $X_2 \in [n] := \{1, \ldots, n\}$. For example, in word embeddings, X_1 could represent a target word, and X_2 a context word (e.g., a second word found close to the target word) [2]. By an *embedding*, we mean a pair of mappings of the form:

$$X_1 = i \mapsto \boldsymbol{u}_i, \quad X_2 = j \mapsto \boldsymbol{v}_j, \tag{1}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j \in \mathbb{R}^d$. The embedding thus maps each value of the random variable to an associated d-dimensional vector. The dimension d is called the *embedding dimension*.

Typically, (see e.g., [2]), we try to learn embeddings such that $\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j$ is large when the pair $(X_1, X_2) = (i, j)$ occurs more frequently. Many algorithms have been proposed for training such embeddings [2]–[5]. While these algorithms have been successful in practice, precise convergence results are difficult to obtain. At root, we wish to understand how well can embeddings be learned? For example, questions include: how well do the correlations, $\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j$ of learned embeddings predict the underlying correlation of events $X_i = i$ and $X_2 = j$. How do these predictions depend on the number of data samples available and embedding dimension?

To study these problems, we propose a simple model for the joint distribution of (X_1, X_2) where

$$\log\left[\frac{P(X_1 = i, X_2 = j)}{P(X_1 = i)P(X_2 = j)}\right] \approx \frac{1}{\sqrt{m}}\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j, \tag{2}$$

for some true embedding vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$. The property (2) indicates that the log correlation of the events that $X_1 = i$ and $X_2 = j$ is proportional to the vector correlation $\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j$ in the embedding space. Importantly, the model also has parameters r_i^u and r_j^v (called the *bias parameters*) that describe the marginal distributions $P(X_1 = i)$ and $P(X_2 = j)$. This feature captures the fact that there is a high variability in the probabilities of terms.

The problem of learning the embeddings can then be thought of as estimating the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ from samples (x_1, x_2). We show that the estimation problem is equivalent to a low-rank matrix factorization problem. Low-rank matrix factorization [6], and variants of such algorithms are often used in learning embeddings [2], [7]. In this work, we consider a variant of low-rank approximate message passing (AMP) method. Several AMP methods are available for low-rank matrix factorization (AMP-KM [8], IterFac [9], Low-rank AMP [10]). In this work, we assume that the number of samples where $(x_1, x_2) = (i, j)$ is distributed as a Poisson random variable with rate proportional to the joint probability, $P(X_1 = i, X_2 = j)$. Under this assumption, we show that we can apply variant of [10], [11] that allows low-rank estimates under general (possibly, non-Gaussian) measurements. The method is modified to incorporate the bias terms and we call the resulting method biased low-rank AMP.

The main advantage of the AMP approach is the precise predictions of the algorithm performance in certain high-dimensional limits. Specifically, we consider the case where the embedding dimension d is fixed while the number of values, m and n, for X_1 and X_2 grow to infinity with m/n constant. Under this assumption, we show that the joint distribution of the true embedding vectors and their estimates can be exactly characterized. The characterization is given by a *state evolution* similar to other AMP algorithms [12], [13]. From this joint distribution, one can evaluate various performance metrics such as mean squared error (MSE) or overlap of the

‡ Now at Optum AI, work done while at UCLA.

true and learned embedding vectors as well as the error in the learned joint probability distribution. The performance, in turn, can be related to key parameters such as the number of data samples per outcome (i,j), the relative frequency of terms, and strength of the dependence of the embedding correlation $\boldsymbol{u}_i^\mathsf{T}\boldsymbol{v}_j$ on the correlation of events $X_1 = i$ and $X_2 = j$.

Prior work: Since the introduction of AMP algorithms by [12], there has been a vast body of literature in their applications to various statistical estimation problems [14]. Early AMP methods for low-rank estimation include [8] and [15]. IAMP methods were proven to be optimal for the case of sparse PCA [16]. The work [17] applied AMP to the stochastic block model which is a popular statistical model for the large-scale structure of complex networks. Authors of [18] address the shortcomings of classical PCA in the high dimensional and low SNR regime. They use an AMP algorithm to solve the non-convex non-negative PCA problem. In [19], authors consider a general form of the problem at hand and provide the MMSE that is in principle achievable in any computational time. Specifically relevant to our study, [10], [11] present a framework to address the constrained low-rank matrix estimation assuming a general prior on the factors, and a general output channel (a biased Poisson channel in our case) through which the matrix is observed. Noting that state evolution is uninformative when the algorithm is initialized near an unstable fixed point, [20] proposes a new analysis of AMP that allows for spectral initializations. The main contribution of the current work is to modify and apply these methods to the embedding learning problem.

For space considerations, all assumptions, proofs, and simulations details are omitted and provided in a full paper [21].

II. Problem Formulation

A. Joint Density Model for the Embedding

As stated in the introduction, we consider embeddings of pairs of discrete random variables (X_1, X_2) with $X_1 \in [m]$ and $X_2 \in [n]$ for some m and n. Let $P_i^{(1)} = P(X_1 = i)$ and $P_j^{(2)} = P(X_2 = j)$ denote the marginal distributions and $P_{ij} = P(X_i = i, X_2 = j)$ denote the joint distribution. We assume the joint distribution has the form,

$$P_{ij} = C \exp\left(\frac{1}{\sqrt{m}}\boldsymbol{u}_i^\mathsf{T}\boldsymbol{v}_j + s_i^u + s_j^v\right), \tag{3}$$

where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^d$ are some "true" embedding vectors, s_i^u and s_j^v are scalars, and $C > 0$ is a normalization constant. It can be verified that, for large m, the marginal distributions of X_1 are X_2 satisfy:

$$\log P_i^{(1)} = C_1 + s_i^u + O(1/\sqrt{m}), \tag{4a}$$

$$\log P_i^{(2)} = C_2 + s_j^v + O(1/\sqrt{m}), \tag{4b}$$

where C_1 and C_2 are constants. Hence, s_i^u and s_j^v, which we will call the *bias* terms, represent the log likelihoods of the values. Also, the PMF (3) satisfies the property

$$\log\left[\frac{P_{ij}}{P_i^{(1)}P_j^{(2)}}\right] = \frac{1}{\sqrt{m}}\boldsymbol{u}_i^\mathsf{T}\boldsymbol{v}_j + O(1/m), \tag{5}$$

Hence, the similarity $\boldsymbol{u}_i^\mathsf{T}\boldsymbol{v}_j$ represents the log of the correlation of the events that $X_1 = i$ and $X_2 = j$.

B. Poisson Measurements

We can try to learn an embedding by fitting a model of the form (3) to the data. To this end, suppose we are given a set of samples, (x_1^t, x_2^t), $t = 1, \ldots, N$. Let

$$Z_{ij} = \left|\{t \mid (x_1^t = i, x_2^t = j)\}\right|, \tag{6}$$

which are the number of instances where $(X_1, X_2) = (i,j)$. If we assume that the samples are independent and identicaly distributed (i.i.d.), with PMF (3) and the number of samples, N, is Poisson distributed, then the measurements Z_{ij} will be independent with distributions,

$$\begin{aligned} Z_{ij} &\sim \text{Poisson}(\lambda_{ij}), \\ \lambda_{ij} &= \lambda_0 \exp\left(\frac{1}{\sqrt{m}}\mathbf{u}_i\mathbf{v}_j^\mathsf{T} + s_i^u + s_j^v\right), \end{aligned} \tag{7}$$

where $\lambda_0 = C\mathbb{E}(N)$.

III. AMP-Based Estimation

A. Bias vector estimation

Our problem is to estimate the embedding vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ and the biases s_i^u and s_j^v from the model (3) and measurements Z_{ij}. In principle, one can use maximum likelihood (ML) estimation. In this work, we consider a simpler two-step estimation method that is easier to analyze.

In the first step, we estimate the bias terms s_i^u and s_j^v. Define

$$r_i^u := e^{-s_i^u}, \quad r_j^v := e^{-s_j^v}. \tag{8}$$

Our plan is to estimate r_i^u and r_j^v and then estimate s_i^u and s_j^v from (8). Note that, by adjusting the bias terms s_i^u or s_j^v, we will assume in the sequel, without loss of generality, that in the model (7)

$$\lambda_0 = 1, \quad \frac{1}{m}\sum_{i=1}^m e^{s_i^u} = 1. \tag{9}$$

Under this assumption, we propose to estimate the bias terms with:

$$\widehat{s}_i^u = -\log(\widehat{r}_i^u), \quad \widehat{s}_j^v = -\log(\widehat{r}_j^v), \tag{10}$$

where $\widehat{r}_i^u$ and $\widehat{r}_j^v$ are estimates of r_i^u and r_j^v given by:

$$\frac{1}{\widehat{r}_i^u} = \frac{m}{Z_{\text{tot}}}\sum_{j=1}^n Z_{ij}, \quad \frac{1}{\widehat{r}_j^v} = \frac{n}{Z_{\text{tot}}}\sum_{i=1}^m Z_{ij} \tag{11}$$

and

$$Z_{\text{tot}} := \sum_{i=1}^m\sum_{j=1}^n Z_{ij}. \tag{12}$$

Note that Z_{tot} represents the total number of measurements. Also, $1/\widehat{r}_i^u$ is proportional to $\sum_j Z_{ij}$, which is simply the

relative frequency of the event $X_1 = i$. Similarly, $1/\widehat{r}_j^v$ is the relative frequency of the event $X_2 = j$.

B. Regularized Maximum Likelihood

Having estimated the bias terms, we next estimate the embedding vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$. Let U and V be the matrices with rows $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$. Consider the loss function,

$$L_0(U,V) := -\sum_{ij} \log P_{\rm out}\left(Z_{ij}|\frac{1}{\sqrt{m}}\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j + s_i^u + s_j^v\right) + \phi_u(U) + \phi_v(V), \tag{13}$$

where $P_{\rm out}(z|\log\lambda) := e^{-\lambda}\lambda^z/z!$ is the Poisson distribution, and $\phi_u(U)$ and $\phi_v(V)$ are regularizers on the matrices of embedding vectors. Ideally, we would obtain estimates for U and V by minimizing this loss:

$$\widehat{U},\widehat{V} = \arg\min_{U,V} L_0(U,V). \tag{14}$$

This minimization would correspond to performing a regularized ML estimation assuming the Poisson model (7).

We will assume the regularizers are row-wise separable meaning

$$\phi_u(U) = \sum_{i=1}^m g_u(u_i), \quad \phi_v(V) = \sum_{j=1}^n g_v(v_j), \tag{15}$$

for some functions $g_u(\cdot)$ and $g_v(\cdot)$. For example, we can use a square norm regularizers such as:

$$g_u(u_i) := \frac{\lambda_u}{2}\|u_i\|^2, \quad g_v(v_j) := \frac{\lambda_v}{2}\|v_j\|^2, \tag{16}$$

for normalization constants λ_u and λ_v. Regularizers can also be used to impose sparsity.

Of course, the loss function (13) depends on the bias terms s_i^u and s_j^v, which are not known. In place, we will use the estimates $\widehat{s}_i^u$ and $\widehat{s}_j^v$ from (10).

C. Biased Low-Rank AMP

One possible approach to minimizing the loss (13) is to use the low-rank AMP method of [10], [11]. This method considers general loss functions of the form

$$L_0(U,V) := -\sum_{ij} \log P_{\rm out}\left(Z_{ij}|\frac{1}{\sqrt{m}}\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j\right) + \phi_u(U) + \phi_v(V). \tag{17}$$

The method [10], [11] permits general probability mappings $P(Z_{ij}|\cdot)$ (these are called the "output channels"). However, there is no direct method to incorporate the bias terms s_i^u and s_j^v that appear in the likelihood in (13).

We will show that, for the particular structure of the Poisson channel, we can modify the low-rank AMP method [10], [11]. The key to the low-rank AMP method is to take a quadratic approximation of the log likelihood of the output channel. We apply a similar approach here and first compute the so-called Fisher score functions:

$$Y_{ij} := \frac{\partial}{\partial u}\log P_{\rm out}(Z_{ij}|u + s_i^u + s_j^v)\Big|_{u=0} = r_i^u r_j^v\left(Z_{ij} - \frac{1}{r_i^u r_j^v}\right). \tag{18}$$

Also, let Δ_{ij} denote the so-called inverse Fisher information:

$$\frac{1}{\Delta_{ij}} := \mathbb{E}\left[\left(\frac{\partial}{\partial u}\log P_{\rm out}(Z_{ij}|u + s_i^u + s_j^v)\right)^2\right] = \frac{1}{r_i^u r_j^v} \tag{19}$$

Next, let $M_{ij} := (\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}_j)/\sqrt{m}$. For large m, M_{ij} is small, so we can take a Taylor's approximation,

$$\log P_{\rm out}(Z_{ij}|M_{ij} + s_i^u + s_j^v) \approx Y_{ij}M_{ij} - \frac{1}{2\Delta_{ij}}M_{ij}^2 + \text{const}. \tag{20}$$

To write this as a quadratic, define the scaled variables:

$$A := R_u^{-1/2}U, \quad B := R_v^{-1/2}V, \quad \widetilde{Y} := R_u^{-1/2}YR_v^{-1/2}, \tag{21}$$

Then, using (18), (19), (20) and some simple algebra shows that the log likelihood can be written in a quadratic form:

$$-\log P_{\rm out}(Z_{ij}|M_{ij} + s_i^u + s_j^v) \approx \frac{1}{2}\left|\widetilde{Y}_{ij} - \frac{1}{\sqrt{m}}[AB^{\mathsf{T}}]_{ij}\right| + \text{const}. \tag{22}$$

Hence, we can approximate the loss function (13) as:

$$L_0(U,V) \approx L(A,B) := \frac{1}{2}\left\|\widetilde{Y} - \frac{1}{\sqrt{m}}AB^{\mathsf{T}}\right\|_F^2 + \phi_u(R_u^{1/2}A) + \phi_v(R_v^{1/2}B), \tag{23}$$

and then find the minima:

$$\widehat{A},\widehat{B} = \operatorname*{argmin}_{A,B} L(A,B). \tag{24}$$

We call $L(A,B)$ the *quadratic approximate loss function*.

Several possible AMP methods are available for the minimization (24). In this work, we consider a generalization of the rank one method in [22] shown in Algorithm 1, which we call Biased Low-Rank AMP. Here, the function $G_a(\cdot)$ is the denoiser

$$G_a(p_i, r_i^u, F^a) := \operatorname*{argmin}_a \frac{1}{2}\|a - p_i\|_{F^a}^2 + g_u(\sqrt{r_i^u}a), \tag{25}$$

where we use the notation $\|x\|_F^2 = x^{\mathsf{T}}Fx$. The denoiser $G_b(\cdot)$ is defined similarly. The updates for the Γ_k^a is

$$\Gamma_k^a = \frac{1}{n}\sum_{j=1}^n \frac{\partial G_b([P_k^b]_{j*}, r_j^v, F_k^b)}{\partial [P_k^b]_{j*}^{\mathsf{T}}} \tag{26a}$$

$$\Gamma_k^b = \frac{1}{m}\sum_{i=1}^m \frac{\partial G_a([P_k^a]_{i*}, r_i^u, F_k^a)}{\partial [P_k^a]_{i*}^{\mathsf{T}}}. \tag{26b}$$

Algorithm 1 Biased Low Rank AMP

Require: Number of iterations K_{it}; denoisers $G_a(\cdot)$, $G_b(\cdot)$; initial matrix $\widehat{B}_0 \in \mathbb{R}^{n\times d}$.

Initialize $k=0$, $\Gamma_k^a = 0$
while $k < K_{it}$ **do**
  $F_k^a = \frac{1}{m}\widehat{B}_k^\intercal \widehat{B}_k - \Gamma_k^a$
  $P_k^a = \frac{1}{\sqrt{m}}\widehat{Y}\widehat{B}_k - \widehat{A}_{k-1}\Gamma_k^a$
  $[\widehat{A}_k]_{i*} = G_a([P_k^a]_{i*}, r_i^u, F_k^a) \quad \forall i \in [m]$
  $\Gamma_k^b = \frac{1}{m}\sum_{i=1}^m \partial G_a([P_k^a]_{i*}, r_i^u, F_k^a)/\partial [P_k^a]_{i*}^\intercal$
  $F_k^b = \frac{1}{m}\widehat{A}_k^\intercal \widehat{A}_k - \Gamma_k^b$
  $P_k^b = \frac{1}{\sqrt{m}}\widehat{Y}^\intercal \widehat{A}_k - \widehat{B}_k\Gamma_k^b$
  $[\widehat{B}_{k+1}]_{j*} = G_b([P_k^b]_{j*}, r_j^v, F_k^b) \quad \forall j \in [n]$
  $\Gamma_{k+1}^a = \frac{1}{n}\sum_{j=1}^n \partial G_b([P_k^b]_{j*}, r_j^v, F_k^b)/\partial [P_k^b]_{j*}^\intercal$
  $k \leftarrow k+1$
end while
return $\widehat{A}_k$ and $\widehat{B}_{k+1}$

Algorithm 2 State Evolution

Require: Number of iterations K_{it}; denoisers $G_a(\cdot)$, $G_b(\cdot)$; initial random row vector $\widehat{B}_0 \in \mathbb{R}^d$.

Initialize $k=0$, $\Gamma_k^a = 0$
while $k < K_{it}$ **do**
  $M_k^b = \mathbb{E}\,(B^\intercal \widehat{B}_k)$, $Q_k^b = \mathbb{E}\,(\widehat{B}_k^\intercal \widehat{B}_k)$
  $F_k^a = Q_k^b - \Gamma_k^a$
  $P_k^a = AM_k^b + \mathcal{N}(0, Q_k^b)$
  $\widehat{A}_k = G_a(P_k^a, R^u, F_k^a)$
  $\Gamma_k^b = \mathbb{E}\,[\partial G_a(P_k^a, R^u, F_k^a)/\partial P_k^a]$
  $M_k^a = \mathbb{E}\,(A^\intercal \widehat{A}_k)$, $Q_k^a = \mathbb{E}\,(\widehat{A}_k^\intercal \widehat{A}_k)$
  $F_k^b = Q_k^a - \Gamma_k^b$
  $P_k^b = BM_k^a + \mathcal{N}(0, Q_k^a)$
  $\widehat{B}_{k+1} = G_b(P_k^b, R^v, F_k^b)$
  $\Gamma_{k+1}^b = \mathbb{E}\,\left[\partial G_a(P_k^b, R^v, F_k^b)/\partial P_k^b\right]$
  $k \leftarrow k+1$
end while
return $\widehat{A}_k$ and $\widehat{B}_{k+1}$

For the squared norm reguarlizer (16), it can be verified that the denoisers are given by:

$$G_a([P_k^a]_{i*}, r_i^u, F_k^a) = [P_k^a]_{i*}(F_k^a + \lambda_u r_i^u I_d)^{-1} \tag{27a}$$

$$G_b([P_k^b]_{j*}, r_j^v, F_k^b) = [P_k^b]_{j*}(F_k^b + \lambda_v r_j^v I_d)^{-1} \tag{27b}$$

Algorithm 1 is similar to the Low Rank AMP algorithm of [10] where we have replaced the MMSE denoiser with the MAP denoiser and added a row dependence for the bias terms. The following Lemma shows that if the algorithm converges, it is, at least, a local minima of the objective.

Lemma 1. *Any fixed point of Algorithm 1 is a local minimum of* (23)*.*

IV. Analysis in the Large System Limit

The benefit of the AMP method is that the performance of the algorithm can be precisely analyzed in a certain *large system limit* (LSL) as is commonly used in studying AMP algorithms. In the LSL, we consider a sequence of problems indexed by n. For each n, we assume that $m = m(n)$ where

$$\lim_{n\to\infty} \frac{m(n)}{n} = \beta, \tag{28}$$

for some $\beta > 0$. That is, the number of values of the random variables X_1 and X_2 grow linearly. The bias terms r_i^u and r_j^v as well as the true embedding vectors u_i and v_j are treated as deterministic vectors that converge empirically with second-order to random variables

$$\{r_i^u\} \xrightarrow{d} R^u, \quad \{r_j^v\} \xrightarrow{d} R^v, \quad \{u_i\} \xrightarrow{d} U, \quad \{v_j\} \xrightarrow{d} V, \tag{29}$$

where R^u and R^v are scalar random variables and U and V are random d-dimensional vectors. We let

$$A = U/\sqrt{R^u}, \quad B = V/\sqrt{R^v} \tag{30}$$

denote the random vectors for the normalized rows. In addition, we assume that the rows of the initial condition converge empirically as:

$$\{[\widehat{B}_0]_{i*}\} \xrightarrow{d} B_0, \tag{31}$$

where B_0 is some random row vector. To simplify the analysis, we assume that the random variables and vectors in (29) are bounded and $G_a(\cdot)$ and $G_b(\cdot)$ are Lipschitz continuous.

Under these assumptions, the joint distribution of true embedding vectors and their estimates can be predicted by a state evolution (SE). The SE, shown in Algorithm 2 is a modification of the result in [23]. The SE generates a sequence of deterministic quantities such as M_k^a, Q_k^a, F_k^a, as well as random row vectors such as P_k^a, and $\widehat{A}_k$. It is argued, in the full paper that the joint distributions of the embedding vectors and their estimates converge as

$$([A]_{i*}, [\widehat{A}_k]_{i*}, r_i^u) \xrightarrow{d} (A, \widehat{A}_k, R^u) \tag{32a}$$

$$([B]_{j*}, [\widehat{B}_k]_{j*}, r_j^v) \xrightarrow{d} (B, \widehat{B}_k, R^v) \tag{32b}$$

From these distributions, we can then compute any row-wise metrics on the error of the estimated embedding vectors – see [22] for examples.

V. Numerical Experiments

A. Synthetic data

To validate the SE equations, we first consider a simple synthetic data example. We use $m = 2000, n = 3000, d = 10$ and use L_2 regularizers (16) with $\lambda_u = \lambda_v = 10^{-4}$. We generate rows of true matrices U_0 and V_0 following:

$$\mathbf{u}_i \sim \mathcal{N}(0, 0.1I) \quad i \in [m]$$
$$\mathbf{v}_j \sim \mathcal{N}(0, 0.1I) \quad j \in [n]$$

To generate the problem instance we assume that s_i^u's and s_j^v's randomly take one of the values from the set $\{5, 6\}$.We will use estimations of these biases via (10) in our algorithms. We run algorithms 1 and 2 for 20 instances and average our results.

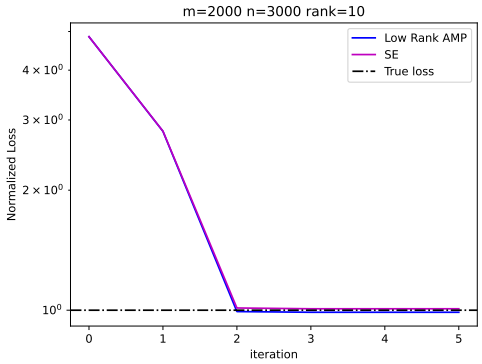

Fig. 1. Normalized loss vs iteration averaged over 20 instances, evaluated for an instance of the problem with $m = 2000$, $n = 3000$, and $d = 10$.

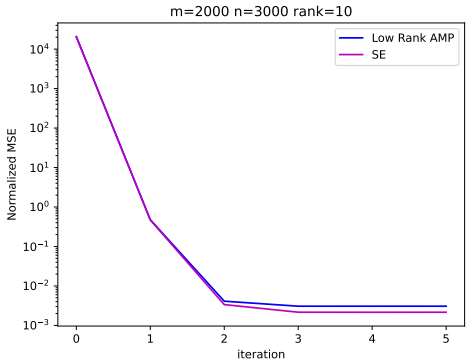

Fig. 2. Normalized MSE vs iteration averaged over 20 instances, evaluated for an instance of the problem with $m = 2000$, $n = 3000$, and $d = 10$. We note that $MSE = 1$ refers to setting $\widehat{A} = \mathbf{0}, \widehat{B} = \mathbf{0}$.

We initialize the $\widehat{A}_k$ and $\widehat{B}_k$ matrices with i.i.d. entries with zero mean and unit variance Gaussian distributions. Fig. 1 shows the loss function (23) (normalized by the true loss) vs iterations, averaged over 20 instances of the problem. We see that the average of the loss function observed in the simulations closely matches the predicted training loss from the SE.

We can also use the SE to estimate the error on the correlation terms: For each iteration index k, let M_{ij} and $\widehat{M}_{ij}^k$ denote the true and estimated correlation values:

$$M_{ij} = [A]_{i*}[B]_{j*}^{\intercal}, \quad \widehat{M}_{ij}^k = [\widehat{A}_k]_{i*}[\widehat{B}_k]_{j*}^{\intercal} \tag{33}$$

At each iteration k, defined the normalized MSE as:

$$MSE_k := \frac{\mathbb{E}(M_{ij} - \widehat{M}_{ij}^k)^2}{\mathbb{E}(M_{ij})^2}, \tag{34}$$

where the expectation is over the indices i and j. This MSE corresponds to how well the true correlation of the events $X_1 = i$ and $X_2 = j$ are predicted. We can similarly obtain a prediction of the MSE from the SE. Figure 2. shows the simulated MSE and SE predictions as a function of the iteration. Again, we see an excellent match.

B. MSE vs. inverse Fisher information

A basic challenge in many text problems is that there is a high variabilty of the terms. In our model, this property is equivalent to variability in the marginal probabilities $P(X_1 = i)$ and $P(X_2 = j)$ over indices i and j. Presumably, the estimation of the correlation $M_{ij} = \boldsymbol{u}_i^{\intercal}\boldsymbol{v}_j$ will be better when the $P(X_1 = i)$ and $P(X_2 = j)$ are higher so that there are more samples with $(x_1, x_2) = (i, j)$. This intuition is predicted by our model. Specifically, the state evolution reveals that the key parameter in estimation accuracy of M_{ij} is the inverse Fisher information, Δ_{ij} in (19). To validate this prediction, Fig. 3 shows a scatter plot of samples of the normalized MSE of M_{ij} vs. Δ_{ij} demonstrating higher inverse Fisher information results in higher MSE. Moreover, the joint distribution of the MSE and Fisher information is well-predicted by the state evolution.

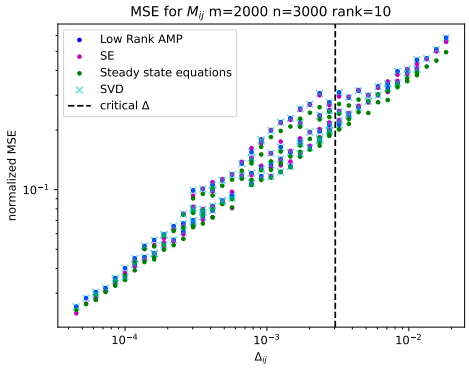

Fig. 3. Effect of individual biases on each element of M. As expected, we see an increasing trend of MSE with respect to Δ.

C. Evaluating the algorithm on a real text dataset

Finally, we apply our proposed algorithm over text data from a publicly available dataset called Large Movie Review Dataset [24]. More specifically, we select a large batch of text documents and perform preprocessing steps explained in the full paper [21]. Next, we construct a document-word co-occurance matrix using the data. This will serve as our Z matrix. We estimate bias vectors s^u and s^v using (10) and use the estimations to compute $\widetilde{Y}$. At this stage, we do not have the ground truth distributions. Thus, we apply algorithm 1 and use the output $\widehat{A}$ and $\widehat{B}$, and corresponding $\widehat{U}$ and $\widehat{V}$ as the ground truths U and V, respectively. Now, we sample m and n rows from ground truth U and V's, respectively and observe the new matrix Z from the Poisson channel. We apply algorithms 1 and 2 to derive the final estimations $\widehat{A}_k$ and $\widehat{B}_k$ and corresponding $\widehat{U}_k$ and $\widehat{V}_k$. Fig. 4 and Fig. 5 show the resulting loss and MSE when we sample $m = 2000$ and $n = 3000$ from the ground truth distributions. Again, we see an excellent match between the SE and simulations.

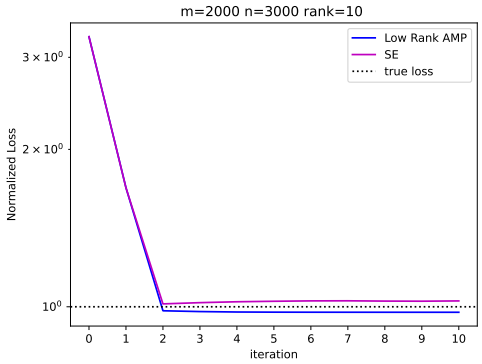

Fig. 4. Loss function vs iteration when sampling from a real dataset.

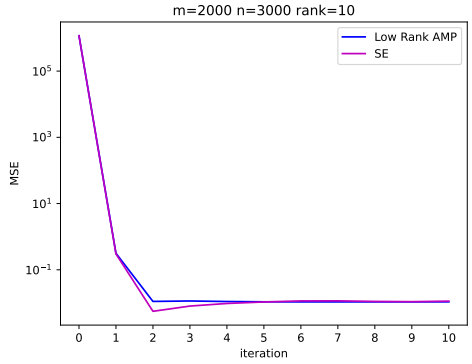

Fig. 5. MSE vs iteration when sampling from a real dataset.

VI. Conclusions

We have proposed a simple Poisson model to study learning of embeddings. Applying an AMP algorithm to this estimation problem enables predictions of how key parameters such as the embedding dimension, number of samples and relative frequency impact embedding estimation. Future work could consider more complex models, where the embedding correlation are described by a neural network. Also, we have assumed that the embedding dimension is known. An interesting avenue is to study the behavior of the methods in both over and under-parameterized regimes.

References

[1] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," *Artificial Intelligence Review*, vol. 56, no. 9, p. 10345–10425, Feb. 2023.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[4] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of hierarchical text classification using word embeddings," *Information Sciences*, vol. 471, pp. 216–232, jan 2019.

[5] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431.

[6] N. K. Kumar and J. Schneider, "Literature survey on low rank approximation of matrices," *Linear and Multilinear Algebra*, vol. 65, no. 11, pp. 2212–2244, 2017.

[7] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS'00. Cambridge, MA, USA: MIT Press, 2000, p. 535–541.

[8] R. Matsushita and T. Tanaka, "Low-rank matrix reconstruction and clustering via approximate message passing," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[9] S. Rangan and A. K. Fletcher, "Iterative estimation of constrained rank-one matrices in noise," *CoRR*, vol. abs/1202.2759, 2012.

[10] T. Lesieur, F. Krzakala, and L. Zdeborová, "Mmse of probabilistic low-rank matrix estimation: Universality with respect to the output channel," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 680–687.

[11] T. Lesieur, F. Krzakala, and L. Zdeborová , "Constrained low-rank matrix estimation: phase transitions, approximate message passing and applications," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2017, no. 7, p. 073403, jul 2017.

[12] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, nov 2009.

[13] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 764–785, 2011.

[14] O. Y. Feng, R. Venkataramanan, C. Rush, and R. J. Samworth, "A unifying tutorial on approximate message passing," 2021.

[15] A. K. Fletcher, S. Rangan, and P. Schniter, "Inference in deep networks in high dimensions," in *Proc. IEEE International Symposium on Information Theory*. IEEE, 2018, pp. 1884–1888.

[16] Y. Deshpande and A. Montanari, "Information-theoretically optimal sparse pca," 2014.

[17] Y. Deshpande, E. Abbe, and A. Montanari, "Asymptotic mutual information for the two-groups stochastic block model," 2015.

[18] A. Montanari and E. Richard, "Non-negative principal component analysis: Message passing algorithms and sharp asymptotics," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1458–1484, 2016.

[19] Y. Kabashima, F. Krzakala, M. Mezard, A. Sakata, and L. Zdeborova, "Phase transitions and sample complexity in bayes-optimal matrix factorization," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 4228–4265, jul 2016.

[20] A. Montanari and R. Venkataramanan, "Estimation of low-rank matrices via approximate message passing," 2019.

[21] G. A. Azar, M. Emami, A. K. Fletcher, and S. Rangan, "Estimation of embedding vectors in high dimensions," 2023.

[22] A. K. Fletcher and S. Rangan, "Iterative reconstruction of rank-one matrices in noise," *Information and Inference: A Journal of the IMA*, vol. 7, no. 3, pp. 531–562, 2018.

[23] A. K. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Expectation consistent approximate inference: Generalizations and convergence," 2017.

[24] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150.