Tensor Decomposition with Missing Indices

Yuto Yamaguchi[†], **Kohei Hayashi**[†]
†AIST, Japan
yuto.ymgc@gmail.com, hayashi.kohei@gmail.com

Abstract

How can we decompose a data tensor if the indices are partially missing? Tensor decomposition is a fundamental tool to analyze the tensor data. Suppose, for example, we have a 3rd-order tensor $\mathcal{X}$ where each element $\mathcal{X}_{ijk}$ takes 1 if user i posts word j at location k on Twitter. Standard tensor decomposition expects all the indices are observed. However, in some tweets, location k can be missing. In this paper, we study a tensor decomposition problem where the indices (i, j, or k) of some observed elements are partially missing. Towards the problem, we propose a probabilistic tensor decomposition model that handles missing indices as latent variables. To infer them, we develop an algorithm based on the variational MAP-EM algorithm, which enables us to leverage the information from the incomplete data. The experiments on both synthetic and real datasets show that the proposed model achieves higher accuracy in the tensor completion task than baselines.

1 Introduction

A tensor data is a set of values indexed by multiple indices. Some kinds of real world data are naturally represented as tensors, such as the number of times a `user` posts a `word` at a specific `location` in social media. Tensor decomposition, such as CP decomposition [Carroll and Chang, 1970; Harshman, 1970], is a fundamental tool to analyze the tensor data, which has various kinds of applications including recommendation [Rendle and Schmidt-Thieme, 2010], ranking semantic Web data [Franz *et al.*, 2009], and many others [Kolda and Bader, 2009].

One of the major applications of tensor decomposition is the estimation of *missing values*. Suppose we have a movie rating dataset, which contains a list of quadruplet (`user`, `movie`, `context`, `score`) that indicates a `user` watched a `movie` at a `context` and evaluated its quality as `score`. As a tensor, (`user`, `movie`, `context`) represents its index and `score` represents its value. In general, there are many missing values, i.e., the number of observed samples (quadruplets) is much smaller than the number of all the possible combinations of `users`, `movies`, and `contexts`. By estimating the parameters from the observed quadruplets, tensor decomposition can predict the missing values.

In real applications, not only values but also a part of **indices can be missing**. Let us take a Twitter data as an example, which consists of a set of tweets and each tweet contains a lot of attributes such as a user ID, a message text, a timestamp, and a location. This data can be represented as a higher-order tensor and analyzing it by tensor decomposition looks promising. However, in Twitter, some attributes are optional and they are not always observed, i.e., some indices are missing. Indeed, only 0.42% of tweets contain the location attribute [Cheng *et al.*, 2010]. We call a sample (a pair of indices and a value) whose indices are partially missing an *incomplete sample*, and call a sample whose indices are completely observed a *complete sample*.

Existing tensor decomposition methods are not directly applicable to the tensors with missing indices. A straightforward way to handle the tensor data containing both complete and incomplete samples is just ignoring the incomplete samples. That is, if we observe that user i posts word j at some unknown location, we simply discard this information. However, when the majority of samples are incomplete, this approach is problematic—it may cause severe degradation of solution due to the decrease of the sample size. Another way of handling missing indices is to discard the index through all samples. That is, we discard the location attributes of all tweets if there is at least one tweet that does not have its location attribute. This approach is again not desirable—we cannot use the location attributes at all even though some tweets have those information.

Present work. In this paper, we study the tensor decomposition problem where indices are partially missing. For this problem, we propose a probabilistic generative model for tensor decomposition that handles missing indices. Our main idea is to deal with the missing indices as *latent variables*. By inferring the latent variables, we can leverage incomplete samples as well as complete samples, to learn tensor decomposition. One of the advantages of the proposed model is its *flexibility*—it can be applied to the higher-order (>3) tensors, and to any tensor decomposition models (e.g., CP decomposition). For parameter inference, we develop an algorithm based on the *variational MAP-EM algorithm* [Gupta *et*

al., 2011]. We perform experiments on both synthetic and real datasets. The experimental results show that utilizing both complete and incomplete samples leads to improving the accuracy of tensor decomposition, comparing the proposed model and baselines that do not appropriately deal with incomplete samples.

Contributions. The contributions of this paper are summarized as follows:

1. **New Problem**: we define a tensor decomposition problem where the indices are partially missing.
2. **Model**: we propose a probabilistic generative model for tensor decomposition, which handles the missing indices as latent variables. The proposed model is flexible, i.e., it can be applied to higher-order (>3) tensors, and to any tensor decomposition models.
3. **Algorithm**: we develop a parameter inference algorithm for the proposed model based on the variational MAP-EM algorithm.

2 Related Work

In the history of the tensor decomposition research [Kolda and Bader, 2009], many decomposition methods have been proposed, such as CP decomposition [Carroll and Chang, 1970; Harshman, 1970], Tucker decomposition [Tucker, 1966], pairwise-interaction decomposition [Rendle and Schmidt-Thieme, 2010], and RESCAL [Nickel *et al.*, 2011], to name a few. Since tensors are often very large, one of the main streams of the tensor decomposition research is about efficient computation, including sampling-based methods [Papalexakis *et al.*, 2015], online/streaming computations [Sun *et al.*, 2006; Maehara *et al.*, 2016], memory efficient algorithms [Kolda and Sun, 2008; Oh *et al.*, 2017], and parallel computations [Jeon *et al.*, 2016].

Tensor data in real world tends to have *missing values*, which has triggered a lot of studies that tackle with them. [Acar *et al.*, 2011a] developed an extension of CP decomposition that can handle missing values, which is solved as a weighted least square problem. [Jain and Oh, 2014] theoretically studied how many samples are needed to exactly reconstruct a data tensor. Similar to our study, some papers [Chu and Ghahramani, 2009; Rai *et al.*, 2014; Zhao *et al.*, 2016] proposed probabilistic models to address missing values. However, the problem these studies address is completely different from our problem, because their focus is on the missing values, whereas our focus is on the missing *indices*.

Although, to the best of our knowledge, there is no study to address the missing indices, the most similar one to ours would be coupled matrix and tensor factorizations (CMTF) [Acar *et al.*, 2011b]. CMTF decomposes a set of tensors and matrices that share some of their modes. Although CMTF is designed for utilizing side information matrices in addition to the target data tensor, it is practically applicable to our problem where some indices are missing. That is, it could utilize incomplete samples by representing them as matrices, and decompose them together with the tensor that represents the set of complete samples. However, in this way, CMTF simply ignores the missing indices. In contrast, our proposed model infers the missing indices and utilize them for improving tensor decomposition. We compare these methods in the experiments.

3 Setup

In this section, we introduce the notation and define the problem. To make the notation and the presentation easy to follow, we only discuss the case where we are given a 3rd-order tensor. However, we emphasize that our model can be naturally generalized to higher-order tensors.

3.1 Conventional Tensor Decomposition

For $n \in \mathbb{N}$, we write by $[n]$ the set $\{1, 2, \dots, n\}$. Let $\mathcal{X}$ be an $I \times J \times K$ tensor that can contain missing values, where $I, J, K \in \mathbb{N}$, and let $\mathcal{D} \subseteq [I] \times [J] \times [K]$ be a set of indices of observed values. We denote by $\emptyset$ the missing values, i.e., $\mathcal{X}_{ijk} = \emptyset$ if $(i, j, k) \notin \mathcal{D}$, or $\mathcal{X}_{ijk} \in \mathbb{R}$ otherwise. Using the squared loss, tensor decomposition of $\mathcal{X}$ is defined as follows:

Definition 1 (Squared-loss tensor decomposition)**.** *Let $\mathcal{P}$ be a space of parameters and $\hat{\mathcal{X}}_{ijk} : \mathcal{P} \to \mathbb{R}$ be a predictive model. Given $\mathcal{X} \in \{\mathbb{R} \cup \{\emptyset\}\}^{I \times J \times K}$, tensor decomposition with $\hat{\mathcal{X}}$ is to compute $\hat{\Theta} = \arg\min_{\Theta \in \mathcal{P}} L(\mathcal{X}; \Theta)$ where*

$$L(\mathcal{X}; \Theta) = \frac{1}{2} \sum_{(i,j,k) \in \mathcal{D}} \left(\mathcal{X}_{ijk} - \hat{\mathcal{X}}(\Theta)_{ijk} \right)^2 .$$

There are several choices for the predictive model $\hat{\mathcal{X}}$. Here we define CP decomposition model that is used as the example in this paper:

Definition 2 (Rank-R CP decomposition)**.** *Given $R \in \mathbb{N}$, let $\Theta = \{U, V, W\}$ where $U \in \mathbb{R}^{I \times R}$, $V \in \mathbb{R}^{J \times R}$, and $W \in \mathbb{R}^{K \times R}$. The predictive model is defined as*

$$\hat{\mathcal{X}}(\Theta)_{ijk} = \sum_{r=1}^{R} U_{ir} V_{jr} W_{kr}.$$

Hereafter we write $\hat{\mathcal{X}}_{ijk}$ instead of $\hat{\mathcal{X}}(\Theta)_{ijk}$ if it is not confusing.

3.2 Tensor Decomposition with Missing Indices

Next we consider the case where a part of indices are missing. In such a case it is difficult to represent data as a tensor. Instead, we consider the data as a set of indices and values. Suppose we have $N \in \mathbb{N}$ observations (i.e., samples) and denote by $X = \{x_n\}_{n=1}^N$ the values and by $\hat{Z} = \{(\hat{i}_n, \hat{j}_n, \hat{k}_n)\}_{n=1}^N$ the corresponding indices that can be missing. As well as missing values, we denote the missing index by $\emptyset$.

The tensor decomposition problem with missing indices is then defined as follows:

Problem 1 (Tensor decomposition with missing indices)**.** *Suppose we have a set of values $X \in \mathbb{R}^N$, a set of indices $\hat{Z} \in \{[I] \cup \{\emptyset\}\} \times \{[J] \cup \{\emptyset\}\} \times \{[K] \cup \{\emptyset\}\}^N$, a tensor decomposition model $(\mathcal{P}, \hat{\mathcal{X}})$, and an oracle $\mathcal{O}$ that returns the true indices for all missing indices. We want to find*

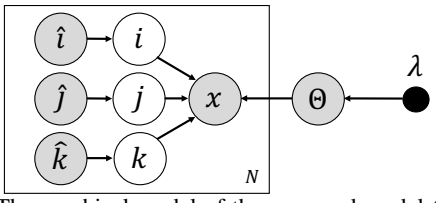


Figure 1: The graphical model of the proposed model (3rd-order). Shaded circles denote the observed variables, white circles denote the latent variables, and small black circle denotes the parameter.

$\hat{\Theta} = \arg\min_{\Theta \in \mathcal{P}} L(X, Z; \Theta)$*, where* $Z = \mathcal{O}(\hat{Z})$ *is a set of true indices fulfilled by the oracle and*

$$L(X, Z; \Theta) = \frac{1}{2} \sum_n \left(x_n - \hat{\mathcal{X}}_{i_n j_n k_n} \right)^2 .$$

4 Proposed Model

Since the oracle in Problem 1 is in reality not in hand, we solve the relaxed problem by estimating set of true indices Z to obtain the minimizer $\hat{\Theta}$. To handle missing indices, we consider them as latent variables. Let $i_n \in [I]$ be the latent variable denoting the true index which is not given, and $\hat{i}_n \in \{[I] \cup \emptyset\}$ be the given index, which can be observed or missing. We hypothesize the latent variable i_n is generated from the following distribution:

$$p_I(i_n|\hat{i}_n) = \begin{cases} \delta(i_n, \hat{i}) & (\hat{i_n} = \hat{i} \in [I]) \\ Unif(1, I) & (\hat{i}_n = \emptyset), \end{cases}$$

where $\delta(a, b)$ is the delta function taking 1 if $a = b$, or 0 otherwise, and $Unif(1, I)$ is the categorical uniform distribution from 1 to I. p_J and p_K are also defined in the same way. The above definition means that, in prior, we have no information about true index i_n if given index $\hat{i}_n$ is missing, whereas true index i_n is always the same as the given index if it is observed. In our model, we infer those latent variables (i.e., missing indices) and learn tensor decomposition simultaneously, which improves the accuracy of tensor decomposition as will be shown in the experiments.

The generative process of our proposed model is written as follows:

- Generate $\Theta \sim p_\Theta(\cdot|\lambda)$
- For n from 1 to N
  - Generate $i_n \sim p_I(\cdot|\hat{i}_n)$
  - Generate $j_n \sim p_J(\cdot|\hat{j}_n)$
  - Generate $k_n \sim p_K(\cdot|\hat{k}_n)$
  - Generate $x_n \sim N(\cdot|\hat{\mathcal{X}}_{i_n j_n k_n}, 1)$

where $N(\cdot|\mu, \sigma^2)$ is Gaussian distribution with mean μ and variance σ^2, and $p_\Theta(\cdot|\lambda)$ is the prior distribution of Θ with some parameter λ depending on the tensor decomposition model. Fig. 1 shows the graphical model of the proposed model. Note that the proposed model allows the duplicated indices, that is , there exist samples n_1 and n_2 such that $\hat{i}_{n_1} = \hat{i}_{n_2}$, $\hat{j}_{n_1} = \hat{j}_{n_2}$, and $\hat{k}_{n_1} = \hat{k}_{n_2}$.

The following proposition states that, if we could estimate the true indices correctly for all n, the MLE of Θ in this model is identical to the solution of Problem 1:

Proposition 3. *If the estimted indices* $\bar{Z}$ *are the same as the true indices* Z*, the following holds:*

$$\arg\min_{\Theta \in \mathcal{P}} L(X, Z; \Theta) = \arg\max_{\Theta \in \mathcal{P}} p(X|\bar{Z}, \Theta). \quad (1)$$

Proof. We can write the log-likelihood as follows:

$$\ln p(X|\bar{Z}, \Theta) = -\frac{1}{2} \sum_n \left(x_n - \hat{\mathcal{X}}_{i_n j_n k_n} \right)^2 + C,$$

where C is some constant, which concludes the proof. □

With regard to the above proposition, we can say that we can solve Problem 1 at least approximately by estimating the missing indices and Θ simultaneously.

To prevent from the overfitting, instead of MLE, we aim to obtain the marginal MAP as follows:

$$\hat{\Theta}_{MAP} = \arg\max_{\Theta} p(\Theta|X, \hat{Z}, \lambda).$$

Since $p(\Theta|X, \hat{Z}, \lambda) \propto p(X, \hat{Z}, \Theta|\lambda)$, the marginal MAP can be obtained maximizing the joint distribution of the model written as follows:

$$\ln p(X, \hat{Z}, \Theta|\lambda) = \ln \sum_Z p(X, Z, \hat{Z}|\Theta) + \ln p(\Theta|\lambda),$$

where

$$\ln \sum_Z p(X, Z, \hat{Z}|\Theta) =$$

$$\sum_n \ln \sum_{i_n j_n k_n} N(x_n|\hat{\mathcal{X}}_{i_n j_n k_n}, 1) p_I(i_n|\hat{i}_n) p_J(j_n|\hat{j}_n) p_K(k_n|\hat{k}_n).$$

4.1 Parameter Inference

In this section, we develop a parameter inference algorithm based on the variational MAP-EM algorithm. The joint distribution defined in the previous section can be written in the variational form as follows:

$$\ln p(X, \hat{Z}, \Theta|\lambda) = L[q, \Theta] + KL(q||p),$$

where

$$L[q, \Theta] = E_{q(Z)} \left[\ln \frac{p(X, Z, \hat{Z}, \Theta|\lambda)}{q(Z)} \right],$$

$$KL(q||p) = E_{q(Z)} \left[\ln \frac{q(Z)}{p(Z|X, \hat{Z}, \Theta, \lambda)} \right],$$

and q is any distribution over Z. Since $KL(q||p) > 0$, the joint distribution is always greater than $L[q, \Theta]$ that is called *variational lower bound* (VLB). In the variational MAP-EM algorithm, instead of directly maximizing the joint distribution, the VLB is maximized to obtain the marginal MAP by alternatively performing E-step and M-step as described in the following paragraphs.

In the E-step, $L[q, \Theta]$ is maximized with regard to $q(Z)$ with Θ fixed. Assuming the *mean field approximation*

$q(Z) = \prod_n q(i_n)q(j_n)q(k_n)$, and differentiating $L[q, \Theta]$ by $q(i_n = i)$, we get the following updating equation:

$$q(i_n = \hat{i}) = \frac{p_I(i_n = \hat{i}|\hat{i}_n)e^{a_{n,\hat{i}}}}{\sum_i p_I(i_n = i|\hat{i}_n)e^{a_{n,i}}}, \quad (2)$$

where $a_{n,i} = -\frac{1}{2}E_{q(j_n,k_n)}\left[(x_n - \hat{\mathcal{X}}_{ij_nk_n})^2\right]$. We can also update $q(j_n)$ and $q(k_n)$ in the same way. Since $q(i_n)$, $q(j_n)$, and $q(k_n)$ depend on each other, we repeatedly calculate them until they converge. Note that if $\hat{i}_n = \hat{i}$ (i.e., $\hat{i}_n$ is not missing), $q(i_n)$ is instantly obtained as $\delta(i_n, \hat{i})$ because $p(i_n|\hat{i}_n) = \delta(i_n, \hat{i})$, which means that there is no need to calculate q for observed indices.

In the M-step, $L[q, \Theta]$ is maximized with regard to Θ with $q(Z)$ obtained in the previous E-step fixed. Differentiating $L[q, \Theta]$ by $\theta \in \Theta$, we get:

$$\frac{\partial L[q,\Theta]}{\partial \theta} = \sum_n \sum_{z_n} q(z_n)\left(x_n - \hat{\mathcal{X}}_{z_n}\right)\frac{\partial \hat{\mathcal{X}}_{z_n}}{\partial \theta} + \frac{\partial \ln p(\Theta|\lambda)}{\partial \theta},$$

where we denote $z_n = (i_n, j_n, k_n)$.

The concrete computations of the E-step and the M-step depend on the tensor decomposition model, in other words, depend on how to define $\hat{\mathcal{X}}_{ijk}$ and $p(\Theta|\lambda)$. In the next subsection, we describe the parameter inference algorithm for CP decomposition as an example. However, we emphasize again that any tensor decomposition models can be plugged-in to our proposed model.

Parameter inference for CP decomposition

Recall that the parameters of CP decomposition are $\Theta = \{U, V, W\}$. Hereafter we write $E[\cdot]$ for $E_q[\cdot]$ if it is not confusing. In the E-step, we can calculate $a_{n,i}$ as follows:

$$a_{n,i} = x_n U_i^T \left(E[V_{j_n}] \odot E[W_{k_n}]\right)$$
$$-\frac{1}{2}U_i^T\left(E\left[V_{j_n}V_{j_n}^T\right] \odot E\left[W_{k_n}W_{k_n}^T\right]\right)U_i, \quad (3)$$

where $\odot$ denotes the Hadamard product, and we omit the constant terms that are not needed to compute $q(i_n)$.

In the M-step, $p_\Theta(\Theta|\lambda)$ needs to be defined. For CP decomposition, we define $p_\Theta(\Theta|\lambda)$ as follows:

$$p_\Theta(\Theta|\lambda) =$$
$$\prod_i N(U_i|\mathbf{0}, \lambda^{-1}\boldsymbol{I})\prod_j N(V_j|\mathbf{0}, \lambda^{-1}\boldsymbol{I})\prod_k N(W_k|\mathbf{0}, \lambda^{-1}\boldsymbol{I}),$$

where $\boldsymbol{I}$ denotes the identity matrix with the appropriate number of dimensions. Keeping in mind the above definition of $p_\Theta(\Theta|\lambda)$, we differentiate $L[q, \Theta]$ by U_i, and get the following updating equation:

$$U_i = \left(G_i + \lambda \boldsymbol{I}\right)^{-1} H_i, \quad (4)$$

where

$$G_i = \sum_{n:i_n=i} q(i_n = i)\left(E\left[V_{j_n}V_{j_n}^{\mathrm{T}}\right] \odot E\left[W_{k_n}W_{k_n}^{\mathrm{T}}\right]\right),$$

and

$$H_i = \sum_{n:i_n=i} q(i_n = i)x_n\left(E[V_{j_n}] \odot E[W_{k_n}]\right).$$

V and W can also be updated in the same way. Since U, V, and W depend on each other, we alternatively update them until they converge. The sufficient statistics of the algorithm are $E[U_{i_n}]$, $E[V_{j_n}]$, $E[W_{k_n}]$, $E\left[U_{i_n}U_{i_n}^T\right]$, $E\left[V_{j_n}V_{j_n}^T\right]$, and $E\left[W_{k_n}W_{k_n}^T\right]$ for all n.

To summarize, the parameter inference algorithm for the proposed model with CP decomposition is shown in Algorithms 1-3. Every time q and Θ are updated, the related sufficient statistics have to be also updated.

Complexity. We show the time complexity of the proposed algorithm for higher-order tensors. Suppose we have Mth-order tensor where the number of dimensions of mth mode is I_m. The time complexity of the proposed algorithm with CP decomposition model for each iteration is $\mathrm{O}(\mathrm{R}^3\sum_{\mathrm{m}}\mathrm{I_m} + \mathrm{R}^2(\mathrm{N} + \sum_{\mathrm{m}}\mathrm{N_m^-I_m}))$, where N_m^- is the number of missing indices for mth mode. This complexity becomes the same as the conventional CP decomposition when there is *no* incomplete samples (i.e., $N_m^- = 0$ for all m).

5 Experiments

In this section, we report the results of the experiments conducted on both synthetic and real datasets. We compare the three variants of the proposed model, and three baselines as follows:

- `MAP-EM (Proposed)`: Algorithm 1 with q inferred by the data.
- `Uniform (Proposed)`: Algorithm 1 with q fixed as the categorical uniform distribution.
- `Prior (Proposed)`: Algorithm 1 with q fixed as the distribution that is estimated from the data as a histogram. It is obtained as $q(i_n = i) = N_i^+/N_I^+$ for all n where N_i^+ is the number of samples where $\hat{i}_n = i$, and N_I^+ is the number of samples where $\hat{i}_n \neq \emptyset$.
- `Minimal`: CP decomposition using only complete samples.
- `Complete`: CP decomposition using only modes without missing indices. For example, when the 3rd index of the 3rd-order tensors can be missing, this method performs matrix decomposition using only 1st and 2nd modes.
- `CMTF`: coupled matrix and tensor factorization proposed in [Acar *et al.*, 2011b]. This method can utilize a set of incomplete samples by representing it as a matrix.

The first three are our proposed models with q set in different way.

Reproducibility. Our code to reproduce the experiments is available at `https://goo.gl/W4K86I`. The real dataset used in the experiments in Section 5.2 is publicly available.

5.1 Synthetic Dataset

Settings. To see on what condition the parameters of the proposed model is correctly recovered, we generate two data tensors with a large/small number of samples. The size of the

Algorithm 1 Variational MAP-EM algorithm for the proposed model with CP decomposition (3rd-order)

Input: R, λ.
Output: U, V, W.
Initialize q, U, V, and W.
Calculate all sufficient statistics.
while convergence criterion is met **do**
  # E-step
  for $n = 1$ to N **do**
    while convergence criterion is met **do**
      if $\hat{i}_n = \emptyset$ **then**
        Update-$q(i_n)$ (Alg. 2)
        Update $E\left[U_{i_n}\right]$ and $E\left[U_{i_n}U_{i_n}^T\right]$
      end if
      if $\hat{j}_n = \emptyset$ **then**
        Update-$q(j_n)$ (Alg. 2)
        Update $E\left[V_{j_n}\right]$ and $E\left[V_{j_n}V_{j_n}^T\right]$
      end if
      if $\hat{k}_n = \emptyset$ **then**
        Update-$q(k_n)$ (Alg. 2)
        Update $E\left[W_{k_n}\right]$ and $E\left[W_{k_n}W_{k_n}^T\right]$
      end if
    end while
  end for
  # M-step
  while convergence criterion is met **do**
    Update-U (Alg. 3)
    for each n where $\hat{i}_n = \emptyset$ **do**
      Update $E\left[U_{i_n}\right]$ and $E\left[U_{i_n}U_{i_n}^T\right]$
    end for
    Update-V (Alg. 3)
    for each n where $\hat{j}_n = \emptyset$ **do**
      Update $E\left[V_{j_n}\right]$ and $E\left[V_{j_n}V_{j_n}^T\right]$
    end for
    Update-W (Alg. 3)
    for each n where $\hat{k}_n = \emptyset$ **do**
      Update $E\left[W_{k_n}\right]$ and $E\left[W_{k_n}W_{k_n}^T\right]$
    end for
  end while
end while

Algorithm 2 Update-$q(i_n)$

Output: $q(i_n)$
$z \leftarrow 0$
for $i = 1$ to I **do**
  Calculate $a_{n,i}$ (Eqn. 3).
  $z \leftarrow z + \exp(a_{n,i})$.
end for
for $i = 1$ to I **do**
  $q(i_n = i) \leftarrow \exp(a_{n,i})/z$.
end for

Algorithm 3 Update-U

Output: U
for $i = 1$ to I **do**
  $G_i \leftarrow \boldsymbol{O}$ # zero matrix of size $R \times R$
  $H_i \leftarrow \mathbf{0}$ # zero vector of size R
end for
for $n = 1$ to N **do**
  $\overline{\boldsymbol{z}} \leftarrow E\left[V_{j_n}V_{j_n}^T\right] \odot E\left[W_{k_n}W_{k_n}^T\right]$
  $\overline{\boldsymbol{z}\boldsymbol{z}} \leftarrow E\left[V_{j_n}V_{j_n}^T\right] \odot E\left[W_{k_n}W_{k_n}^T\right]$
  if $\hat{i}_n = \emptyset$ **then**
    for $i = 1$ to I **do**
      $H_i \leftarrow H_i + q(i_n = i)x_n\overline{\boldsymbol{z}}$
      $G_i \leftarrow G_i + q(i_n = i)\overline{\boldsymbol{z}\boldsymbol{z}}$
    end for
  else
    $H_{\hat{i}_n} \leftarrow H_{\hat{i}_n} + x_n\overline{\boldsymbol{z}}$
    $G_{\hat{i}_n} \leftarrow G_{\hat{i}_n} + \overline{\boldsymbol{z}\boldsymbol{z}}$
  end if
end for
for $i = 1$ to I **do**
  $U_i \leftarrow \left(G_i + \lambda\boldsymbol{I}\right)^{-1} H_i$
end for

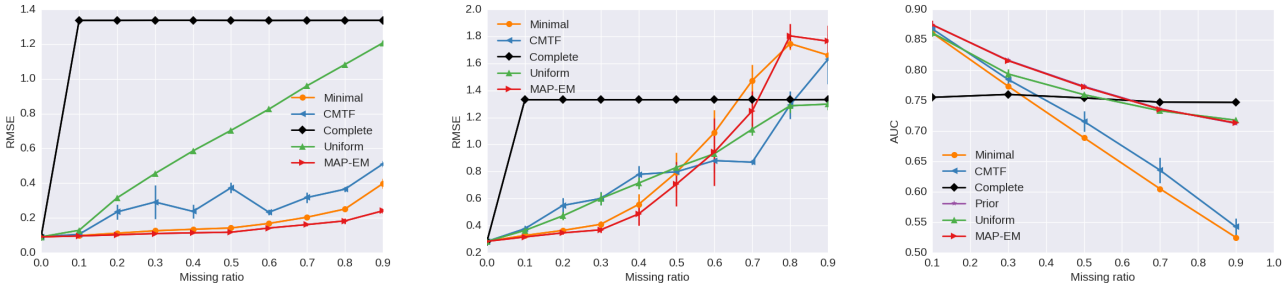
tensors is $10 \times 10 \times 10$. First, we generate each element of $U^{\text{true}} \in \mathbb{R}^{10\times R}$, $V^{\text{true}} \in \mathbb{R}^{10\times R}$, and $W^{\text{true}} \in \mathbb{R}^{10\times R}$ from Gaussian distribution $N(\cdot|0, \lambda^{-1})$ with $\lambda = 1.0$ and $R = 3$. Then, we randomly divide the set of all indices $D = [10] \times [10] \times [10]$ into training set D_{train} (90%) and test set D_{test} (10%). For each index in D_{train}, we generate S samples from Gaussian distribution $N(\cdot|\mathcal{X}_{ijk}, 1)$, where $\mathcal{X}_{ijk} = \sum_r U_{ir}^{\text{true}}V_{jr}^{\text{true}}W_{kr}^{\text{true}}$. $S = 1$ for small dataset, whereas $S = 10$ for large dataset. For each index in D_{test}, we use $\mathcal{X}_{ijk}$ itself without Gaussian noise.

We compare the cases where the different ratio of indices are missing. After generating the data tensors, we delete the 3rd index by probability α. We vary α from 0 to 0.9 by the step size 0.1. For each α, we perform 5 trials and report the mean and the standard deviation of the results. Note that since `Complete` discards the 3rd mode, it predicts the test samples as matrix decomposition, i.e., $\hat{\mathcal{X}}_{ijk} = \sum_r U_{ir}V_{jr}$ for all k. We use the root mean square error (RMSE) defined as follows: $RMSE = \sqrt{(1/|D_{\text{test}}|)\sum_{(i,j,k)\in D_{\text{test}}}(\mathcal{X}_{ijk} - \hat{\mathcal{X}}_{ijk})^2}$.

Results. Figs. 2(a) and 2(b) show the results. We do not compare `Prior` because the prior is the categorical uniform distribution in this experiment, which is the same as `Uniform`. On the large dataset (Fig. 2(a)), `MAP-EM` achieves the smallest RMSE among all compared methods, which means that the parameters are correctly recovered. On the other hand, on the small dataset (Fig. 2(b)), `Uniform` is better than `MAP-EM` when $\alpha \geq 0.6$, which indicates that the distribution q is not correctly inferred when the number of samples is not enough and the missing ratio is too large. `CMTF` and `Minimal` are almost always worse than `MAP-EM` and `Uniform`, meaning the proposed model works reasonably well for our problem where some indices are missing. `Complete` does not work at all when the missing ratio is not so large, because it discards all the information about the 3rd mode when at least one sample is incomplete. However, it shows comparable (or even better) results to other methods when the number of samples is too few and the missing ratio is too large, because, in such a case, the provided data is almost a *matrix* (i.e., most of 3rd indices are missing).

5.2 Real Dataset

Settings. We use the Twitter dataset that is publicly avail-



(a) Synthetic (large; $S = 10$) (b) Synthetic (small; $S = 1$) (c) Twitter data

Figure 2: The experimental results on (a) synthetic data (large), (b) synthetic data (small), and (c) Twitter data. The x-axes denote the ratio of missing indices, and the y-axes denote (a,b) the RMSE or (c) the AUC. Error bars denote the standard deviation. `MAP-EM` achieves the best results except when there is too few samples and the ratio of missing indices is too large.

able.[1] This dataset consists of about 13M geotagged tweets. We construct a *user-hashtag-location* (3rd-order) tensor as follows. First, we discard all the tweets posted from outside the US. The number of remained tweets posted from inside the US is about 2.85M. Then we extract *user ids*, *hashtags*, and *locations* from those tweets. We use 49 states that appear in the dataset as locations. We filter user ids that occur less than 5 times, and hashtags that occur less than 30 times. As a result, we construct a $25297 \times 2616 \times 49$ tensor $\mathcal{X}$ where $\mathcal{X}_{ijk} = 1$ if user i posts hashtag j at location k, or $\mathcal{X}_{ijk} = 0$ otherwise. The number of non-zero elements is $125,017$. Since this data is very sparse, we randamly sample the same number of zero elements as non-zero elements, that is, the number of observed (zero and non-zero) samples is $N = 250,034$. We randomly divide the samples into training set (70%) and test set (30%).

As the same as the previous experiment, we delete the 3rd index by probability α that vary from 0.1 to 0.9 by the step size 0.2. To delete the indices, we randomly sample user ids by probability α, and then delete the 3rd index (i.e., location) of the samples. For each α, we perform 5 trials and report the mean and the standard deviation of the results. Parameters are set to $\lambda = 1.0$ and $R = 10$.

We use the area under ROC curve (AUC) [Fawcett, 2006] as the evaluation metric. The AUC is often used to evaluate binary classification. In this experiment, our objective is to predict the value of the target data tensor, which takes 0 or 1. Hence this task can be evaluated as binary classification. To calculate the AUC, we first sort the test samples in descending order in terms of its predicted values. Then we calculate the number of true positives and false positives at each position in the ranking, and calculate the AUC using those numbers.

Results. Fig. 2(c) shows the results. First of all, our three proposed methods (i.e., `MAP-EM`, `Prior`, and `Uniform`) outperform `CMTF` and `Minimal`, which suggests that the proposed model works well even on the real dataset. However, when there are too many missing indices ($\alpha \geq 0.7$), `Complete` shows the highest AUCs, which agrees with the previous experiment on the small synthetic data.

Comparing the three proposed methods, `MAP-EM` and `Prior` show almost the same AUCs, which are larger than that of `Uniform` especially when $\alpha \leq 0.5$. We observed that the q inferred by `MAP-EM` is very close to the prior q in this experiment. This observation suggests that, even though `MAP-EM` can infer the global trend, it cannot infer the correct q for each sample n when the missing ratio is large.

Although we do not report the actual computational time, it almost follows the theoretical time complexity.

6 Conclusion

In this paper, we define and address a tensor decomposition problem where the indices can be missing. We propose a probabilistic generative model that handles the missing indices as a latent variables, and develop the parameter inference algorithm based on the variational MAP-EM algorithm. The experimental results show that our proposed model outperforms the baselines except when there are only too few samples and the ratio of missing indices is too large. There are options how to deal with the latent variables. Performing the full variational MAP-EM works the best when the enough samples are available, whereas when we are given only scarce samples, simply using the prior q achieves reasonable results and reduces the computational cost.

There are several open problems as follows. First, it is important to theoretically guarantee how many samples are needed to recover the missing indices correctly. Second, since any tensor decomposition models can be plugged-in to our proposed model, it is interesting to develop the concrete parameter inference algorithm for those models. Finally, we can also consider different models that are not assuming Gaussian. For example, for 0-1 or count data such as Twitter data, assuming Poisson or Bernoulli distribution for $p(x_n|i_n, j_n, k_n, \Theta)$ would be promissing.

Acknowledgements

This research was partially supported by the program "Research and Development on Real World Big Data Integration and Analysis" of RIKEN, Japan, and by a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

[1] `http://noisy-text.github.io/2016/geo-shared-task.html`

References

[Acar *et al.*, 2011a] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[Acar *et al.*, 2011b] Evrim Acar, Tamara G Kolda, and Daniel M Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422*, 2011.

[Carroll and Chang, 1970] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[Cheng *et al.*, 2010] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *CIKM*, pages 759–768, 2010.

[Chu and Ghahramani, 2009] Wei Chu and Zoubin Ghahramani. Probabilistic models for incomplete multi-dimensional arrays. In *AISTATS*, pages 89–96, 2009.

[Fawcett, 2006] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[Franz *et al.*, 2009] Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *International semantic web conference*, pages 213–228. Springer, 2009.

[Gupta *et al.*, 2011] Maya R Gupta, Yihua Chen, et al. Theory and use of the em algorithm. *Foundations and Trends® in Signal Processing*, 4(3):223–296, 2011.

[Harshman, 1970] Richard A Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis. 1970.

[Jain and Oh, 2014] Prateek Jain and Sewoong Oh. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*, pages 1431–1439, 2014.

[Jeon *et al.*, 2016] Inah Jeon, Evangelos E Papalexakis, Christos Faloutsos, Lee Sael, and U Kang. Mining billion-scale tensors: algorithms and discoveries. *The VLDB Journal*, 25(4):519–544, 2016.

[Kolda and Bader, 2009] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[Kolda and Sun, 2008] Tamara G Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 363–372. IEEE, 2008.

[Maehara *et al.*, 2016] Takanori Maehara, Kohei Hayashi, and Ken-ichi Kawarabayashi. Expected tensor decomposition with stochastic gradient descent. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1919–1925. AAAI Press, 2016.

[Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816, 2011.

[Oh *et al.*, 2017] Jinoh Oh, Kijung Shin, Evangelos E Papalexakis, Christos Faloutsos, and Hwanjo Yu. S-hot: Scalable high-order tucker decomposition. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 761–770. ACM, 2017.

[Papalexakis *et al.*, 2015] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. P ar c ube: Sparse parallelizable candecomp-parafac tensor decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):3, 2015.

[Rai *et al.*, 2014] Piyush Rai, Yingjian Wang, Shengbo Guo, Gary Chen, David B Dunson, and Lawrence Carin. Scalable bayesian low-rank decomposition of incomplete multiway tensors. In *ICML*, pages 1800–1808, 2014.

[Rendle and Schmidt-Thieme, 2010] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90. ACM, 2010.

[Sun *et al.*, 2006] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.

[Tucker, 1966] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[Zhao *et al.*, 2016] Qibin Zhao, Guoxu Zhou, Liqing Zhang, Andrzej Cichocki, and Shun-Ichi Amari. Bayesian robust tensor factorization for incomplete multiway data. *IEEE transactions on neural networks and learning systems*, 27(4):736–748, 2016.