

Efficient Nonparametric Tensor Decomposition for Binary and Count Data

Zerui Tao^{1,2}, Toshihisa Tanaka^{1,2}, Qibin Zhao^{2,1*}

¹Tokyo University of Agriculture and Technology, Japan

²RIKEN Center for Advanced Intelligence Project (AIP), Japan
zerui.tao@riken.jp, tanakat@cc.tuat.ac.jp, qibin.zhao@riken.jp

Abstract

In numerous applications, binary reactions or event counts are observed and stored within high-order tensors. Tensor decompositions (TDs) serve as a powerful tool to handle such high-dimensional and sparse data. However, many traditional TDs are explicitly or implicitly designed based on the Gaussian distribution, which is unsuitable for discrete data. Moreover, most TDs rely on predefined multi-linear structures, such as CP and Tucker formats. Therefore, they may not be effective enough to handle complex real-world datasets. To address these issues, we propose ENTED, an Efficient Nonparametric TENSOR Decomposition for binary and count tensors. Specifically, we first employ a nonparametric Gaussian process (GP) to replace traditional multi-linear structures. Next, we utilize the Pólya-Gamma augmentation which provides a unified framework to establish conjugate models for binary and count distributions. Finally, to address the computational issue of GPs, we enhance the model by incorporating sparse orthogonal variational inference of inducing points, which offers a more effective covariance approximation within GPs and stochastic natural gradient updates for nonparametric models. We evaluate our model on several real-world tensor completion tasks, considering binary and count datasets. The results manifest both better performance and computational advantages of the proposed model.

1 Introduction

Tensor data are ubiquitous in many real-world applications, such as visual processing (Liu et al. 2012; Zhao, Zhang, and Cichocki 2015), spatial-temporal forecasting (Bahadori, Yu, and Liu 2014; Qiu et al. 2021), probabilistic modeling (Glasser et al. 2019; Novikov, Panov, and Oseledets 2021), among many others. Tensor decomposition (TD) is a powerful tool for handling such high-order data. Due to the large tensor sizes, TD aims to factorize the original data into much smaller tensor factors. Through these sharing tensor factors, underlying structures or correlations among different tensor modes can be captured. Based on this idea, many elegant TD models have been proposed, such as CP decomposition (Hitchcock 1927), Tucker decomposition (Tucker 1966), tensor train/ring decomposition (Oseledets 2011; Zhao et al. 2019) and many variants (Kolda and Bader 2009; Cichocki

et al. 2016). Choosing proper TD structures often involves domain knowledge and can largely affect the final performance (Li and Sun 2020; Li et al. 2022).

While most TD models focus on continuous problems, many real-world applications may encounter binary or count data. For example, in click-through-rate (CTR) prediction tasks, a tensor of shape $user \times item \times time$ may store records of whether a specific user clicked on the item at the time. Additionally, many tensors consist of multi-way events, where each element is the count of the event history (Schein et al. 2015, 2016). For example, the Covid-19 dataset (Dong, Du, and Gardner 2020) contains the number of infection claims, and each observation is associated with several attributes (tensor modes) such as locations, time, and claim types. However, less effort is made to deal with such discrete observations. Compared to continuous counterparts, handling discrete distributions brings additional difficulties when constructing probabilistic models, due to their non-differentiable and non-conjugacy nature.

To address these issues, we propose ENTED, an Efficient Nonparametric Tensor Decomposition for binary and count data. The proposed model is inherited from the Gaussian process tensor factorization (GPTF, Zhe et al. 2016). Specifically, we adopt a Gaussian process (GP) to replace traditional multi-linear contraction rules. The great flexibility of nonparametric GPs enables us to learn underlying structures of complex real-world datasets adaptively, rather than picking one beforehand. To cope with discrete data, the Pólya-Gamma (PG) augmentation (Polson, Scott, and Windle 2013) is adopted, which provides a unified framework to establish conjugate models for both binary and count data (Klami 2015). Moreover, to efficiently approximate infeasible covariance matrices in GPs, we derive a novel sparse orthogonal variational inference (SOLVE, Shi, Titsias, and Mnih 2020) scheme for our model that incorporates PG augmentation and natural gradient (NG) updates. Notably, our model allows for stochastic optimization that is scalable to large tensors. The contributions are summarized as follows:

- We propose a flexible nonparametric TD for binary and count data. By using GPs, the model can adaptively learn complex hidden structures of high-order tensors.
- A PG augmentation scheme is adopted, resulting in a unified augmented model for binary and count data. Due to the conjugacy, efficient NG updates can be derived.

*Corresponding author

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

- To obtain an efficient covariance approximation, we derive a SOLVE framework with PG augmentation and NG updates for nonparametric tensor factorization, which enables fast and stochastic optimization.

Finally, we demonstrate the proposed model on binary and count tensor completion tasks. Our model shows superior prediction accuracy and distributional estimation on the six real-world datasets. In addition, ablation studies on inducing points are conducted to show the effectiveness and computational benefits of our model.

2 Backgrounds

2.1 Notations

We denote scalars, vectors, matrices, and tensors as lowercase letters, bold lowercase letters, bold capital letters, and sans-serif bold capital letters, *e.g.*, $x, \mathbf{x}, \mathbf{X}$ and \mathbf{X} , respectively. For an order- D tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \dots \times I_D}$, we denote its (i_1, \dots, i_D) -th entry as $x_{\mathbf{i}}$, where $\mathbf{i} = (i_1, \dots, i_D)$. Moreover, $\mathcal{N}(\cdot, \cdot)$ denotes Normal distribution, $\mathcal{B}(\cdot)$ denotes Bernoulli distribution, $NB(\cdot, \cdot)$ denotes negative binomial (NB) distribution, $PG(\cdot)$ denotes Pólya-Gamma (PG) distribution and $D_{\text{KL}}(p||q)$ denotes the Kullback-Leibler (KL) divergence between two distributions p and q .

2.2 Tensor Decomposition

Tensor decomposition (TD, Kolda and Bader 2009) aims to factorize an order- D tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \dots \times I_D}$ into D smaller latent factors $\mathbf{Z}^{(d)} \in \mathbb{R}^{I_d \times R_d}, \forall d = 1, \dots, D$, where the sequence (R_1, \dots, R_D) is the tensor rank. Traditional TDs depend on predefined contraction rules. For example, the CP decomposition (Hitchcock 1927) assumes,

$$x_{\mathbf{i}} = \sum_{r=1}^R \lambda_r z_{i_1 r}^{(1)} \dots z_{i_D r}^{(D)}, \quad (1)$$

where λ_r are factor weights for each rank-1 components and $R = R_1 = \dots = R_D$. Tucker decomposition (Tucker 1966) extends CP to have multiway weights, *i.e.*, $x_{\mathbf{i}} = \sum_{r_1=1}^{R_1} \dots \sum_{r_D=1}^{R_D} \lambda_{r_1 \dots r_D} z_{i_1 r_1}^{(1)} \dots z_{i_D r_D}^{(D)}$. Other popular TDs include tensor train (TT, Oseledets 2011), tensor ring (TR, Zhao et al. 2019), t-SVD (Kilmer et al. 2013) and many variants (Kolda and Bader 2009; Cichocki et al. 2016). All these TD methods employ predefined multi-linear structures, possibly insufficient to cope with complex real-world datasets.

To mitigate the issue, a line of work (Chu and Ghahramani 2009; Xu, Yan, and Qi 2012; Zhe et al. 2016) studied nonparametric TDs, which can adaptively learn nonlinear structures from data. Here, we briefly introduce the Gaussian process tensor factorization (GPTF, Zhe et al. 2016) due to its flexibility and scalability. In particular, given latent factors $\mathbf{Z}^{(d)} \in \mathbb{R}^{I_d \times R}, \forall d = 1, \dots, D$ with $R = R_1 = \dots = R_D$, we denote latent factors associated with index \mathbf{i} as $\mathbf{m}_{\mathbf{i}} = [z_{i_1}^{(1)}, \dots, z_{i_D}^{(D)}] \in \mathbb{R}^{DR}$, where $z_{i_d}^{(d)} \in \mathbb{R}^R$ is the i_d -th row of $\mathbf{Z}^{(d)}$. In GPTF, the linear contraction forms of traditional TDs (*e.g.*, Eq. (1)) is characterized by a GP,

$$p(x_{\mathbf{i}}) = \mathcal{N}(x_{\mathbf{i}} | f_{\mathbf{i}}, \beta^{-1}), p(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, k(\mathbf{M}_{\Omega}, \mathbf{M}_{\Omega})),$$

where $k(\cdot, \cdot)$ is a kernel function, $\Omega = \{\mathbf{i}_1, \dots, \mathbf{i}_N\}$ denotes all observed indices and $\mathbf{M}_{\Omega} \in \mathbb{R}^{N \times DR}$ is concatenated by latent factors associating with all observed entries, *i.e.*, $\{\mathbf{m}_{\mathbf{i}} : \mathbf{i} \in \Omega\}$. To deal with binary data, Zhe et al. (2016) adopted an augmented variable $\omega_n, \forall n = 1, \dots, N$ and the Probit model,

$$p(x_{i_n} | \omega_n) = \mathcal{B}(\Phi(\omega_n)), \quad p(\omega_n | f_n) = \mathcal{N}(\omega_n | f_n, 1),$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of standard Normal distribution. Finally, the joint distribution for binary GPTF becomes,

$$p(\mathbf{x}_{\Omega}, \mathbf{f}, \boldsymbol{\omega}, \mathbf{Z}) = p(\mathbf{f} | \mathbf{Z}) \cdot \prod_{d=1}^D p(\mathbf{Z}^{(d)}) \cdot \prod_{n=1}^N p(x_{i_n} | \omega_n) p(\omega_n | f_n), \quad (2)$$

where we denote $\mathbf{Z} = \{\mathbf{Z}^{(d)}\}_{d=1}^D$ for simplicity.

However, learning this model requires cubic complexity with sample sizes, *i.e.*, $\mathcal{O}(N^3)$, which is prohibited in real applications with massive observations. To address the issue, Zhe et al. (2016) adopted the sparse variational GP (SVGP, Titsias 2009) framework and derived a distributed evidence lower bound (ELBO) analogous to Gal, Van Der Wilk, and Rasmussen (2014). In specific, a small set of inducing inputs $\mathbf{B} \in \mathbb{R}^{p \times DR}$ and points $\mathbf{u} \in \mathbb{R}^p$ are introduced. By assuming the observations are conditionally independent given the inducing points, the probabilistic model in Eq. (2) can be expressed as

$$p(\mathbf{x}_{\Omega}, \mathbf{f}, \boldsymbol{\omega}, \mathbf{Z}, \mathbf{u}) = \prod_{d=1}^D p(\mathbf{Z}^{(d)}) \cdot \prod_{n=1}^N p(x_{i_n} | \omega_n) p(\omega_n | f_n) \cdot p(\mathbf{f} | \mathbf{u}, \mathbf{Z}) p(\mathbf{u}), \quad (3)$$

where

$$p(\mathbf{f} | \mathbf{u}, \mathbf{Z}) = \mathcal{N}(\mathbf{f} | \mathbf{K}_{MB} \mathbf{K}_{BB}^{-1} \mathbf{u}, \tilde{\mathbf{K}}), \quad (4)$$

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \mathbf{0}, \mathbf{K}_{BB}). \quad (5)$$

For simplicity, we denote $\mathbf{K}_{MB} = k(\mathbf{M}_{\Omega}, \mathbf{B}), \mathbf{K}_{BB} = k(\mathbf{B}, \mathbf{B}), \mathbf{K}_{MM} = k(\mathbf{M}_{\Omega}, \mathbf{M}_{\Omega})$ and $\tilde{\mathbf{K}} = \mathbf{K}_{MM} - \mathbf{K}_{MB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{BM}$. To learn posteriors of the inducing points \mathbf{u} and the augmented variable $\boldsymbol{\omega}$, a variational distribution $q(\mathbf{u}, \boldsymbol{\omega}) = q(\mathbf{u})q(\boldsymbol{\omega})$ is adopted. The ELBO becomes,

$$\log p(\mathbf{x}_{\Omega}, \mathbf{Z}) \geq \mathbb{E}_{p(\mathbf{f}|\mathbf{u})q(\mathbf{u})q(\boldsymbol{\omega})} \log p(\mathbf{x}_{\Omega} | \mathbf{Z}, \mathbf{f}, \mathbf{u}, \boldsymbol{\omega}) - D_{\text{KL}}(q(\mathbf{u})||p(\mathbf{u})) - D_{\text{KL}}(q(\boldsymbol{\omega})||p(\boldsymbol{\omega})) + \log p(\mathbf{Z}). \quad (6)$$

Zhe et al. (2016) showed that analytical solutions of $q(\mathbf{u}), q(\boldsymbol{\omega})$ can be obtained and the collapsed form of Eq. (6) can be computed in a *distributed* manner. The computational complexity is reduced to $\mathcal{O}(Np^2 + p^3)$. Nevertheless, this approach can not handle integer observations like count data. Also, the distributed objective cannot be optimized in a stochastic way, which is desirable in many applications, *e.g.*, when powerful computing clusters are not available or the samples come in streams. To enable stochastic optimization, we present an extension of GPTF in Appendix A.

3 Proposed Model

3.1 Nonparametric Tensor Decomposition with Pólya-Gamma Augmentation

While many tensor data encounter discrete observations, GPTF (Zhe et al. 2016) cannot deal with them directly, since discrete distributions yield non-conjugate GPs. To address this issue, we improve GPTF by employing the PG augmentation (Polson, Scott, and Windle 2013), that provides a unified framework to establish conjugate models for Bernoulli and NB distributions (Klami 2015).

Pólya-Gamma Augmentation A PG variable $\omega \sim PG(b, c)$ is defined as (Polson, Scott, and Windle 2013),

$$\omega \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k-1/2)^2 + c^2/(4\pi^2)},$$

where $g_k \stackrel{\text{iid}}{\sim} Ga(b, 1)$, and $\stackrel{D}{=}$ means equality in distribution. Given PG variable $\omega \sim PG(b, 0)$, we have,

$$\frac{\exp(t)^a}{(1 + \exp(t))^b} = 2^{-b} \exp((a-b/2)t) \int_0^{\infty} \exp(-\omega t^2/2) p(\omega) d\omega. \quad (7)$$

For binary data $x_i \in \{0, 1\}, \forall i \in \Omega$, we adopt the logistic transform,

$$p(x_{i_n} | f_n) = \mathcal{B}(x_{i_n} | \sigma(f_n)) = (1 + \exp(-f_n))^{-1},$$

where $\sigma(\cdot)$ is the logistic function. For count data $x_i \in \mathbb{N}, \forall i \in \Omega$, we adopt the negative binomial (NB) model,

$$p(x_{i_n} | f_n) = NB(x_{i_n} | \zeta, p_n) = \frac{\Gamma(\zeta + x_{i_n})}{x_{i_n}! \Gamma(\zeta)} p_n^{x_{i_n}} (1-p_n)^\zeta$$

where $p_n = 1/(1 + \exp(-f_n))$ and the number of successes ζ is a hyper-parameter. It was shown that the Bernoulli and NB distributions can be augmented by PG variables as follows (Klami 2015),

$$p(x_{i_n}, \omega_n | f_n) \propto 2^{-b} \exp(\chi_n f_n - \frac{1}{2} \omega_n f_n^2) PG(\omega_n | b, 0), \quad (8)$$

where ω_n is the PG augmented variable. For binary data, $b = 1$ and $\chi_n = x_{i_n} - 1/2$. For count data, $b = x_i + \zeta$ and $\chi_n = (x_{i_n} - \zeta)/2$. According to Eq. (7), the original distribution $p(\mathbf{x}_\Omega | \mathbf{f})$ can be recovered by marginalizing out the augmented variable ω_n in Eq. (8). Note that Eq. (8) admits a quadratic form, which is conjugate to Gaussian distribution. Therefore, PG augmentation is widely used to handle non-Gaussian likelihoods. Finally, by adopting the PG augmentation, the joint distribution of Eq. (3) becomes,

$$p(\mathbf{x}_\Omega, \mathbf{f}, \boldsymbol{\omega}, \mathbf{Z}, \mathbf{u}) = \prod_{d=1}^D p(\mathbf{Z}^{(d)}). \quad (9)$$

$$p(\mathbf{x}_\Omega | \boldsymbol{\omega}, \mathbf{f}) p(\boldsymbol{\omega}) p(\mathbf{f} | \mathbf{u}, \mathbf{Z}) p(\mathbf{u}).$$

The prior $p(\mathbf{Z}^{(d)})$ is chosen to be standard Gaussian. Other distributions can be found in Eqs. (4), (5) and (8).

Evidence Lower Bound To marginalize out latent variables $\mathbf{f}, \boldsymbol{\omega}, \mathbf{u}$ in the joint PDF Eq. (9) in a scalable way, we have to seek for variational approximation. In particular, we can assign $q(\mathbf{f}, \boldsymbol{\omega}, \mathbf{u}) = p(\mathbf{f} | \mathbf{u}) q(\boldsymbol{\omega}) q(\mathbf{u})$, where

$$q(\boldsymbol{\omega}) = \prod_{n=1}^N PG(b, c_n), \quad q(\mathbf{u}) = \mathcal{N}(\boldsymbol{\mu}^{(u)}, \boldsymbol{\Sigma}^{(u)}). \quad (10)$$

Then, the ELBO becomes,

$$\log p_\theta(\mathbf{x}_\Omega, \mathbf{Z}) \geq \mathbb{E}_{p(\mathbf{f}|\mathbf{u})q(\boldsymbol{\omega})q(\mathbf{u})} \log p(\mathbf{x}_\Omega | \boldsymbol{\omega}, \mathbf{f}) - D_{\text{KL}}(q(\mathbf{u})q(\boldsymbol{\omega}) || p(\mathbf{u})p(\boldsymbol{\omega})) + \log p(\mathbf{Z}). \quad (11)$$

The optimal solution for $q(\boldsymbol{\omega})$ can be derived analytically, as we will show later. The time complexity of Eq. (11) is $\mathcal{O}(Np^2 + p^3)$, where p is the number of inducing points. More importantly, unlike in Zhe et al. (2016), this objective Eq. (11) is factorized over samples. Hence, N can be replaced by mini-batches, making it scalable to large datasets.

Inference with Natural Gradients Maximizing the ELBO Eq. (11) yields efficient stochastic variational inference (SVI) with natural gradient (NG) updates. Firstly, for local parameter \mathbf{c} , we have the following optimal solution,

$$c_n = \sqrt{\tilde{k}_{n,n} + \boldsymbol{\kappa}_n^{(u)} \boldsymbol{\Sigma}^{(u)} \boldsymbol{\kappa}_n^{(u),\top} + \boldsymbol{\mu}^{(u),\top} \boldsymbol{\kappa}_n^{(u)} \boldsymbol{\kappa}_n^{(u),\top} \boldsymbol{\mu}^{(u)}},$$

where $\tilde{k}_{n,n}$ are diagonal elements of $\tilde{\mathbf{K}}$, $\boldsymbol{\kappa}_n^{(u)}$ is the n -th row of $\mathbf{K}_{MB} \mathbf{K}_{BB}^{-1}$ (we treat it as a column vector for notation consistency). Then, we can derive the NGs of natural parameters $\boldsymbol{\eta}_1^{(u)} = \boldsymbol{\Sigma}^{(u),-1} \boldsymbol{\mu}^{(u)}$ and $\boldsymbol{\eta}_2^{(u)} = -\frac{1}{2} \boldsymbol{\Sigma}^{(u),-1}$ as

$$\tilde{\nabla} \boldsymbol{\eta}_1^{(u)} = \frac{N}{s} \sum_{n \in \mathcal{S}} \chi_n \boldsymbol{\kappa}_n^{(u)} - \boldsymbol{\eta}_1^{(u)}$$

$$\tilde{\nabla} \boldsymbol{\eta}_2^{(u)} = -\frac{1}{2} \left(\mathbf{K}_{BB}^{-1} + \frac{N}{s} \sum_{n \in \mathcal{S}} \theta_n \boldsymbol{\kappa}_n^{(u)} \boldsymbol{\kappa}_n^{(u),\top} \right) - \boldsymbol{\eta}_2^{(u)},$$

where $\tilde{\nabla}$ means NG. \mathcal{S} denotes the set of mini-batch data with $s = |\mathcal{S}|$, and $\theta_n = \frac{\tanh(c_n/2)}{2c_n}$. Finally, the latent factors \mathbf{Z} and inducing inputs \mathbf{B} are then optimized by maximizing the ELBO Eq. (11) using gradient-based methods such as Adam (Kingma and Ba 2014). For detailed derivations, please check Appendix B.2.

3.2 Efficient Orthogonally Decoupled Approximation

Although the sparse variational GP tensor decomposition presented in Section 3.1 can handle large-scale binary or count tensors. One may need to use many inducing points to approximate the full covariance matrix, which leads to large computational costs. To allow efficient sparse approximation, we derive another lower bound using the sparse orthogonal variational inference (SOLVE, Shi, Titsias, and Mnih 2020) framework for our model.

The idea is to decompose the GP into two orthogonal processes in the reproduce Hilbert kernel space (RHKS), and then adopt two sets of inducing points \mathbf{u}, \mathbf{v} to approximate

these two processes respectively. Specifically, we decompose function \mathbf{f} in Eq. (4) into two orthogonal components,

$$p(\mathbf{f}_\perp) = \mathcal{N}(\mathbf{0}, \tilde{\mathbf{K}}), \quad \mathbf{f} = \mathbf{f}_\perp + \mathbf{K}_{MB} \mathbf{K}_{BB}^{-1} \mathbf{u},$$

where $\tilde{\mathbf{K}}$ is defined in Eq. (4). Then, apart from inducing points \mathbf{u} , we use another set of inducing points \mathbf{v} to approximate \mathbf{f}_\perp separately,

$$p(\mathbf{v}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{HH}),$$

where \mathbf{K}_{HH} is the covariance matrix of corresponding inducing inputs \mathbf{H} . The joint probability in Eq. (9) becomes,

$$p(\mathbf{x}_\Omega, \mathbf{f}_\perp, \boldsymbol{\omega}, \mathbf{Z}, \mathbf{u}, \mathbf{v}) = \prod_{d=1}^D p(\mathbf{Z}^{(d)}) \cdot p(\mathbf{x}_\Omega | \boldsymbol{\omega}, \mathbf{f}) p(\boldsymbol{\omega}) p(\mathbf{u}) p(\mathbf{f}_\perp | \mathbf{v}) p(\mathbf{v}). \quad (12)$$

Similar to Eqs. (4) and (5), we have

$$p(\mathbf{f}_\perp | \mathbf{v}) = \mathcal{N}(\mathbf{K}_{MH} \mathbf{K}_{HH}^{-1} \mathbf{v}, \tilde{\mathbf{K}} - \mathbf{K}_{MH} \mathbf{K}_{HH}^{-1} \mathbf{K}_{HM}),$$

$$p(\mathbf{v}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{HH}),$$

where $\mathbf{K}_{MH} = k(\mathbf{M}_\Omega, \mathbf{H})$ and $\mathbf{K}_{HH} = k(\mathbf{H}, \mathbf{H})$.

To get the variational lower bound, apart from variational distributions in Eq. (10), we assign an additional variational distribution on \mathbf{v} , namely,

$$q(\mathbf{v}) = \mathcal{N}(\boldsymbol{\mu}^{(v)}, \boldsymbol{\Sigma}^{(v)}).$$

Due to the conjugate nature of GPs, we can get the approximated posterior of \mathbf{f}_\perp ,

$$q(\mathbf{f}_\perp) = \int p(\mathbf{f}_\perp | \mathbf{v}) q(\mathbf{v}) d\mathbf{v} = \mathcal{N}(\boldsymbol{\mu}^{(f_\perp)}, \boldsymbol{\Sigma}^{(f_\perp)}),$$

where

$$\boldsymbol{\mu}^{(f_\perp)} = \mathbf{C}_{MH} \mathbf{C}_{HH}^{-1} \boldsymbol{\mu}^{(v)},$$

$$\boldsymbol{\Sigma}^{(f_\perp)} = \tilde{\mathbf{K}} + \mathbf{C}_{MH} \mathbf{C}_{HH}^{-1} (\boldsymbol{\Sigma}^{(v)} - \mathbf{C}_{HH}) \mathbf{C}_{HH}^{-1} \mathbf{C}_{HM},$$

$$\mathbf{C}_{MH} = \mathbf{K}_{MH} - \mathbf{K}_{MB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{BH},$$

$$\mathbf{C}_{HH} = \mathbf{K}_{HH} - \mathbf{K}_{HB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{BH},$$

with $\mathbf{K}_{BH} = k(\mathbf{B}, \mathbf{H})$. Then we can derive the ELBO as

$$\log p_\theta(\mathbf{x}_\Omega, \mathbf{Z}) \geq \mathbb{E}_{q(\boldsymbol{\omega})q(\mathbf{u})q(\mathbf{f}_\perp)} \log p(\mathbf{x}_\Omega | \boldsymbol{\omega}, \mathbf{f}) - D_{\text{KL}}(q(\mathbf{u})q(\mathbf{v})q(\boldsymbol{\omega}) || p(\mathbf{u})p(\mathbf{v})p(\boldsymbol{\omega})) + \log p(\mathbf{Z}). \quad (13)$$

Note that the biggest difference between Eq. (11) and Eq. (13) is that, in Eq. (11) we take expectation to $p(\mathbf{f} | \mathbf{u})$, which is a prior, while in Eq. (13), we take expectation to $q(\mathbf{f}_\perp)$, which is a learnable posterior and leads to more flexible learning processes. Similarly, we can derive the closed-form updates for the PG variable,

$$c_n = \sqrt{\mu_n^{(f),2} + \sigma_{n,n}^{(f_\perp)} + \boldsymbol{\kappa}_n^{(u),\top} \boldsymbol{\Sigma}^{(u)} \boldsymbol{\kappa}_n^{(u)}},$$

where $\mu_n^{(f)} = \mu_n^{(f_\perp)} + \boldsymbol{\kappa}_n^{(u),\top} \boldsymbol{\mu}^{(u)}$ and $\sigma_{n,n}^{(f_\perp)}$ is the n -th diagonal element of $\boldsymbol{\Sigma}^{(f_\perp)}$. The NG for $q(\mathbf{u})$ and $q(\mathbf{v})$ can also be derived.

$$\tilde{\nabla}_{\boldsymbol{\eta}_1^{(i)}} = \sum_{n=1}^N (\chi_n - \theta_n \boldsymbol{\kappa}_n^{(j),\top} \boldsymbol{\mu}_n^{(j)}) \boldsymbol{\kappa}_n^{(i)} - \boldsymbol{\eta}_1^{(i)},$$

$$\tilde{\nabla}_{\boldsymbol{\eta}_2^{(i)}} = -\boldsymbol{\eta}_2^{(i)} - \frac{1}{2} (\mathbf{K}_{II}^{-1} + \sum_{n=1}^N \theta_n \boldsymbol{\kappa}_n^{(i)} \boldsymbol{\kappa}_n^{(i),\top}),$$

where i can be replaced by u or v , $j = v$ if $i = u$ (vice versa), and $I = B$ if $i = u$ or H if $i = v$. Moreover, $\boldsymbol{\kappa}_n^{(v)}$ is defined as the n -th row of $\mathbf{C}_{MH} \mathbf{C}_{HH}^{-1}$. As in Section 3.1, the latent factors \mathbf{Z} and inducing inputs \mathbf{B}, \mathbf{H} is then optimized by maximizing the ELBO Eq. (13). Detailed derivations are presented in Appendix B.3.

Complexity analysis Despite a more structured representation of variational approximations, SOLVE has additional computational benefits due to the decoupled inducing points. Suppose we choose $2p$ inducing points, computing Eq. (11) requires $\mathcal{O}(4Np^2 + 8p^3)$ time complexity. In SOLVE, supposing we decouple $2p$ points into two sets of p points, the complexity of optimizing Eq. (13) reduces to $\mathcal{O}(2Np^2 + 2p^3)$. Both approaches allow mini-batch training, which makes our model scalable to large datasets.

4 Related Work

Traditional TDs usually rely on specific multi-linear contraction rules, such as CP (Hitchcock 1927), Tucker (Tucker 1966), tensor train/ring (TT/TR, Oseledets 2011; Zhao et al. 2019) and many variants (Kolda and Bader 2009; Cichocki et al. 2016). While these models mainly focus on continuous cases, binary and count data have also been considered. Chi and Kolda (2012) proposed non-negative CP built on Poisson distribution. Then, the Bayesian version of Poisson CP was established (Schein et al. 2015, 2016) using gamma-Poisson conjugacy. Concurrently, Rai et al. (2014, 2015) proposed Bayesian CP for binary and count data using PG augmentation (PGA). Tao, Tanaka, and Zhao (2023) extended it to the TR format. Recently, Wang and Li (2020) proposed a low-rank Bernoulli model based on CP decomposition. Lee and Wang (2021) established a TD generated by summing a series of signs. Generalized CP (GCP, Hong, Kolda, and Duersch 2020) summarized diverse types of distributions and loss functions learned via gradient-based methods. Soulat et al. (2021) proposed a Bayesian GCP using PGA to deal with NB distribution for spiking count data. All of these methods are based on multi-linear structures and may lack flexibility for complex datasets.

To enhance flexibility, many non-linear TDs have been proposed. In particular, Chu and Ghahramani (2009); Xu, Yan, and Qi (2012); Zhe et al. (2015, 2016) proposed GP tensor factorizations (GPTFs) that use GPs to replace multi-linear contractions. To deal with binary data, a Probit transform was adopted. However, their model is unable to deal with count data. Several following-up works adopt Gamma distribution and Hawkes process to predict the happening time of each event (Zhe and Du 2018; Pan, Wang, and Zhe 2020; Wang et al. 2022). Our model differs from these works in several ways. Firstly, we adopt PGA to get a unified framework for both binary and count tensors. Secondly, we derived efficient natural gradient updates. Finally, a more efficient covariance approximation scheme is established for our model. Recently, Ibrahim et al. (2023) proposed a TD for count data using neural networks, which is out of the scope of this paper as we use GPs. Moreover, it requires side information for each mode, which is not always available.

In the context of GPs, handling non-conjugate models

<i>Digg</i>	AUC \uparrow			NLL \downarrow		
	Rank 3	Rank 5	Rank 10	Rank 3	Rank 5	Rank 10
GCP	0.566 \pm 0.034	0.555 \pm 0.026	0.539 \pm 0.045	5.356 \pm 0.302	6.747 \pm 1.011	8.696 \pm 1.874
BCP	0.564 \pm 0.016	0.566 \pm 0.031	0.547 \pm 0.024	0.688 \pm 0.003	0.690 \pm 0.003	0.689 \pm 0.001
SBTR	0.698 \pm 0.028	0.694 \pm 0.039	0.728 \pm 0.024	0.647 \pm 0.013	0.646 \pm 0.012	0.632 \pm 0.009
GPTF	0.629 \pm 0.025	0.655 \pm 0.028	0.737 \pm 0.017	0.663 \pm 0.013	0.662 \pm 0.010	0.611 \pm 0.019
CoSTCo	0.594 \pm 0.033	0.597 \pm 0.047	0.620 \pm 0.076	0.693 \pm 0.015	0.693 \pm 0.014	0.679 \pm 0.123
ENTED	0.720 \pm 0.039	0.743 \pm 0.039	0.767 \pm 0.032	0.622 \pm 0.041	0.599 \pm 0.030	0.577 \pm 0.030
<i>Enron</i>						
GCP	0.848 \pm 0.010	0.852 \pm 0.022	0.847 \pm 0.028	3.501 \pm 0.611	3.541 \pm 0.948	3.608 \pm 1.070
BCP	0.664 \pm 0.055	0.651 \pm 0.060	0.652 \pm 0.059	0.599 \pm 0.008	0.590 \pm 0.010	0.590 \pm 0.008
SBTR	0.899 \pm 0.031	0.905 \pm 0.019	0.896 \pm 0.010	0.487 \pm 0.039	0.470 \pm 0.036	0.476 \pm 0.022
GPTF	0.921 \pm 0.015	0.928 \pm 0.019	0.950 \pm 0.013	0.398 \pm 0.053	0.366 \pm 0.043	0.349 \pm 0.038
CoSTCo	0.549 \pm 0.072	0.602 \pm 0.045	0.838 \pm 0.114	0.693 \pm 0.010	0.693 \pm 0.014	0.485 \pm 0.477
ENTED	0.926 \pm 0.012	0.938 \pm 0.004	0.950 \pm 0.013	0.381 \pm 0.047	0.359 \pm 0.025	0.309 \pm 0.063
<i>DBLP</i>						
GCP	0.926 \pm 0.003	0.941 \pm 0.002	0.950 \pm 0.002	0.836 \pm 0.004	0.804 \pm 0.004	0.775 \pm 0.006
SBTR	0.897 \pm 0.008	0.915 \pm 0.004	0.954 \pm 0.000	0.460 \pm 0.029	0.440 \pm 0.010	0.317 \pm 0.005
GPTF	0.942 \pm 0.003	0.945 \pm 0.003	0.954 \pm 0.002	0.384 \pm 0.015	0.363 \pm 0.013	0.339 \pm 0.010
CoSTCo	0.892 \pm 0.006	0.904 \pm 0.006	0.907 \pm 0.001	0.440 \pm 0.502	0.403 \pm 0.518	0.381 \pm 0.513
ENTED	0.950 \pm 0.003	0.959 \pm 0.003	0.962 \pm 0.002	0.286 \pm 0.009	0.263 \pm 0.010	0.250 \pm 0.007

Table 1: Binary tensor completion.

and establishing efficient approximations are important topics. To approximate large covariance matrices, sparse variational Gaussian process (SVGP, Titsias 2009) was proposed to variationally learn inducing points. SVGP can be scaled to large datasets using stochastic optimization (Hensman, Fusi, and Lawrence 2013) or distributed learning (Gal, Van Der Wilk, and Rasmussen 2014). Based on the idea of SVGP, Hensman, Matthews, and Ghahramani (2015) proposed scalable GPs for binary classification. Wenzel et al. (2019) proposed to use PG augmentation for GPs, which yields conjugate models and fast NG updates. Recently, Shi, Titsias, and Mnih (2020) proposed sparse orthogonal variational Gaussian process (SOLVE-GP), which was shown to be more efficient in learning sparse GPs. However, SOLVE-GP was not designed for binary and count data and the authors did not utilize NG updates. Therefore, our model is not only contributing to the TD community, but also non-conjugate GPs in more general applications.

5 Experiments

In this section, we present empirical evaluations of the proposed model. All experiments are conducted on a workstation with an Intel Xeon Silver 4316 CPU@2.30GHz, 512GB RAM and NVIDIA RTX A6000 GPUs. More details about experimental settings and results are shown in Appendix C. The code is mainly based on PyTorch (Paszke et al. 2019) and available at <https://github.com/taozerui/gptd>

5.1 Binary Tensor Completion

Datasets We test our model on three binary tensor datasets: (1) Digg (Xu, Yan, and Qi 2012), an order-3 tensor

of shape $581 \times 124 \times 48$, extracted from the *digg.com* social news website, describing interactions among *news* \times *keyword* \times *topic*. It has 0.024% non-zero entries. (2) Enron (Xu, Yan, and Qi 2012), an order-3 tensor of shape $203 \times 203 \times 200$, storing records of an email system (*sender* \times *receiver* \times *time*) with 0.01% non-zero entries. (3) DBLP (Zhe et al. 2016), an order-3 tensor of shape $10k \times 200 \times 10k$, extracted from the DBLP database, depicting relationships among *author* \times *conference* \times *keyword* with 0.001% non-zero entries. For Digg and Enron, we randomly sample an equal number of zero entries to obtain a balanced dataset. For DBLP, the same train/test split with Zhe et al. (2016) is adopted. For binary datasets, we evaluate the area under the ROC curve (AUC) and the negative log-likelihood (NLL) of estimated Bernoulli distributions. We report the mean and standard deviation of 5-fold cross-validation.

Baselines We compare with five models: (1) GCP (Hong, Kolda, and Duersch 2020), a generalized CP designed for diverse types of data distributions and loss functions using gradient-based optimization. (2) BCP (Wang and Li 2020), a binary CPD with ALS-based algorithms. (3) SBTR (Tao, Tanaka, and Zhao 2023), a scalable Bayesian tensor ring that uses PGA to handle binary data, which can be regarded as a TR version of Rai et al. (2014). (4) GPTF (Zhe et al. 2016), the GP tensor factorization that uses the Probit likelihood Eq. (2) for binary data. (5) CoSTCo (Liu et al. 2019), a non-linear TD uses convolutional neural networks for learn latent mappings. Among the baselines, (1-3) are traditional multi-linear TDs and (4-5) are non-linear ones. Note that CoSTCo was originally designed for continuous data, but can be easily fitted to binary domains (Appendix C.1).

<i>JHU</i>	RMSE ↓			MAPE ↓		
	Rank 3	Rank 5	Rank 10	Rank 3	Rank 5	Rank 10
GCP	0.847 ± 0.017	0.857 ± 0.008	0.873 ± 0.013	0.666 ± 0.003	0.668 ± 0.002	0.677 ± 0.001
NCPD	0.856 ± 0.015	0.861 ± 0.014	0.876 ± 0.013	0.666 ± 0.004	0.668 ± 0.005	0.674 ± 0.003
BPCP	0.852 ± 0.015	0.862 ± 0.014	0.881 ± 0.005	0.669 ± 0.002	0.671 ± 0.004	0.678 ± 0.001
VBGCP	0.681 ± 0.255	0.484 ± 0.104	0.567 ± 0.255	0.365 ± 0.055	0.299 ± 0.013	0.286 ± 0.024
GPTF	0.505 ± 0.019	0.508 ± 0.018	0.546 ± 0.014	0.496 ± 0.089	0.403 ± 0.014	0.391 ± 0.011
MDTF	0.554 ± 0.116	0.529 ± 0.098	0.522 ± 0.047	0.715 ± 0.041	1.030 ± 0.218	0.578 ± 0.074
ENTED	0.438 ± 0.026	0.426 ± 0.026	0.406 ± 0.042	0.313 ± 0.005	0.296 ± 0.011	0.292 ± 0.037
<i>Article</i>						
GCP	0.946 ± 0.003	0.939 ± 0.004	0.929 ± 0.005	0.511 ± 0.002	0.506 ± 0.002	0.499 ± 0.002
NCPD	0.937 ± 0.004	0.931 ± 0.006	0.923 ± 0.009	0.507 ± 0.003	0.502 ± 0.003	0.498 ± 0.004
BPCP	0.945 ± 0.003	0.940 ± 0.005	0.938 ± 0.006	0.513 ± 0.002	0.510 ± 0.003	0.509 ± 0.003
VBGCP	0.805 ± 0.026	0.804 ± 0.026	0.804 ± 0.026	0.342 ± 0.006	0.341 ± 0.006	0.340 ± 0.006
GPTF	0.630 ± 0.026	0.629 ± 0.028	0.655 ± 0.027	0.190 ± 0.005	0.197 ± 0.001	0.206 ± 0.003
MDTF	0.749 ± 0.030	0.783 ± 0.027	0.851 ± 0.079	0.216 ± 0.011	0.224 ± 0.011	0.253 ± 0.016
ENTED	0.620 ± 0.020	0.628 ± 0.026	0.636 ± 0.024	0.164 ± 0.003	0.170 ± 0.004	0.173 ± 0.002
<i>EMS</i>						
GCP	0.728 ± 0.049	0.852 ± 0.047	0.955 ± 0.008	0.431 ± 0.013	0.474 ± 0.006	0.562 ± 0.009
NCPD	0.835 ± 0.029	0.904 ± 0.014	0.968 ± 0.003	0.437 ± 0.009	0.478 ± 0.005	0.565 ± 0.014
BPCP	0.722 ± 0.082	0.869 ± 0.008	0.936 ± 0.005	0.429 ± 0.007	0.474 ± 0.013	0.542 ± 0.003
VBGCP	0.345 ± 0.064	0.343 ± 0.062	0.321 ± 0.048	0.428 ± 0.042	0.423 ± 0.040	0.398 ± 0.032
GPTF	0.530 ± 0.062	0.460 ± 0.048	0.403 ± 0.060	0.957 ± 0.149	0.637 ± 0.026	0.458 ± 0.063
MDTF	0.377 ± 0.055	0.381 ± 0.057	0.387 ± 0.055	0.667 ± 0.138	0.738 ± 0.119	0.809 ± 0.181
ENTED	0.305 ± 0.049	0.319 ± 0.056	0.301 ± 0.045	0.399 ± 0.058	0.414 ± 0.102	0.355 ± 0.027

Table 2: Count tensor completion.

Settings For baseline models, we mainly adopt their default settings. All stochastic methods are optimized using batch size 128. Moreover, gradient-based models are optimized using Adam with a learning rate chosen from $\{3 \times 10^{-3}, 1 \times 10^{-3}, 3 \times 10^{-4}, 1 \times 10^{-4}\}$, except GCP, whose default optimizer is L-BFGS. We test all methods with different tensor ranks ranging from $\{3, 5, 10\}$. For GP-based methods, we use 100 inducing points and RBF kernel with bandwidth 1.0, consistent with previous work (Zhe et al. 2016; Zhe and Du 2018). Note that, for ENTED, the inducing points number is 50 + 50 for u and v , respectively. More results on the influence of inducing point numbers can be found in Section 5.3.

Results The completion results are shown in Table 1. For DBLP, the results of BCP are not available, due to its limited scalability. Our model consistently outperforms competing models. In particular, we observe that GPTF and our model perform much better than other baselines, which shows the effectiveness of adopting non-linear mappings. Although CoSTCo is built upon non-linear CNNs, it performs poorly on these tasks. We hypothesize two reasons. When a binary tensor is generated from low-rank signals after non-linear transforms, *e.g.*, logistic transform, the probability is not necessarily low-rank (Lee and Wang 2021). Thus, it is more reasonable to factorize the natural parameter rather than original binary observations. Moreover, due to its highly unconstrained structures, CoSTCo easily over-

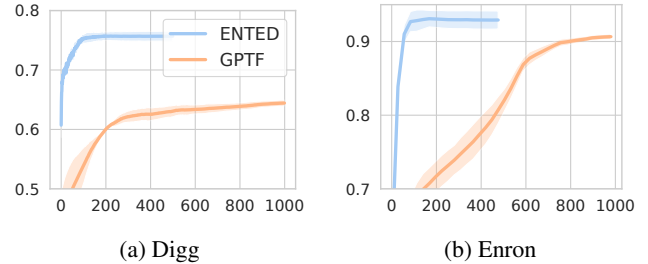


Figure 1: Convergence results. The x-axes are epochs and y-axes are AUC values.

fits for sparse tensors. Our model further outperforms GPTF, since we adopt more efficient covariance approximation and stochastic NGs. Moreover, due to the use of NGs, our model converges much faster than GPTF. The learning processes of rank 3 are illustrated in Fig. 1.

5.2 Count Tensor Completion

Datasets We evaluate the proposed model on three count tensors. (1) JHU (Dong, Du, and Gardner 2020), an order-4 tensor of shape $51 \times 3 \times 48 \times 8$, recording the Covid patient claims collected by JHU. The maximum count is 5182. The data is fully observed and we use 20% observations to predict the rest entries. (2) Article (Zhe and Du 2018), an order-3 tensor of shape $5 \times 1895 \times 2987$, extracted from

the DeskDrop dataset, recording user operations to articles. There are 50938 entries observed, and the maximum count is 76. (3) EMS (Zhe and Du 2018), an order-2 tensor recording the Emergency Medical Service (EMS) calls in Montgomery County, PA. The data shape is 72×69 , corresponding to $EMS\ title \times township$. There are 2494 observe entries and the maximum count is 545. We evaluate our model using the relative root mean square error (RMSE), mean absolute percentage error (MAPE), and negative log-likelihood (NLL). The definitions are shown in Appendix C.2.

Baselines We compare with six baselines. Apart from (1) GCP, we also compare with: (2) NCPD (Chi and Kolda 2012), a non-negative CP adopting Poisson likelihood. (3) BPCP (Schein et al. 2015), a Bayesian Poisson factorization with CP format. (4) VB-GCP (Soulat et al. 2021), a Bayesian version of GCP learned via variational inference. (5) GPTF (Zhe et al. 2016), a continuous GPTF using Gaussian likelihood. (6) MDTF (Fan 2022), a non-linear TD using neural networks to transform tensor factors. Similarly, (1-4) are multi-linear TDs and (5-6) are non-linear models. The settings are similar with Section 5.1. Therefore, we omit details here and present them in Appendix C.2.

Results The RMSE and MAPE results are shown in Table 2. Our model outperforms baselines using Gaussian, Poisson, or NB distributions. Compared with binary cases, the improvements over GPTF are much more significant in count datasets, which reveals the importance of choosing appropriate distributions. Besides, our model outperforms VBGCP, which also employs NB distribution, especially when the rank is small. This indicates the advantage of using non-linear structures. Table 3 shows the NLL of several probabilistic baselines. With a proper choice of the distribution, our model also achieves much better distributional estimation, as opposed to GPTF with Gaussian likelihood.

5.3 Additional Results on Inducing Points

Additional experiments are conducted to demonstrate the efficiency of our model. In specific, different inducing point numbers are tested, since it is essential for GP approximations. Apart from GPTF, we also compare with the model presented in Section 3.1, which is an extension of GPTF using PG augmentation and NG updates, denoted as GPTF-PG. To show the influence of the inducing points, we fit the model of rank 10 on the *Article* dataset, varying inducing point numbers in $\{2^3, 2^6, 2^7, 2^8, 2^9, 2^{10}\}$. Similarly, in ENTED, the inducing point numbers for u and v are equal.

Results of RMSE and computing times are shown in Fig. 2. In particular, Fig. 2a shows that our model consistently outperforms GPTF and GPTF-PG with different numbers of inducing points. For GPTF, more inducing points lead to worse performance, maybe due to over-fitting. However, for GPTF-PG and ENTED, the performance improves as the inducing points grow, since they adopt more structured inference and NG. Fig. 2b shows the computing time of one epoch, conducted on a single NVIDIA RTX A6000 GPU. Our model is slower than GPTF and GPTF-PG when the inducing points number is small, since we need to implement the NG updates manually. Nevertheless, when the in-

<i>JHU</i>	NLL ↓		
	Rank 3	Rank 5	Rank 10
GCP	INF	INF	INF
VBGCP	62.1 ± 0.5	64.7 ± 0.6	69.0 ± 0.5
GPTF	$237. \pm 40.$	$216. \pm 21.$	$185. \pm 25.$
ENTED	3.78 ± 0.01	3.74 ± 0.04	3.67 ± 0.07
<i>Article</i>			
GCP	7.17 ± 0.17	7.10 ± 0.11	7.16 ± 0.17
VBGCP	1.63 ± 0.01	1.63 ± 0.01	1.63 ± 0.01
GPTF	1.71 ± 0.15	1.65 ± 0.12	1.70 ± 0.11
ENTED	1.35 ± 0.01	1.35 ± 0.01	1.35 ± 0.01
<i>EMS</i>			
GCP	11.8 ± 1.3	16.4 ± 1.6	27.9 ± 2.1
VBGCP	23.2 ± 1.7	23.3 ± 1.7	22.5 ± 1.8
GPTF	62.2 ± 20.4	54.3 ± 10.9	43.7 ± 11.2
ENTED	2.95 ± 0.05	2.97 ± 0.08	2.91 ± 0.04

Table 3: NLL results of count tensor completion experiments. The NLL of GCP for the JHU dataset goes to infinity.

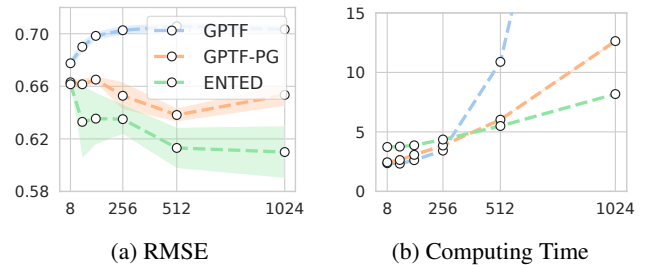


Figure 2: Results on different numbers of inducing points. The x-axes denote the number of inducing points. In subfigure (a), the y-axis is the RMSE and in subfigure (b), the y-axis is the computing time (in seconds) for each epoch.

ducing points number becomes larger, our model is notably faster than GPTF and GPTF-PG. This reveals the computational advantage of our model in handling complex datasets, where a large number of inducing points may be needed.

6 Conclusions

An efficient nonparametric tensor decomposition for binary and count data is presented. By replacing traditional multi-linear products with non-linear Gaussian processes (GP), the model can capture more complex relations among different tensor modes. By using the Pólya-Gamma augmentation, a conjugate model and natural gradient updates can be derived analytically. Moreover, we derived the sparse orthogonal variational inference framework to enable faster and more flexible covariance matrix approximation. Experiments are conducted on binary and count tensor completion tasks, to show the superior performance and computational benefits of our model. In the future, it is of interest to extend our model in streaming data and continual learning settings.

Acknowledgments

Zerui Tao was supported by the RIKEN Junior Research Associate Program. This work was supported by the JSPS KAKENHI Grant Numbers JP20H04249, JP23H03419.

References

- Bahadori, M. T.; Yu, Q. R.; and Liu, Y. 2014. Fast Multivariate Spatio-temporal Analysis via Low Rank Tensor Learning. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Chi, E. C.; and Kolda, T. G. 2012. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4): 1272–1299.
- Chu, W.; and Ghahramani, Z. 2009. Probabilistic Models for Incomplete Multi-dimensional Arrays. In van Dyk, D.; and Welling, M., eds., *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, 89–96. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR.
- Cichocki, A.; Lee, N.; Oseledets, I.; Phan, A.-H.; Zhao, Q.; Mandic, D. P.; et al. 2016. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5): 249–429.
- Dong, E.; Du, H.; and Gardner, L. 2020. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet infectious diseases*, 20(5): 533–534.
- Fan, J. 2022. Multi-Mode Deep Matrix and Tensor Factorization. In *International Conference on Learning Representations*.
- Gal, Y.; Van Der Wilk, M.; and Rasmussen, C. E. 2014. Distributed variational inference in sparse Gaussian process regression and latent variable models. *Advances in neural information processing systems*, 27.
- Glasser, I.; Sweke, R.; Pancotti, N.; Eisert, J.; and Cirac, I. 2019. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in neural information processing systems*, 32.
- Hensman, J.; Fusi, N.; and Lawrence, N. D. 2013. Gaussian Processes for Big Data. In *Uncertainty in Artificial Intelligence*, 282. Citeseer.
- Hensman, J.; Matthews, A.; and Ghahramani, Z. 2015. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, 351–360. PMLR.
- Hitchcock, F. L. 1927. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4): 164–189.
- Hong, D.; Kolda, T. G.; and Duersch, J. A. 2020. Generalized canonical polyadic tensor decomposition. *SIAM Review*, 62(1): 133–163.
- Ibrahim, S.; Fu, X.; Hutchinson, R.; and Seo, E. 2023. Under-Counted Tensor Completion with Neural Incorporation of Attributes. In *International Conference on Machine Learning*. PMLR.
- Kilmer, M. E.; Braman, K.; Hao, N.; and Hoover, R. C. 2013. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1): 148–172.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Klami, A. 2015. Polya-gamma augmentations for factor models. In *Asian Conference on Machine Learning*, 112–128. PMLR.
- Kolda, T. G.; and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500.
- Lee, C.; and Wang, M. 2021. Beyond the signs: Nonparametric tensor completion via sign series. *Advances in Neural Information Processing Systems*, 34: 21782–21794.
- Li, C.; and Sun, Z. 2020. Evolutionary topology search for tensor network decomposition. In *International Conference on Machine Learning*, 5947–5957. PMLR.
- Li, C.; Zeng, J.; Tao, Z.; and Zhao, Q. 2022. Permutation search of tensor network structures via local sampling. In *International Conference on Machine Learning*, 13106–13124. PMLR.
- Liu, H.; Li, Y.; Tsang, M.; and Liu, Y. 2019. Costco: A neural tensor completion model for sparse tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 324–334.
- Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2012. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 208–220.
- Novikov, G. S.; Panov, M. E.; and Oseledets, I. V. 2021. Tensor-train density estimation. In *Uncertainty in artificial intelligence*, 1321–1331. PMLR.
- Oseledets, I. V. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5): 2295–2317.
- Pan, Z.; Wang, Z.; and Zhe, S. 2020. Scalable nonparametric factorization for high-order interaction events. In *International Conference on Artificial Intelligence and Statistics*, 4325–4335. PMLR.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Polson, N. G.; Scott, J. G.; and Windle, J. 2013. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American statistical Association*, 108(504): 1339–1349.
- Qiu, H.; Li, C.; Weng, Y.; Sun, Z.; He, X.; and Zhao, Q. 2021. On the Memory Mechanism of Tensor-Power Recurrent Models. In Banerjee, A.; and Fukumizu, K., eds., *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, 3682–3690. PMLR.

- Rai, P.; Hu, C.; Harding, M.; and Carin, L. 2015. Scalable Probabilistic Tensor Factorization for Binary and Count Data. In *IJCAI*, 3770–3776.
- Rai, P.; Wang, Y.; Guo, S.; Chen, G.; Dunson, D.; and Carin, L. 2014. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *International Conference on Machine Learning*, 1800–1808. PMLR.
- Schein, A.; Paisley, J.; Blei, D. M.; and Wallach, H. 2015. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21th ACM SIGKDD International conference on knowledge discovery and data mining*, 1045–1054.
- Schein, A.; Zhou, M.; Blei, D.; and Wallach, H. 2016. Bayesian poisson tucker decomposition for learning the structure of international relations. In *International Conference on Machine Learning*, 2810–2819. PMLR.
- Shi, J.; Titsias, M.; and Mnih, A. 2020. Sparse orthogonal variational inference for Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 1932–1942. PMLR.
- Soulat, H.; Keshavarzi, S.; Margrie, T.; and Sahani, M. 2021. Probabilistic tensor decomposition of neural population spiking activity. *Advances in Neural Information Processing Systems*, 34: 15969–15980.
- Tao, Z.; Tanaka, T.; and Zhao, Q. 2023. Scalable Bayesian Tensor Ring Factorization for Multiway Data Analysis. In *International Conference on Neural Information Processing*, 490–503. Springer.
- Titsias, M. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, 567–574. PMLR.
- Tucker, L. R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311.
- Wang, M.; and Li, L. 2020. Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. *The Journal of Machine Learning Research*, 21(1): 6146–6183.
- Wang, Z.; Xu, Y.; Tillinghast, C.; Li, S.; Narayan, A.; and Zhe, S. 2022. Nonparametric Embeddings of Sparse High-Order Interaction Events. In *International Conference on Machine Learning*, 23237–23253. PMLR.
- Wenzel, F.; Galy-Fajou, T.; Donner, C.; Kloft, M.; and Opper, M. 2019. Efficient Gaussian process classification using Pölya-Gamma data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5417–5424.
- Xu, Z.; Yan, F.; and Qi, Y. 2012. Infinite tucker decomposition: nonparametric Bayesian models for multiway data analysis. In *Proceedings of the 29th International Conference on Machine Learning*, 1675–1682.
- Zhao, Q.; Sugiyama, M.; Yuan, L.; and Cichocki, A. 2019. Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 8608–8612. IEEE.
- Zhao, Q.; Zhang, L.; and Cichocki, A. 2015. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9): 1751–1763.
- Zhe, S.; and Du, Y. 2018. Stochastic nonparametric event-tensor decomposition. *Advances in Neural Information Processing Systems*, 31.
- Zhe, S.; Xu, Z.; Chu, X.; Qi, Y.; and Park, Y. 2015. Scalable nonparametric multiway data analysis. In *Artificial Intelligence and Statistics*, 1125–1134. PMLR.
- Zhe, S.; Zhang, K.; Wang, P.; Lee, K.-c.; Xu, Z.; Qi, Y.; and Ghahramani, Z. 2016. Distributed flexible nonlinear tensor factorization. *Advances in neural information processing systems*, 29.