ICML Paper Draft

Chanwoo Lee

Jan 12, 2020

1 Introduction

2 Model

2.1 Low-rank cumulative model

For the N-mode ordinal tensor $\mathcal{Y} = [\![y_{i_1,\cdots,i_N}]\!] \in \{1, 2, \cdots, K\}^{d_1 \times \cdots \times d_N}$, we assume its entries are realizations of independent multinomial random variables, such that

$$\mathbb{P}(y_{i_1,\cdots,i_N} = l) = f_l(\theta_{i_1,\cdots,i_N}), \quad (i_1, \cdots, i_N) \in [d_1] \times \cdots \times [d_N]. \tag{1}$$

In this model, a twice differentiable function $f_l : \mathbb{R} \to [0, 1]$ with $l \in [K]$ is strictly increasing and satisfies $\sum_{l=1}^{K} f_l(\theta) = 1$ for a fixed θ. The tensor $\Theta = [\![\theta_{i_1,\cdots,i_N}]\!] \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ is a hidden parameter which we are interested in. We assume the parameter tensor Θ is continuous value and admits a rank $\boldsymbol{r} = (r_1, \cdots, r_N)$ Tucker decomposition,

$$\Theta = \mathcal{C} \times_1 A_1 \cdots \times_N A_N, \tag{2}$$

where $\mathcal{C} \in \mathbb{R}^{r_1 \times \cdots \times r_N}$ is a core tensor and $A_n \in \mathbb{R}^{d_n \times r_n}$, for $n \in [N]$ is a factor matrix. We can get information about the influence of each mode by checking factor matrices.

The tensor model (1) has an equivalent representation as a latent model with K-level quantization. (Davenport et al. 2012; Lan, Studer, and Baraniuk 2014; Bhaskar and Javanmard 2015; Cai and Zhou 2013) where $\mathcal{Y} = [\![y_{i_1,\cdots,i_N}]\!]$ is a quantized value such that,

$$y_{i_1,\cdots,i_N} = \mathcal{Q}(\theta_{i_1,\cdots,i_N} + \epsilon_{i_1,\cdots,i_N}), \quad (i_1, \cdots, i_N) \in [d_1] \times \cdots [d_N], \tag{3}$$

where $\mathcal{E} = [\![\epsilon_{i_1,\cdots,i_N}]\!]$ is a noise tensor of *i.i.d.* from cumulative distribution function $\Phi(\cdot)$ and the function $\mathcal{Q} : \mathbb{R} \to [K]$ is a quantizer having the following rule.

$$\mathcal{Q}(x) = l, \text{ if } \omega_{l-1} < x \leq \omega_l, l \in [K], \tag{4}$$

where $\boldsymbol{\omega} = (\omega_1, \cdots, \omega_{K-1})$ is a cutpoints vector and $-\infty = \omega_0 < \omega_1 < \cdots < \omega_K = \infty$. That is, the entries of observed tensor $\mathcal{Y}$ fall in category l when the associated entries of the latent tensor $\Theta + \mathcal{E}$

fall in the lth interval of values. Then we have

$$\begin{aligned} f_l(\theta_{i_1,\cdots,i_N}) &= \mathbb{P}(y_{i_1,\cdots,i_N} = l) \\ &= \Phi(\omega_l - \theta_{i_1,\cdots,i_N}) - \Phi(\omega_{l-1} - \theta_{i_1,\cdots,i_N}). \end{aligned} \tag{5}$$

We can diversify our model by the choices of $\Phi(\cdot)$, or equivalently the distribution of $\mathcal{E}$ with given $\boldsymbol{\omega}$. The followings are 2 common choices of $\Phi(\cdot)$.

Example 1 (Logistic link/Logistic noise)**.** *The logistic model is represented by* (1) *with* $f_l(\theta) = \Phi_{log}(\frac{\omega_l-\theta}{\sigma}) - \Phi_{log}(\frac{\omega_{l-1}-\theta}{\sigma})$ *where* $\Phi_{log}(x/\sigma) = (1 + e^{-x/\sigma})$*. Equivalently, the noise* $\epsilon_{i_1,\cdots,i_N}$ *in* (3) *follows i.i.d. logistic distribution with the scale parameter* σ*.*

Example 2 (Probit link/Gaussian noise)**.** *The probit model is represented by* (1) *with* $f_l(\theta) = \Phi_{norm}(\frac{\omega_l-\theta}{\sigma}) - \Phi_{norm}(\frac{\omega_{l-1}-\theta}{\sigma})$ *where* Φ_{norm} *is the cumulative distribution function of the standard Gaussian. Equivalently, the noise* $\epsilon_{i_1,\cdots,i_N}$ *in* (3) *follows i.i.d.*$N(0, \sigma^2)$*.*

2.2 Rank-constrained likelihood-based estimation

Our goal is to estimate unknown parameter tensor Θ and a cutpoints vector $\boldsymbol{\omega}$ from observed tensor $\mathcal{Y}$ using a constrained likelihood approach. The log-likelihood function for (1) is

$$\mathcal{L}_{\mathcal{Y}}(\Theta, \boldsymbol{\omega}) = \sum_{i_1,\cdots,i_N} \left[\sum_{l\in[K]} \log(f_l(\theta_{i_1,\cdots,i_N})) \mathbb{1}_{\{y_{i_1,\cdots,i_N}=l\}} \right], \tag{6}$$

where the cutpoints vector $\boldsymbol{\omega}$ is implicitly contained in the function f_l. Considering the Tucker structure in (2), we have the constrained optimization problem.

$$\begin{aligned} &\max_{(\Theta,\boldsymbol{\omega})\in\mathcal{D}\times\mathcal{W}} \mathcal{L}_{\mathcal{Y}}(\Theta, \boldsymbol{\omega}), \text{ where} \\ &\mathcal{D} = \{\Theta \in \mathbb{R}^{d_1\times\cdots\times d_N} : \text{rank}(\Theta) = \boldsymbol{r} \text{ and } \|\Theta\|_\infty \le \alpha\} \\ &\mathcal{W} = \{\boldsymbol{\omega} \in \mathbb{R}^{K-1} : \omega_1 < \cdots < \omega_{K-1}\}, \end{aligned} \tag{7}$$

for a given rank $\boldsymbol{r} \in {\mathbb{N}_+}^N$ and a bound $\alpha \in \mathbb{R}_+$. The search space $\mathcal{D}$ has two constraints on unknown parameter Θ. The first constraint ensure that the unknown parameter Θ admits the Tucker decomposition with rank $\boldsymbol{r}$. The second constraint makes the entries of Θ bounded by a constant α. This bound condition is a technical assumption to help to recover Θ in the noiseless case. Similar conditions has been imposed in many literatures for the matrix case. (Davenport et al. 2012; Bhaskar and Javanmard 2015; Cai and Zhou 2013; Bhaskar 2016). The search space $\mathcal{W}$ makes sure that the probability function f_l in (5) is positive.

2.3 Optimization

In this section, we describe the algorithm to seek the optimizer of (7). The objective function $\mathcal{L}_{\mathcal{Y}}(\Theta, \boldsymbol{\omega})$ is concave in $(\Theta, \boldsymbol{\omega})$ whenever $\Phi(x)$ is log-concave (McCullagh 1980; Burridge 1981). However, the feasible set $\mathcal{D}$ is not a convex set, which makes the optimization (7) a non-convex problem. One approach to handle this problem is utilizing the Tucker decomposition and converting optimization into a block-wise convex problem. Let us denote the objective function as

$$\mathcal{L}(\mathcal{C}, A_1, \cdots, A_N, \boldsymbol{\omega}) = \mathcal{L}_{\mathcal{Y}}(\mathcal{C} \times_1 A_1 \cdots \times_N A_N, \boldsymbol{\omega}). \tag{8}$$

From (8), we have $N+2$ blocks of variables in the objective function, one for the cutpoints vector $\boldsymbol{\omega}$, one for the core tensor $\mathcal{C}$ and N for the factor matrices A_n's. We notice that the optimization problem becomes simple convex problem if any $N+1$ out of the $N+2$ blocks being fixed. Therefore, we can alternatively update one block at a time while other blocks being fixed. The algorithm 1 gives the full description.

Algorithm 1 Ordinal tensor decomposition

Input: Ordinal tensor $\mathcal{Y} \in \{1, \cdots, K\}^{d_1 \times \cdots \times d_N}$, rank $\boldsymbol{r} \in {\mathbb{N}_+}^{K-1}$, entrywise bound $\alpha \in \mathbb{R}_+$.
Output: Solution $(\hat{\Theta}, \hat{\boldsymbol{\omega}}) = \arg\max_{(\Theta, \boldsymbol{\omega}) \in \mathcal{D} \times \mathcal{W}} \mathcal{L}_{\mathcal{Y}}(\Theta, \boldsymbol{\omega})$.

Initialize random tensor $\mathcal{C}^{(0)} \in \mathbb{R}^{r_1 \times \cdots \times r_N}$, matrices $A^{(0)} = \{A_1^{(0)}, \cdots, A_N^{(0)}\}$ and $\boldsymbol{\omega}^{(0)} = (\omega_1^{(0)}, \cdots, \omega_{(K-1)}^{(0)})$ such that $\omega_1^{(0)} < \cdots < \omega_{K-1}^{(0)}$.
for t = 1,2,$\cdots$, **do** until convergence,
  Update A_n
  for n = 1,2,$\cdots$,N **do**
    $A_n^{(t+1)} \leftarrow \arg\max_{A_n \in \mathbb{R}^{d_n \times r_n}} \mathcal{L}\big(\mathcal{C}^{(t)}, A_1^{(t+1)}, \cdots, A_n, \cdots, A_N^{(t)}, \boldsymbol{\omega}^{(t)}\big)$
      such that $\|\mathcal{C}^{(t)} \times_1 A_1^{(t+1)} \cdots \times_n A_n \cdots \times_N A_N^{(t)}\|_\infty \leq \alpha$
  end for
  Update $\mathcal{C}$
  $\mathcal{C}^{(t+1)} \leftarrow \arg\max_{\mathcal{C} \in \mathbb{R}^{r_1 \times \cdots \times r_N}} \mathcal{L}\big(\mathcal{C}, A_1^{(t+1)}, \cdots, A_N^{(t+1)}, \boldsymbol{\omega}^{(t)}\big)$
    such that $\|\mathcal{C} \times_1 A_1^{(t+1)} \cdots \times_N A_N^{(t+1)}\|_\infty \leq \alpha$
  $\Theta^{(t+1)} \leftarrow \mathcal{C}^{(t+1)} \times_1 A_1^{(t+1)} \cdots \times_N A_N^{(t+1)}$
  Update $\boldsymbol{\omega}$
  $\boldsymbol{\omega}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\omega} \in \mathbb{R}^{K-1}} \mathcal{L}_{\mathcal{Y}}\big(\Theta^{(t+1)}, \boldsymbol{\omega}\big)$ such that $\omega_1^{(0)} < \cdots < \omega_{K-1}^{(0)}$.
end for
return $\Theta, \boldsymbol{\omega}$

2.4 Rank selection

Algorithm 1 takes the rank $\boldsymbol{r}$ as an input variable. In practice, the rank $\boldsymbol{r}$ is hardly known. Estimating an appropriate rank $\boldsymbol{r}$ from a given tensor is an important issue. We suggest to use

Bayesian Information Criterion(BIC) to choose the rank.

$$\begin{aligned}\hat{\boldsymbol{r}} &= \arg\min_{\boldsymbol{r}\in\mathbb{N}_+^N} BIC(\boldsymbol{r}) \\ &= \arg\min_{\boldsymbol{r}\in\mathbb{N}_+^N} \Big[-2\mathcal{L}_{\mathcal{Y}}(\hat{\Theta}(\boldsymbol{r})) + \big(\prod_{n\in[N]} r_n + \sum_{n\in[N]} (d_n r_n - r_n^2) \big) \log(\prod_{n\in[N]} d_n) \Big],\end{aligned} \tag{9}$$

where $\hat{\Theta}(\boldsymbol{r})$ is a maximum likelihood estimater given the rank $\boldsymbol{r}$. The second part of the summation in (9) is the number of independent parameters in the model. We select the rank $\hat{\boldsymbol{r}}$ that minimizes BIC value through the grid search method.

3 Real-world Data Applications

In this section, we apply our ordinal tensor decomposition method to two real-world datasets of ordinal tensors. In the first application, we use our model to analyze an ordinal tensor consisting of structural connectivity patterns among 68 brain regions for 136 individuals from Human Connectome Project (HCP) (Geddes 2016). In the second application, the dataset is a tensor with missing values. This tensor records the ratings from scale 1 to 5 of 42 users to 139 songs on 26 contexts (Baltrunas et al. 2011).

3.1 Human Connectome Project (HCP)

The human connectome project (HCP) is a $68 \times 68 \times 136$ tensor where 68 modes represent brain regions and 136 modes are individuals. All the individual images were preprocessed following a standard pipeline (Zhang et al. 2018), and the brain was parcellated to 68 regions of interest following the Desikan atlas (Desikan et al. 2006). The tensor entries consist of $\{1, 2, 3\}$, the strength of fiber connections between 68 brain regions for each of the 136 individuals. First, we apply our ordinal tensor decomposition method with a logistic link function to the HCP data and identify similar brain nodes based on latent parameters. The BIC result suggests $\boldsymbol{r} = (23, 23, 8)$ with $\mathcal{L}_{\mathcal{Y}}(\hat{\Theta}, \hat{\boldsymbol{\omega}}) = -216645.8$.

Also, we compare our ordinal tensor decomposition method with a logistic link function with the classical continuous Tucker decomposition method. We use 5 folded cross validation method for the comparison. Specifically, we randomly split the tensor entries into 5 similar sized pieces and alternatively use each piece of entries as a test data using other 80% entries as a training data. The entries in the test data are encoded as missing and then predicted based on the tensor decomposition from the training data. We set the rank of the tensor as $\boldsymbol{r} = (23, 23, 8)$. Table 1 shows RMSE,

	Tensor decomposition method					
	Ordinal (logistic link)			Continuous-valued		
	RMSE	MAE	Error	RMSE	MAE	Error
Average	0.1503711	0.1502662	0.1502151	0.1603685	0.1600219	0.1598499

Table 1: Tensor completion for the HCP data. Two methods are compared: the proposed ordinal tensor decomposition and the classical continuous Tucker decomposition.

MAE and error(false prediction rate), averaged over 5 test set results. We can check that our ordinal tensor decomposition model outperforms the classic continuous decomposition in all criteria.

3.2 Music data with missing values

InCarMusic is a mobile application that offers music recommendation to passengers of cars based on contexts (Baltrunas et al. 2011). Our goal is to get the tensor completion from the $42 \times 139 \times 26$ tensor using the ordinal tensor decomposition model and thereby we can offer context-specific music recommendation to users. The tensor entries consist of $\{1, 2, 3, 4, 5\}$, the ratings of 42 users to 139 songs on 26 contexts and are encoded as -1 for missing values. The number of missing values is 148904 and the number of available values is 2884.

References

Baltrunas, Linas et al. (2011). "InCarMusic: Context-Aware Music Recommendations in a Car". In: *EC-Web*.

Bhaskar, Sonia A. (2016). "Probabilistic Low-Rank Matrix Completion from Quantized Measurements". In: *J. Mach. Learn. Res.* 17, 60:1–60:34.

Bhaskar, Sonia A. and Adel Javanmard (2015). "1-bit matrix completion under exact low-rank constraint". In: *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6.

Burridge, Jim (1981). "A Note on Maximum Likelihood Estimation for Regression Models using Grouped Data". In:

Cai, Tony and Wen-Xin Zhou (2013). "A max-norm constrained minimization approach to 1-bit matrix completion". In: *J. Mach. Learn. Res.* 14, pp. 3619–3647.

Davenport, Mark A. et al. (2012). "1-Bit Matrix Completion". In: *ArXiv* abs/1209.3672.

Desikan, Rahul S. et al. (2006). "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest". In: *NeuroImage* 31, pp. 968–980.

Geddes, Linda (2016). "Human brain mapped in unprecedented detail". In:

Lan, Andrew S., Christoph Studer, and Richard G. Baraniuk (2014). "Matrix recovery from quantized and corrupted measurements". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4973–4977.

McCullagh, Peter (1980). "Regression Models for Ordinal Data". In:

Zhang, Zhengwu et al. (2018). "Mapping population-based structural connectomes". In: *NeuroImage* 172, pp. 130–145.