Efficient Bayesian Model Selection in PARAFAC via Stochastic Thermodynamic Integration

Thanh Huy Nguyen, *Student Member, IEEE,* Umut Şimşekli, *Member, IEEE,* Gaël Richard, *Fellow, IEEE*
Ali Taylan Cemgil, *Member, IEEE*

***Abstract*—Parallel factor analysis (PARAFAC) is one of the most popular models in the field tensor factorization. Even though it has proven successful in diverse application fields, the performance of PARAFAC usually hinges up on the rank of the factorization, which is typically specified manually by the practitioner. In this study, we develop a novel parallel and distributed Bayesian model selection technique for rank estimation in large-scale PARAFAC models. The proposed approach integrates ideas from the emerging field of stochastic gradient Markov Chain Monte Carlo, statistical physics, and distributed stochastic optimization. As opposed to the existing methods, which are based on certain heuristics, our method has a clear mathematical interpretation, and has significantly lower computational requirements, thanks to data sub-sampling and parallelization. We provide formal theoretical analysis on the bias induced by the proposed approach. Our experiments on synthetic and large-scale real datasets show that our method is able to find the optimal model order while being significantly faster than the state-of-the-art.**

***Index Terms*—Tensor factorization, PARAFAC, Bayesian model selection, Markov Chain Monte Carlo.**

I. Introduction

PARAFAC decomposition is one of the most popular tensor factorization approaches, which have a variety of applications in signal processing [1], [2], computer vision [3], [4], data mining [5], [6], neuroscience [7], [8], chemometrics [9], [10], and psychometrics [9], [11]. Here, the aim is to decompose an observed *three-way tensor* $\mathbf{X} \equiv \{x_{ijk}\}_{i,j,k} \in \mathbb{R}^{I\times J\times K}$ into an outer product of three different matrices, $\mathbf{A} \equiv \{a_{ir}\}_{i,r} \in \mathbb{R}^{I\times R}$, $\mathbf{B} \equiv \{b_{jr}\}_{j,r} \in \mathbb{R}^{J\times R}$, and $\mathbf{C} \equiv \{c_{kr}\}_{k,r} \in \mathbb{R}^{K\times R}$, given as follows:

$$x_{ijk} \approx \hat{x}_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}. \tag{1}$$

Here, the observed tensor $\mathbf{X}$ is approximated as a sum of R different 'rank-one' tensors (1), where we call a rank-one tensor as the outer products of three vectors. Accordingly, R is called the rank of the PARAFAC model.

The performance of PARAFAC-based algorithms usually hinges up on the rank of the factorization. Automatic estimation of this rank turns out to be a challenging task, and there has been several attempts to address it, to name a few [10], [12-16]. The common theme in these approaches is that they are first based on some other matrix/tensor decomposition techniques, such as singular value decomposition (SVD), higher-order SVD, and matrix diagonalization, and then they are appended with certain heuristics. Even though these methods have proven useful in certain applications, they often have at least one of the two major problems. Firstly, they do not have a clear mathematical interpretation, since they are based on heuristics. Secondly, the performance of these methods might be limited in large-scale problems, since they often require computationally expensive matrix operations.

In this study, we propose a novel Bayesian model selection technique for rank estimation in PARAFAC models. In particular, we develop a marginal likelihood estimation method that is based on the recently developed Stochastic Thermodynamic Integration (STI) algorithm [17], which combines ideas from stochastic optimization, Markov Chain Monte Carlo (MCMC), and statistical physics. We then propose a novel parallel and distributed variant of STI by exploiting the conditional independence structure of the PARAFAC models, so that the computational complexity of the resulting algorithm can be further reduced by a dramatic factor. We further improve the convergence speed of this approach by incorporating the local geometry of the problem. We provide formal theoretical analysis, where we show that the bias induced by the ultimately proposed method is bounded under certain regularity conditions. We illustrate the proposed methods on both synthetic and real datasets. Our results show that the algorithms can successfully estimate the rank of PARAFAC models with a low computational budget, even in large-scale distributed settings.

 T. H. Nguyen, U. Şimşekli and Gaël Richard are with the LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France. A. T. Cemgil is with the Department of Computer Engineering, Boğaziçi University, Bebek, 34342, Istanbul, Turkey. E-mail: {thanh.nguyen, umut.simsekli, gael.richard}@telecom-paristech.fr, taylan.cemgil@boun.edu.tr

Manuscript received December 22, 2017.

II. Preliminaries

A. Probabilistic interpretation

In this study, we consider a probabilistic PARAFAC model that has the following hierarchical generative structure:

$$p(\mathbf{A}) = \prod_{i,r} p(a_{ir}),\ \ p(\mathbf{B}) = \prod_{j,r} p(b_{jr}),\ \ p(\mathbf{C}) = \prod_{k,r} p(c_{kr})$$
$$p(\mathbf{X}|\mathbf{A},\mathbf{B},\mathbf{C}) = \prod_{i,j,k} p(x_{ijk}|\mathbf{A}_{i:},\mathbf{B}_{j:},\mathbf{C}_{k:}) \tag{2}$$

where $p(a_{ir}), p(b_{jr})$, and $p(c_{kr})$ are called the prior distributions, $p(x_{ijk}|\cdot)$ is called the likelihood function, and $\mathbf{M}_{i:}$ denotes the i^{th} column of a matrix $\mathbf{M}$. This probabilistic approach generalizes the classical cost-minimization-based formulation of PARAFAC [9], as one can show that

it corresponds to a *maximum a-posteriori* estimation in the probabilistic model defined in (2).

B. Stochastic Thermodynamic Integration

Bayesian model selection techniques require the computation of the marginal likelihood of a given model, that is defined as follows:

$$p(x|m) = \int p(x|\theta, m)p(\theta|m)\mathrm{d}\theta, \tag{3}$$

where $x = \{x_n\}_{n=1}^N$ denotes a set of i.i.d random variables considered as the observed data, θ is a latent variable, and $m \in \{1, \dots, M\}$ denotes the model-order, which will be our main source of interest. In this setting, $p(x|\theta, m)$ and $p(\theta|m)$ are respectively the likelihood and the prior of the model.

In model selection applications, our aim is to find the model-order $m^\star$ that maximizes the marginal likelihood, given as follows: $m^\star = \arg\max_m \int p(x|\theta, m)p(\theta|m)\mathrm{d}\theta$. Unfortunately, computing $m^\star$ turns out to be intractable except for simple models, motivating the need for approximate methods.

In this study, we consider the recently proposed the STI algorithm [17], which combines ideas from the newly emerging field of stochastic gradient MCMC (SG-MCMC) and statistical physics. The STI algorithm aims at directly computing the logarithm of the marginal likelihood by introducing a temperature variable t and making use of the following identity [18]:[1]

$$\log p(x) = \int_0^1 \langle \log p(x|\theta) \rangle_{p(\theta|t)} \mathrm{d}t \tag{4}$$

where $\langle f(x) \rangle_{q(x)}$ denotes the expectation of a function $f(x)$ under the distribution $q(x)$. Here, the key quantity $p(\theta|t)$ constitutes a 'geometric path' from $p(\theta)$ to $p(x|\theta)$, and formally defined as follows: $p(\theta|t) \propto p(\theta)p(x|\theta)^t$, where $\propto$ denotes proportionality up to a positive multiplicative constant.

The main idea in STI is to approximate the one-dimensional integration over t by using a deterministic numerical integration method and approximate the expectations by using SG-MCMC. In particular, for approximating the integration over t, STI uses a trapezoidal rule, given as follows: $\log p(x) \approx$

$$\sum_{i=0}^{T-1} \Delta t_i \frac{\langle \log p(x|\theta) \rangle_{p(\theta|t_i)} + \langle \log p(x|\theta) \rangle_{p(\theta|t_{i+1})}}{2} \tag{5}$$

where $0 = t_0 < t_1 < \ldots < t_T = 1$ and $\Delta t_i = t_{i+1} - t_i$. For the expectations in (5), STI uses an SG-MCMC algorithm, namely the stochastic gradient Langevin dynamics (SGLD), which iteratively applies the following update equation for generating samples from the distribution $p(\theta|t)$:

$$\begin{aligned}\theta^{(t,l)} =& \theta^{(t,l-1)} + \epsilon^{(t,l)} \Big(\frac{N}{N_s} t \sum_{n \in S^{(t,l)}} \nabla_\theta \log p(x_n|\theta^{(t,l-1)}) \\ &+ \nabla_\theta \log p(\theta^{(t,l-1)}) \Big) + \eta^{(t,l)},\end{aligned} \tag{6}$$

where $\theta^{(t,l)}$ denotes the samples (asymptotically) drawn from $p(\theta|t)$. Here, $\epsilon^{(t,l)}$ denotes the step-sizes, and $\eta^{(t,l)}$ is Gaussian noise: $\eta^{(t,l)} \sim \mathcal{N}(0, 2\epsilon^{(t,l)}\mathbf{I})$ with $\mathbf{I}$ being the identity matrix, $S^{(t,l)}$ denotes random subsets of $[N] \triangleq \{1, 2, \dots, N\}$, and $N_s = |S^{(t,l)}|$ is the size of each $S^{(t,l)}$. In an algorithmic sense, this algorithm is identical to the well-known optimization algorithm, stochastic gradient descent (SGD), except that it injects an additional Gaussian noise at each iteration.

By using the samples $\theta^{(t,l)}$, STI finally approximates the expectations by using sample averages, given as follows:

$$\langle \log p(x|\theta) \rangle_{p(\theta|t)} \approx \frac{1}{L} \frac{N}{N_s} \sum_{l=1}^{L} \sum_{n \in S^{(t,l)}} \log p(x_n|\theta^{(t,l)}) \tag{7}$$

where the same data subsamples $S^{(t,l)}$ are used in both (6) and (7). Verbally, STI generates a sample by using (6) and immediately evaluates its loglikelihood in (7). These computations are then ultimately used in (5). Thanks to data subsampling, STI forms a powerful yet a simple algorithm that can be suitable for large-scale problems.

III. Parallel and Distributed STI for PARAFAC

In this section, we will customize the STI algorithm for the rank estimation problem in PARAFAC. We first represent the PARAFAC model defined in (2) within the notation introduced in (3) by setting $x \equiv \{x_{ijk}\}_{i,j,k} \in \mathbb{R}^{IJK}$, a vector containing all the observations, and $\theta \equiv \{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \in \mathbb{R}^{IR+JR+KR}$, a vector containing all the entries of the hidden matrices.

In this context, we apply STI on the model given in (2), for estimating the rank R. Once the samples $\theta^{(t,l)}$ are generated for a given rank R, then the log-marginal likelihood for this rank $\log p(\mathbf{X}|R)$ can be approximated by using (7) and (5).

A. Non-negative PARAFAC models

In certain applications, all the elements in the observed tensor $\mathbf{X}$ and the hidden factors $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are required to be non-negative; resulting in a *non-negative* PARAFAC decomposition [19-21]. In such cases, the SGLD update rules will not be applicable since they might result in samples with negative entries due to the additive update rules.

If a non-negative PARAFAC problem is considered, by following [22-24], we propose to make use of a *mirroring trick* at each update step: if there are negative elements in the updated latent variables, we replace them by their absolute values. This operation does not violate the convergence guarantees [23].

B. Parallel / distributed implementation

The main computational advantage of STI stems from the fact that it uses data subsampling. However, we can improve the efficiency of the algorithm even more by using a systematic subsampling scheme, rather than drawing arbitrary sub-samples. In this section we extend SGLD (6), by taking the multi-linear structure of the PARAFAC model into account and we will show that this approach significantly reduces the computational needs by enabling parallelism.

Our approach is inspired by the distributed SGD algorithm for PARAFAC, which was proposed in [25]. In order to parallelize SGLD, we first need to carefully partition the observed data into mutually disjoint subsets, and also partition the latent variables according to these subsets. An example

[1]For simplicity, we further ignore the order m of the model and consider the following definition for the marginal likelihood: $p(x) = \int p(x|\theta)p(\theta)\mathrm{d}\theta$.

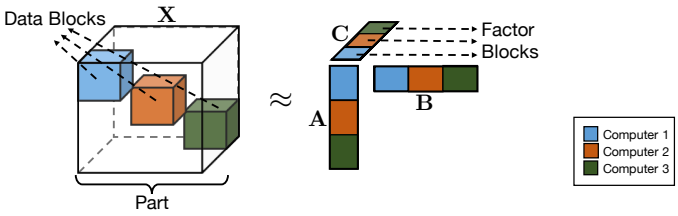

Fig. 1: Illustration of the parts and blocks.

of such a partitioning scheme is shown in Fig. 1. Here, the observed tensor $\mathbf{X}$ is partitioned into $3\times3\times3$ disjoint 'blocks' and the hidden factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are partitioned accordingly into 3 blocks. At each iteration, we will subsample 3 blocks from $\mathbf{X}$ (i.e., the smaller cubes, shown in different colors), such that these blocks will not intersect. We will call the combinations of such blocks as the 'parts'. With this partitioning scheme, the blue (orange, green, respectively) block is used only while updating the corresponding blue (orange, green, respectively) block of the three latent matrices. Therefore we can update the blue, orange, and green block at the same time, i.e. in parallel, without any conflicts. At the end of each iteration, only the blocks of the factor matrices need to be communicated among the processors, which typically yields a negligible communication cost. In the general case, the data will be partitioned into $B \times B \times B = B^3$ blocks and from these blocks we can form B^2 valid parts. Accordingly, the factor blocks will be partitioned into B blocks. We formally define this procedure along with the blocks and the parts in the supplementary document. Note that, as the stochastic gradients are still unbiased, the same theoretical properties hold.

IV. Extending STI with Preconditioned SGLD

Even though SGLD has proven successful in many applications, it might suffer from poor convergence rates when the target distribution has scale differences across dimensions [26], [27]. As a remedy, Li et al. [26] proposed the preconditioned SGLD (PSGLD) algorithm by extending SGLD with a diagonal preconditioning matrix $\mathbf{G}(\theta)$ that aims to capture the local geometry of the target densities. The PSGLD algorithm applies the following update rules for sampling from the distribution $p(\theta|t)$: $\theta^{(t,l)} = \theta^{(t,l-1)} +$

$$\epsilon^{(t,l)}\Big[\mathbf{G}(\theta^{(t,l-1)})\Big(\frac{Nt}{N_s}\sum_{n\in S^{(t,l)}}\nabla_\theta \log p(x_n|\theta^{(t,l-1)}) + \nabla_\theta \log p(\theta^{(t,l-1)})\Big)\Big] + \sqrt{\mathbf{G}(\theta^{(t,l-1)})}\eta^{(t,l)}. \quad (8)$$

Here, $\mathbf{G}(\theta)$ is defined as follows: (for $\lambda > 0$)

$$\mathbf{G}(\theta^{(t,l)}) \triangleq \text{diag}\Big(\mathbf{1} \oslash \big(\lambda\mathbf{1} + \sqrt{\mathbf{v}(\theta^{(t,l)})}\big)\Big), \quad (9)$$

where (for $\alpha \in [0,1]$)

$$\mathbf{v}(\theta^{(t,l)}) \triangleq \alpha\mathbf{v}(\theta^{(t,l-1)}) + (1-\alpha)\bar{\mathbf{g}}^{(t,l-1)} \odot \bar{\mathbf{g}}^{(t,l-1)}$$
$$\bar{\mathbf{g}}^{(t,l-1)} \triangleq (t/N_s)\sum\nolimits_{n\in S^{(t,l)}} \nabla_\theta \log p(x_n|\theta^{(t,l-1)}).$$

The operators $\odot$ and $\oslash$ denote the element-wise product and division, respectively, and $\mathbf{1}$ denotes a vector composed of ones. The matrix $\mathbf{G}$ aims to approximate the diagonal part of the inverse Fisher information matrix and in practice it makes the step-sizes more adaptive, i.e., flat directions will have larger step-sizes than the curved directions.

Despite the fact that PSGLD can achieve a better rate of convergence when compared to SGLD, the preconditioning scheme unfortunately introduces an additional bias [26]. In the sequel, we analyze the overall bias that is induced by the STI algorithm when it is combined with the PSGLD algorithm for generating samples.

Theorem 1. *Let $\mathcal{L} = \int_0^1 f(t)dt$ be the log-marginal likelihood (Eq. (4)) with $f(t) = \langle \log p(x|\theta)\rangle_{p(\theta|t)}$ and $\hat{\mathcal{L}}$ be the estimator of $\mathcal{L}$ by STI (Eq. (7), (5)) using pSGLD as the sampling method for $\theta^{(t,l)}$ with constant step-size ϵ. Under certain regularity conditions, the following bound holds:*

$$\Big|\langle\hat{\mathcal{L}}\rangle - \mathcal{L}\Big| = \mathcal{O}\left(\frac{1}{K\epsilon} + \epsilon + \frac{1}{T^2} + \frac{1-\alpha}{\alpha^{3/2}}\right), \quad (10)$$

where $\Delta_i = 1/T$ for all $i = 1,\dots,T$.

The detailed proof and the required conditions are provided in the supplementary document. This bound is identical to the one of STI with standard SGLD [17], except that the last term in the right hand side of (10) is introduced by PSGLD. However, in practice α is usually set to a value that is close to 1, therefore this additional bias can be neglected.

Due to the multi-linear structure of the PARAFAC, PSGLD can be also easily parallelized by using the same approach described in Sec. III-B. However, in this case we would need to partition and communicate the preconditioning variable $\mathbf{v}$, as well as the hidden factor matrices, which would result in a slightly increased communication cost.

V. Experiments

In order to evaluate the proposed algorithms, we conduct several experiments. We first apply STI with SGLD (STI-SGLD) and STI with pSGLD (STI-PSGLD) on a simple Gaussian model, whose marginal likelihood is analytically available. We show that both algorithms yield accurate estimates, whereas STI-PSGLD attains a faster convergence rate as expected. Due to space constraints, we provide the results of those experiments in the supplementary document.

In the rest of this section, we will present our experiments on a non-negative probabilistic PARAFAC model that has the following probabilistic generative structure:

$$a_{ir} \sim \mathcal{E}(\lambda_a), \quad b_{jr} \sim \mathcal{E}(\lambda_b), \quad c_{kr} \sim \mathcal{E}(\lambda_c)$$
$$x_{ijk}|\mathbf{A}_{i:},\mathbf{B}_{j:},\mathbf{C}_{k:} \sim \mathcal{PO}(\sum\nolimits_{r=1}^{R} a_{ir}b_{jr}c_{kr}) \quad (11)$$

where $\mathcal{E}$ and $\mathcal{PO}$ denote the exponential and Poisson distributions, respectively.

We carry out all the experiments on a Dell desktop with 3.2 GHz Quad-core Intel Xeon, 12 GB of memory. We do not use parallelization for the experiments on the synthetic data and we run the experiments in Python. On the other hand, we perform the real data experiments by using the parallel scheme in a simulated distributed environment with a single computer, where we implement the proposed algorithm in C with the Open MPI library for parallel computations.

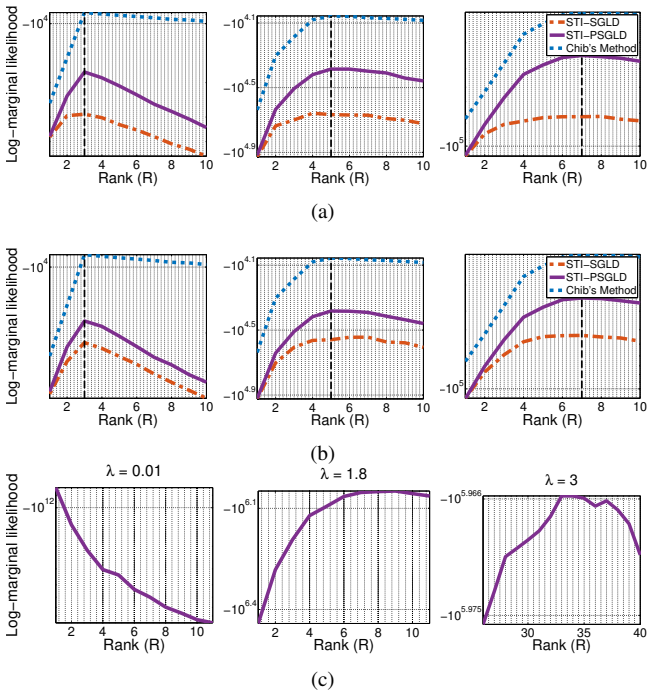

Fig. 2: Simulation results on 2(a) synthetic data with small number of iterations; 2(b) synthetic data with large number of iterations; and 2(c) Facebook dataset.

A. Experiments on synthetic data

In this section, we will present our experiments that we conduct on synthetic data. We first generate $\mathbf{A}, \mathbf{B}, \mathbf{C}$ by using generative model (11). We then estimate the log-marginal likelihood of the model for different values of the true rank. Unfortunately, the marginal likelihood of this model does not have an explicit analytical expression. Therefore, we compare the proposed approaches with an existing marginal likelihood estimation algorithm, so called Chib's method [28], [29]. This method is known to be unbiased, however its computational cost rapidly increases with the size of the data and can therefore only be applied to very small-sized problems.

In these experiments, we set $I = 10$, $J = 15$, $K = 20$, $\lambda_a = \lambda_b = \lambda_c = 3$ and we estimate the log-marginal likelihood for $R \in \{1, \ldots, 11\}$ using STI-SGLD, STI-PSGLD, and the Chib's method. We set $T = 10$, $N_s = IJK/25$. We then generate $L = 2000$ samples for each temperature t_i (for both STI-SGLD and STI-PSGLD) and use the last 500 samples for approximating the expectations. During the burn-in period, we use a decreasing the step-size $\epsilon = (a_\epsilon/l)^b_\epsilon$ with $a_\epsilon = 10^{-8}$, $b_\epsilon = 0.51$ and keep the step-size fixed after burn-in, where l denotes the iteration number of an SGLD or a PSGLD run. For Chib's method, we generate 800 samples for each rank.

The results are shown in Fig. 2(a). As can be seen from the figure, STI-PSGLD performs better than STI-SGLD: the estimates obtained via STI-PSGLD are closer to the ones of Chib's method, while predicting well the true rank of the model. The gap between the log-marginal likelihood estimates obtained by Chib's method and our methods is caused by the fact that our methods are biased. Nonetheless, this gap does not prevent the methods to correctly estimate the optimal rank.

Next, we set $L = 4000$ and use the last 500 samples for estimating the parameters. The results are shown in Fig. 2(b). We can observe that, when we increase the number of iterations, the estimates obtained via STI-SGLD tend to those of STI-PSGLD, which are almost unchanged when compared with the previous experiment. This result illustrates the advantage of STI-PSGLD in terms of speed of convergence.

B. Experiments on real data

In this section, we apply our proposed method to a real large-scale dataset, called the Facebook dataset [30]. This dataset is represented as a three-way tensor of dimensions: $42390 \times 39986 \times 1506$, which contains the information about which user posted on another user's wall on what date (User, User, Date). We model this dataset by using the model defined in (11) and only consider the parallel variant of STI-PSGLD for determining the optimal rank for the PARAFAC decomposition for the given prior distribution. We estimate the log-marginal likelihood for $R \in \{1, \ldots, 11\}$, we set $T = 5$, $\lambda_a = \lambda_b = \lambda_c = \lambda$, then generate $K = 3000$ samples at each run and use the last 500 samples for estimating the expectations. For parallelization, we choose $B = 12$.

The experiments show that the optimal rank changes depending on the prior distribution parameter λ. Indeed, increasing λ would imply that the factor matrices are expected to be sparser, hence the optimal rank would naturally increase to adapt this sparsity. The results for three typical λ values are given in Fig. 2(c): the predicted ranks become 1, 9, and 33 when λ is set to 0.01, 1.8, and 3, respectively.

We compare our algorithm with the recently proposed large-scale rank estimation algorithm in PARAFAC models, called efficient core consistency diagnostics (CONCORDIA) [12]. As reported in [12], when applied to the Facebook dataset, the CONCORDIA algorithm produces similar results to the ones obtained via our method with $\lambda = 1.8$. The key advantage of the proposed method over CONCORDIA appears in the computation time and the memory requirements. As CONCORDIA is based on expensive SVD computations, the implementation provided in [12] runs out of memory in our experimental setup, even when $R = 3$. On the other hand, we observe that STI-PSGLD still requires less computation time even if we compare it with the results reported in [12], in which a much more powerful computer (with 1 TB of memory) is considered. The total time consumed by STI-PSGLD for this experiment is 30% less than the time consumed by CORCONDIA. Besides, our computational cost can be made even lower if we further increase B. A computation time analysis of STI-PSGLD is provided in the supplementary document.

VI. Conclusion

We developed a novel Bayesian model selection technique for rank estimation in large-scale PARAFAC models. While having a clear mathematical interpretation, the proposed method also has significantly lower computational needs when compared to existing approaches. We provided an upper-bound for the bias induced by the proposed approach. Our experiments showed that our method is able to find the optimal model order in large-scale problems, while being significantly faster than the state-of-the-art.

REFERENCES

[1] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[2] N. D. Sidiropoulos, "Low-rank decomposition of multi-way arrays: A signal processing perspective," in *Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2004*. IEEE, 2004, pp. 52–58.

[3] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth parafac decomposition for tensor completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5423–5436, 2016.

[4] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 792–799.

[5] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2016.

[6] M. Bieroza, A. Baker, and J. Bridgeman, "New data mining and calibration approaches to the assessment of water treatment efficiency," *Advances in Engineering Software*, vol. 44, no. 1, pp. 126–135, 2012.

[7] S. K. Schmitz, P. P. Hasselbach, B. Ebisch, A. Klein, G. Pipa, and R. A. Galuske, "Application of parallel factor analysis (parafac) to electrophysiological data," *Frontiers in neuroinformatics*, vol. 8, p. 84, 2015.

[8] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of eeg signals: a brief review," *Journal of neuroscience methods*, vol. 248, pp. 59–69, 2015.

[9] R. Bro, "Parafac. tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[10] R. Bro and H. A. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *Journal of chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.

[11] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind parafac receivers for ds-cdma systems," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 810–823, 2000.

[12] E. E. Papalexakis and C. Faloutsos, "Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5441–5445.

[13] F. Roemer and M. Haardt, "A closed-form solution for parallel factor (PARAFAC) analysis," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 2365–2368.

[14] J. P. C. L. da Costa, F. Roemer, M. Haardt, and R. T. de Sousa, "Multi-dimensional model order selection," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 26, 2011.

[15] S. Pouryazdian, S. Beheshti, and S. Krishnan, "CANDECOMP/PARAFAC model order selection based on reconstruction error in the presence of kronecker structured colored noise," *Digital Signal Processing*, vol. 48, pp. 12–26, 2016.

[16] K. Liu, J. P. C. da Costa, H. C. So, L. Huang, and J. Ye, "Detection of number of components in CANDECOMP/PARAFAC models via minimum description length," *Digital Signal Processing*, vol. 51, pp. 110–123, 2016.

[17] U. Şimşekli, R. Badeau, G. Richard, and A. T. Cemgil, "Stochastic thermodynamic integration: efficient Bayesian model selection via stochastic gradient MCMC," in *ICASSP*. IEEE, 2016, pp. 2574–2578.

[18] A. Gelman and X. L. Meng, "Simulating normalizing constants: from importance sampling to bridge sampling to path sampling," *Statist. Sci.*, vol. 13, no. 2, pp. 163–185, 05 1998.

[19] M. Mørup, L. K. Hansen, and S. M. Arnfred, "Algorithms for sparse nonnegative Tucker decompositions," *Neural computation*, vol. 20, no. 8, pp. 2112–2131, 2008.

[20] K. Y. Yılmaz, A. T. Cemgil, and U. Şimşekli, "Generalised coupled tensor factorisation," in *Advances in neural information processing systems*, 2011, pp. 2151–2159.

[21] U. Şimşekli, "Tensor fusion: Learning in heterogeneous and distributed data," Ph.D. dissertation, Boğaziçi University, 2015.

[22] S. Patterson and Y. W. Teh, "Stochastic gradient Riemannian Langevin dynamics on the probability simplex," in *Advances in Neural Information Processing Systems*, Dec. 2013.

[23] U. Şimşekli, H. Koptagel, H. Güldaş, A. T. Cemgil, F. Öztoprak, and Ş. İ. Birbil, "Parallel stochastic gradient Markov Chain Monte Carlo for matrix factorisation models," *arXiv preprint arXiv:1506.01418*, 2015.

[24] U. Şimşekli, A. Durmus, R. Badeau, G. Richard, E. Moulines, and A. T. Cemgil, "Parallelized stochastic gradient Markov Chain Monte Carlo algorithms for non-negative matrix factorization," in *ICASSP*, 2017.

[25] A. Beutel, P. P. Talukdar, A. Kumar, C. Faloutsos, E. E. Papalexakis, and E. P. Xing, "Flexifact: Scalable flexible factorization of coupled tensors on Hadoop," in *ICDM*. SIAM, 2014, pp. 109–117.

[26] C. Li, C. Chen, D. E. Carlson, and L. Carin, "Preconditioned stochastic Gradient Langevin Dynamics for Deep Neural Networks." in *AAAI*, vol. 2, no. 3, 2016, p. 4.

[27] U. Şimşekli, R. Badeau, A. T. Cemgil, and G. Richard, "Stochastic quasi-Newton Langevin Monte Carlo," in *ICML*, 2016.

[28] U. Şimşekli and A. T. Cemgil, "Markov chain Monte Carlo inference for probabilistic latent tensor factorization," in *2012 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, Sep. 2012, pp. 1–6.

[29] S. Chib, "Marginal likelihood from the Gibbs output," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1313–1321, 1995.

[30] "Facebook dataset," http://socialnetworks.mpi-sws.org/data-wosn2009.html.