



Efficient Bayesian Model Selection in PARAFAC via Stochastic Thermodynamic Integration

Thanh Huy Nguyen, Umut Şimşekli, Gael Richard, Ali Taylan Cemgil

► To cite this version:

Thanh Huy Nguyen, Umut Şimşekli, Gael Richard, Ali Taylan Cemgil. Efficient Bayesian Model Selection in PARAFAC via Stochastic Thermodynamic Integration. IEEE Signal Processing Letters, 2018. hal-01779074

HAL Id: hal-01779074

<https://hal.science/hal-01779074>

Submitted on 26 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 tensor factorization models. 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 some heuristics, our method has a clear mathematical interpretation, and has significantly lower computational requirements, thanks to data subsampling 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 and has 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]. Its 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:

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

In this model, the observed tensor \mathbf{X} is approximated as a sum of R different ‘rank-one’ tensors (1), where we define a rank-one tensor as the outer product 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 have been several attempts to address it (see for example [10], [12-16]). The common theme in these approaches is that they use a matrix/tensor decomposition technique, such as the singular value decomposition (SVD), higher-order SVD, and matrix diagonalization, which is then combined with some heuristics. Even though these methods have proven useful in some applications, they often have at least one of the two following 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]. We then propose a novel parallel and distributed variant of STI by exploiting the multi-linear structure of the PARAFAC models, so that the computational complexity of the resulting algorithm can be reduced by a dramatic factor. We further extend this approach and improve its convergence speed 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 some regularity conditions. We illustrate the proposed methods on both synthetic and real datasets. Our results show that the proposed algorithms can successfully estimate the rank of PARAFAC models with a low computational budget, even in large-scale distributed settings.

II. PRELIMINARIES

A. Probabilistic formulation

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

$$\begin{aligned} p(\mathbf{A}) &= \prod_{i,r} p(a_{ir}), \quad p(\mathbf{B}) = \prod_{j,r} p(b_{jr}), \quad p(\mathbf{C}) = \prod_{k,r} p(c_{kr}) \\ p(\mathbf{X}|\mathbf{A}, \mathbf{B}, \mathbf{C}) &= \prod_{i,j,k} p(x_{ijk}|\mathbf{A}_i, \mathbf{B}_j, \mathbf{C}_k) \end{aligned} \quad (2)$$

where $p(a_{ir})$, $p(b_{jr})$, and $p(c_{kr})$ are the prior distributions, $p(x_{ijk}|\cdot)$ is 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 such a formulation corresponds to a *maximum a-posteriori* estimation in the probabilistic model defined in (2).

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org. 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. The work was supported by the French National Research Agency grant ANR-16-CE23-0014 (FBIMATRIX).

Manuscript received December 21, 2017.

B. Stochastic Thermodynamic Integration

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

$$p(x|m) = \int p(x|\theta, m)p(\theta|m)d\theta, \quad (3)$$

where $x = \{x_n\}_{n=1}^N$ denotes a set of independent and identically distributed 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^* that maximizes the marginal likelihood: $m^* = \arg \max_m \int p(x|\theta, m)p(\theta|m)d\theta$. Unfortunately, computing m^* turns out to be intractable except for simple models, motivating the need for approximate methods.

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

$$\log p(x) = \int_0^1 \langle \log p(x|\theta) \rangle_{p(\theta|t)} dt \quad (4)$$

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 $p(\theta|x)$, and is 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: $\log p(x) \approx$

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

where $0 = t_0 < t_1 < \dots < t_T = 1$ 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) [19], which iteratively applies the following update equation for generating samples from the distribution $p(\theta|t)$:

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

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

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

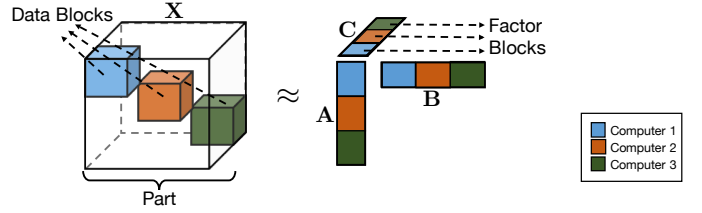


Fig. 1: Illustration of a data part (left) and the factor blocks (right). Each color represents a different computational unit and the blocks having the same color reside in the same unit.

sense, this algorithm is identical to the well-known stochastic gradient descent (SGD) algorithm, 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:

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

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 simple algorithm that is suitable for large-scale problems.

III. PARALLEL AND DISTRIBUTED STI FOR PARAFAC

In this section, we will customize STI for the rank estimation problem in PARAFAC. We first represent the PARAFAC model defined in (2) within the notation introduced in (3) by setting $x \equiv \text{vec}(\mathbf{X}) \in \mathbb{R}^{IJK}$, a vector containing all the observations, and $\theta \equiv [\text{vec}(\mathbf{A}), \text{vec}(\mathbf{B}), \text{vec}(\mathbf{C})] \in \mathbb{R}^{IR+JR+KR}$, 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 , the log-marginal likelihood for this rank $\log p(\mathbf{X}|R)$ can be approximated by using (7) and (5).

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

Our approach is inspired by the distributed SGD algorithm for PARAFAC, which was proposed in [20]. 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 illustration of such a partitioning scheme is shown in Fig. 1. Here, the observed tensor \mathbf{X} is partitioned into $3 \times 3 \times 3$ 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), in such a way that these blocks will not intersect

in any dimension. In this scheme, the blue (orange, green, respectively) data block is used only for the computation of the gradients with respect to the blue (orange, green, respectively) blocks of the three latent matrices. Therefore, we can update the blue, orange, and green blocks at the same time, i.e. in parallel, without any conflicts.

We call the combination of such blocks a ‘part’. In our example, we have 3×3 disjoint parts, which indicates that we need 3^2 iterations to cover all the data points in \mathbf{X} . At the end of each iteration, some of the factor blocks 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.

Remark. In some applications, the elements in \mathbf{X} , \mathbf{A} , \mathbf{B} , and \mathbf{C} are required to be non-negative; resulting in a *non-negative* decomposition [21–23]. In such cases, SGLD will not be applicable since it might result in samples with negative entries due to the additive update rules. If a non-negative PARAFAC problem is considered, by following [24–26], 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.

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 [27], [28]. As a remedy, Li et al. [27] 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 $p(\theta|t)$: $\theta^{(t,l)} = \theta^{(t,l-1)} +$

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

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

$$\mathbf{G}(\theta^{(t,l)}) \triangleq \text{diag} \left(\mathbf{1} \oslash (\sigma \mathbf{1} + \sqrt{\mathbf{v}(\theta^{(t,l)})}) \right), \quad (9)$$

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

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

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., the 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 additional bias [27]. In the sequel, we analyze the overall bias that is induced by the STI algorithm when it is combined with PSGLD.

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

$$\left| \langle \hat{\mathcal{L}} \rangle - \mathcal{L} \right| = \mathcal{O} \left(\frac{1}{L\epsilon} + \max_t \left\{ \frac{1}{L} \sum_{l=1}^L \langle \|\Delta V^{(t,l)}\| \rangle \right\} + \epsilon + \frac{1}{T^2} + \frac{1 - \alpha}{\alpha^{3/2}} \right), \quad (10)$$

where $\Delta V^{(t,l)} = (N\bar{\mathbf{g}}^{(t,l)} - \mathbf{g}^{(t,l)})^{\top} \mathbf{G}(\theta^{(t,l)}) \nabla_{\theta}$ is an operator and $\mathbf{g}^{(t,l)} = t \sum_{n=1}^N \nabla_{\theta} \log p(x_n | \theta^{(t,l)})$ is the full gradient.

The detailed proof and the required conditions are provided in the supplementary document. The difference between this bound and the one of STI with standard SGLD [17] is the presence of the second term and the last term in the right hand side of (10). While the second term is a bounded quantity due to our assumptions on the gradients, the last term can be neglected in practice if α is set to a value close to 1.

Thanks to the multi-linear structure of PARAFAC, PSGLD can also be parallelized by using the same approach described in Sec. III. However, in this case we would need to partition and communicate the preconditioning variable \mathbf{v} as well, 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 composite 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:

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

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 a Message Passing Interface (MPI) library, called Open MPI [29], for parallel computations.

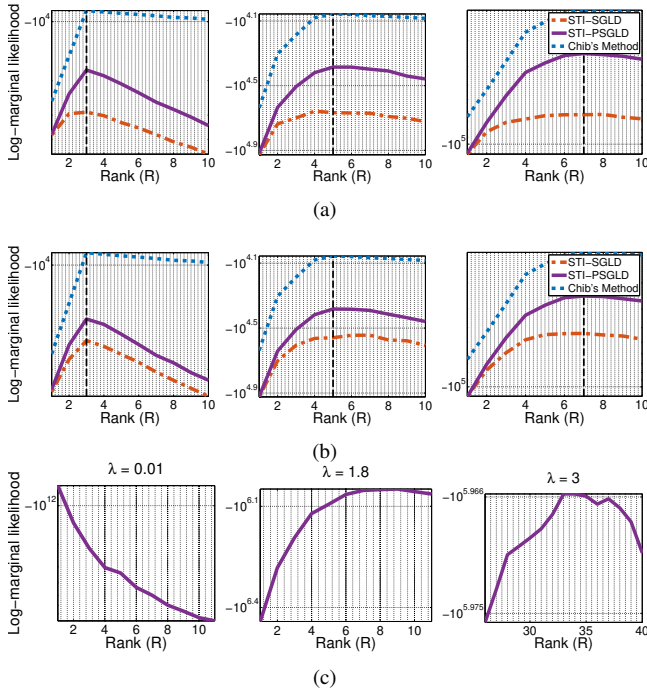


Fig. 2: Simulation results on 2(a) synthetic data with small number of iterations; 2(b) synthetic data with large number of iterations; and 2(c) Facebook dataset.

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 the model (11), then we 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. Hence, we compare the proposed approaches with an existing algorithm, the so called Chib's method [30], [31]. This method is unbiased; however, its complexity rapidly increases with the problem size and can 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 $R \in \{1, \dots, 11\}$ using STI-SGLD, STI-PSGLD, and Chib's method. We set $T = 10$, $N_s = IJK/25$, $\alpha = 0.99$, $\sigma = 10^{-5}$, then we 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 step-size $\epsilon = (a_\epsilon/l)^\epsilon$ with $a_\epsilon = 10^{-8}$, $b_\epsilon = 0.51$ and keep the step-size fixed after burn-in. For Chib's method, we generate 800 samples for each rank.

The results are shown in Fig. 2(a). We observe that 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 expectations. 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 get closer 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 convergence rate.

B. Experiments on real data

In this section, we apply STI-PSGLD to a real large-scale dataset, called the Facebook dataset [32]. This dataset is represented as a three-way tensor of dimensions: $42390 \times 39986 \times 1506$, and 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 $R \in \{1, \dots, 11\}$, we set $T = 5$, $\lambda_a = \lambda_b = \lambda_c = \lambda$, $\alpha = 0.99$, $\sigma = 10^{-5}$, then generate $K = 3000$ samples at each run and use the last 500 samples for estimations. For parallelization, we choose $B = 12$.

It is known that the true marginal likelihood is sensitive to the choice of the prior distribution parameters [33], and in our experiments we observe that the optimal rank changes depending on the prior parameter λ . Indeed, increasing λ implies that the factor matrices are expected to be sparser, hence the optimal rank naturally increases to take this sparsity into account. From this point of view, any Bayesian model selection method estimates the optimal rank, provided the expected level of sparsity λ . 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, respectively².

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 with 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 CONCORDIA. 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 and provided an upper-bound for the induced bias. Our experiments showed that the proposed method is able to find the optimal model order in large-scale problems, while being significantly faster than the state-of-the-art.

²The parameter λ can be estimated [34]; however, it is beyond our scope.

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, article 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, article 84, 2015.
- [8] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristanieni, "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 *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2015, pp. 5441–5445.
- [13] F. Roemer and M. Haardt, "A closed-form solution for parallel factor (PARAFAC) analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.* 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, article 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. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proceedings of the 28th International Conference on Machine Learning*, Jun. 2011, pp. 681–688.
- [20] 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.
- [21] 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.
- [22] K. Y. Yılmaz, A. T. Cemgil, and U. Şimşekli, "Generalised coupled tensor factorisation," in *Advances in neural information processing systems*, 2011, pp. 2151–2159.
- [23] U. Şimşekli, "Tensor Fusion: Learning in Heterogeneous and Distributed Data," Ph.D. dissertation, Computer Engineering, Boğaziçi University, 2015.
- [24] S. Patterson and Y. W. Teh, "Stochastic gradient Riemannian Langevin dynamics on the probability simplex," in *Advances in Neural Information Processing Systems*, pp. 3102–3110.
- [25] U. Şimşekli, H. Koptagel, H. GÜltaş, 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.
- [26] 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, pp. 2242–2246.
- [27] 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.
- [28] U. Şimşekli, R. Badeau, A. T. Cemgil, and G. Richard, "Stochastic quasi-Newton Langevin Monte Carlo," in *ICML*, 2016.
- [29] "Open MPI," <https://www.open-mpi.org/>.
- [30] 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.
- [31] S. Chib, "Marginal likelihood from the Gibbs output," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1313–1321, 1995.
- [32] "Facebook dataset," <http://socialnetworks.mpi-sws.org/data-wosn2009.html>, 2009.
- [33] N. Friel and A. N. Pettitt, "Marginal likelihood estimation via power posteriors," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 3, pp. 589–607, Jul. 2008.
- [34] S. Mandt, M. Hoffman, and D. Blei, "A variational analysis of stochastic gradient algorithms," in *International Conference on Machine Learning*, 2016, pp. 354–363.