Bayes method for low rank tensor estimation

†Taiji Suzuki and ‡Heishiro Kanagawa
†‡Graduate School of Information Science and Engineering, Tokyo Institute of Technology, O-okayama 2-12-1, Meguro-ku, Tokyo 152-8552, JAPAN
†PRESTO, JST
E-mail: †s-taiji@is.titech.ac.jp, ‡kanagawah.ab@m.titech.ac.jp

Abstract. We investigate the statistical convergence rate of a Bayesian low-rank tensor estimator, and construct a Bayesian nonlinear tensor estimator. The problem setting is the regression problem where the regression coefficient forms a tensor structure. This problem setting occurs in many practical applications, such as collaborative filtering, multi-task learning, and spatio-temporal data analysis. The convergence rate of the Bayes tensor estimator is analyzed in terms of both in-sample and out-of-sample predictive accuracies. It is shown that a fast learning rate is achieved without any strong convexity of the observation. Moreover, we extend the tensor estimator to a nonlinear function estimator so that we estimate a function that is a tensor product of several functions.

1. Introduction
Tensor modeling is a powerful tool for representing higher order relations between several data sources. The second order correlation has been a main tool in data analysis for a long time. However, because of an increase in the variety of data types, we frequently encounter a situation where higher order correlations are important for transferring information between more than two data sources. In this situation, a tensor structure is required. For example, a recommendation system with a three-mode table, such as user × movie × context, is regarded as comprising the tensor data analysis [1]. The noteworthy success of tensor data analysis is based on the notion of the low rank property of a tensor, which is analogous to that of a matrix. The rank of a tensor is defined by a generalized version of the singular value decomposition for matrices. This enables us to decompose a tensor into a few factors and find higher order relations between several data sources.

A naive approach to computing tensor decomposition requires non-convex optimization [2]. Several authors have proposed convex relaxation methods to overcome the computational difficulty caused by non-convexity [3–7]. The main idea of convex relaxations is to unfold a tensor into a matrix, and apply trace norm regularization to the matrix thus obtained. This technique connects low rank tensor estimation to the well-investigated convex low rank matrix estimation. Thus, we can apply the techniques developed in low rank matrix estimation in terms of optimization and statistical theories. To address the theoretical aspects, [6] gave the statistical convergence rate of a convex tensor estimator that utilizes the so-called overlapped Schatten 1-norm defined by the sum of the trace norms of all unfolded matricizations. [8] showed that the bound given by [6] is tight, but can be improved by a modified technique called square deal. [7] proposed another approach called latent Schatten 1-norm regularization that is defined by the infimum convolution of trace norms of all unfolded matricizations, and analyzed its convergence rate. These theoretical studies revealed the qualitative dependence of learning rates on the rank and size of the
underlying tensor. However, one problem of convex methods is that, reducing the problem to a matrix estimation, one may lose statistical efficiency in exchange for computational efficiency.

One important question is whether we can construct a tractable estimator that possesses a (near) optimal learning rate for tensor estimation. The Bayes method is a promising way to satisfy this requirement [9–13]. In particular, [13] showed that a Bayes estimator can achieve the near-minimax optimal rate for the tensor estimation problem. Roughly speaking, it is shown that the mean squared error of the estimator \( \hat{A} \) is evaluated as

\[
\| \hat{A} - A^* \|_n^2 = O_p\left( \frac{d^*(\sum_{k=1}^K M_k) \log(K \sqrt{n(\sum_k M_k)^K})}{n} \right),
\]

where \( n \) is the sample size, \( \hat{A} \) is the Bayes estimator, \( A^* \) is the true tensor, \( d^* \) is the CP-rank of the true tensor (its definition will be given in Section 2), and \( (M_1, \ldots, M_K) \) is the size of \( A^* \in \mathbb{R}^{M_1 \times \cdots \times M_K} \). A remarkable point of the analysis of [13] is that the rate is proven without assuming any strong convexity on the empirical \( L_2 \) norm. A kind of restricted strong convexity [14, 15] has been usually assumed in the existing regularization approaches. However, for the analysis of the predictive accuracy of a Bayes estimator, we don’t need that assumption.

Another issue we deal with in this article is the nonlinear extension of the tensor model. All the works introduced above are about a linear model where the function we want to estimate is a linear function having a tensor form. This model can be extended to a nonlinear model [16]. So far, an alternating minimization with an RKHS regularization has been investigated numerically [16]. However, its theoretical properties are not well investigated. In this article, we present a Gaussian process approach to estimate the nonlinear tensor model. The Gaussian process approach is derived as a natural extension of the linear one. We will show its effectiveness in numerical experiments.

2. Problem Settings

In this section, the problem setting of this article is shown. Suppose that there exists the true tensor \( A^* \in \mathbb{R}^{M_1 \times \cdots \times M_K} \) of order \( K \), and we observe \( n \) samples \( D_n = \{(Y_i, X_i)\}_{i=1}^n \) from the following linear model:

\[
Y_i = \langle A^*, X_i \rangle + \epsilon_i.
\]

Here, \( X_i \) is a tensor in \( \mathbb{R}^{M_1 \times \cdots \times M_K} \) and the inner product \( \langle \cdot, \cdot \rangle \) between two tensors \( A, X \in \mathbb{R}^{M_1 \times \cdots \times M_K} \) is defined by \( \langle A, X \rangle = \sum_{j_1, \ldots, j_K=1}^{M_1, \ldots, M_K} A_{j_1, \ldots, j_K} X_{j_1, \ldots, j_K} \). \( \epsilon_i \) is i.i.d. noise from a normal distribution \( \mathcal{N}(0, \sigma^2) \) with mean 0 and variance \( \sigma^2 \).

Now, we assume the true tensor \( A^* \) is “low-rank.” The notion of rank considered in this article is CP-rank (Canonical Polyadic rank) [17, 18]. We say a tensor \( A \in \mathbb{R}^{M_1 \times \cdots \times M_K} \) has CP-rank \( d' \) if there exist matrices \( U^{(k)} \in \mathbb{R}^{d' \times M_k} \) \( (k = 1, \ldots, K) \) such that \( A_{j_1, \ldots, j_K} = \sum_{r=1}^{d'} U^{(1)}_{r,j_1} U^{(2)}_{r,j_2} \cdots U^{(K)}_{r,j_K} \), and \( d' \) is the minimum number to yield this decomposition (we do not require the orthogonality of \( U^{(k)} \)). This is called CP-decomposition. When \( A \) satisfies this relation for \( U = (U^{(1)}, U^{(2)}, \ldots, U^{(K)}) \), we write

\[
A = AU = \left[ (U^{(1)}, U^{(2)}, \ldots, U^{(K)}) \right] := \left( \sum_{r=1}^{d'} U^{(1)}_{r,j_1} U^{(2)}_{r,j_2} \cdots U^{(K)}_{r,j_K} \right)_{j_1, \ldots, j_K}.
\]

We denote by \( d^* \) the CP-rank of the true tensor \( A^* \). Notice that, for the special case of matrices \( (K = 2) \), the CP-rank coincides with the usual rank of a matrix.

In this article, we are interested in the predictive accuracy of the linear model with the assumption that \( A^* \) has low CP-rank. Because of the low CP-rank assumption, the learning problem becomes more structured than an ordinary linear regression problem on a vector. This problem setting includes the well-known low rank matrix estimation as a special case \( K = 2 \). There are two types of predictive
accuracy: \textit{in-sample} and \textit{out-of-sample} ones. The in-sample predictive accuracy of an estimator $\hat{A}$ is defined by

$$
\|\hat{A} - A^*\|_2^n := \frac{1}{n} \sum_{i=1}^{n} \langle X_i, \hat{A} - A^* \rangle^2,
$$

(3)

where $\{X_i\}_{i=1}^{n}$ is the observed input samples. The out-of-sample one is defined by

$$
\|\hat{A} - A^*\|_{L_2(P(X))}^2 := \mathbb{E}_{X \sim P(X)}[\langle X, \hat{A} - A^* \rangle^2],
$$

(4)

where $P(X)$ is the distribution of $X$ that generates the observed samples $\{X_i\}_{i=1}^{n}$ and the expectation is taken over independent realization $X$ from the observed ones.

\textbf{Example 1. Tensor completion under random sampling.} Suppose that we have partial observations of a tensor. A tensor completion problem consists of denoising the observational noise and completing the unobserved elements. In this problem, $X_i$ is independently identically distributed from a set $\{\mathbf{e}_{j_1,\ldots,j_K} \mid 1 \leq j_k \leq M_k \ (k = 1, \ldots, K)\}$, where $\mathbf{e}_{j_1,\ldots,j_K}$ is an indicator tensor that has 1 at its $(j_1, \ldots, j_K)$-element and 0 elsewhere, and thus, $Y_i$ is an observation of one element of $A^*$ contaminated with noise $\epsilon_i$.

The out-of-sample accuracy measures how accurately we can recover the underlying tensor $A^*$ from the partial observation. If $X_i$ is uniformly distributed, $\|\hat{A} - A^*\|_{L_2(P(X))} = \frac{1}{\sqrt{M_1 \ldots M_K}} \|\hat{A} - A^*\|_2$, where $\|\cdot\|_2$ is the $\ell_2$-norm obtained by summing the squares of all the elements. If $K = 2$, this problem is reduced to the standard matrix completion problem. In that sense, our problem setting is a wide generalization of the low rank matrix completion problem.

\textbf{Example 2. Multi-task learning.} Suppose that several tasks are aligned across a 2-dimensional space. For each task $(s, t) \in \{1, \ldots, M_1\} \times \{1, \ldots, M_2\}$ (indexed by two numbers), there is a true weight vector $a_{(s,t)}^* \in \mathbb{R}^{M_3}$. The tensor $A^*$ is an array of the weight vectors $a_{(s,t)}^*$ that is, $A^*_{s,t,j} = a_{(s,t),j}^*$. The input vector $X_i$ is a vector of predictor variables for one specific task, say $(s, t)$, and takes a form such that $X_i(s',t';\cdot) = \begin{cases} x_{i,(s',t')} \in \mathbb{R}^{M_3}, & ((s', t') = (s, t)), \\ 0, & (\text{otherwise}) \end{cases}$. By assuming $A^*$ is low-rank in the sense of CP-rank, the problem becomes a multi-task feature learning with a two dimensional structure in the task space [19].

As shown in the examples, the estimation problem of low-rank tensor $A^*$ is a natural extension of low-rank matrix estimation. However, it has a much richer structure than matrix estimation. Thus far, some convex regularized learning problems have been proposed analogously to spectrum regularization on a matrix, and their theoretical analysis has also been provided. However, no method have been proved to be statistically optimal. There is a huge gap between a matrix and higher order array. One reason for this gap is the computational complexity of the convex envelope of CP-rank. It is well known that the trace norm of a matrix is a convex envelope of the matrix rank on a set of matrices with a restricted operator norm [20]. However, as for tensors, computing CP-rank and CP-decomposition themselves is NP-hard [21]. A nonlinear model as an extension of the model (1) will be considered in Section 6.

3. Bayesian tensor estimator of linear model

Here, we provide the prior distribution of the Bayes estimator for the linear tensor model (1) (as for the nonlinear model we present that in Section 6.1). On a decomposition of a rank $d'$ tensor $A = [U^{(1)}, U^{(2)}, \ldots, U^{(K)}] \ (U^{(k)} \in \mathbb{R}^{d' \times M_k})$, we place a Gaussian prior:

$$
\pi(U^{(1)}, \ldots, U^{(K)}|d') \propto \exp\left\{ - \frac{d'}{2\sigma^2} \sum_{k=1}^{K} \text{Tr}[U^{(k)^\top} U^{(k)}] \right\},
$$
where $\sigma_p > 0$. Moreover, we placed a prior distribution on the rank $1 \leq d' \leq d_{\text{max}}$ as
\[
\pi(d') = \frac{1}{N_\xi} \mathcal{C}^{d'(M_1 + \cdots + M_K)},
\]
where $0 < \xi < 1$ is some positive real number, $d_{\text{max}}$ is a sufficiently large number that is supposed to be larger than $d^*$, and $N_\xi$ is the normalizing constant, $N_\xi = \frac{1 - \mathcal{C}^{d_{\text{max}}(M_1 + \cdots + M_K)}}{\xi - \mathcal{C}^{d_0}(M_1 + \cdots + M_K)}$.

Now, since the noise is Gaussian, the likelihood of a tensor $A$ is given by
\[
p(D_n|A) =: p_{n,A} \propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \langle A, X_i \rangle)^2 \right\}.
\]

The posterior distribution is given by
\[
\Pi(A \in C|D_n) = \frac{\sum_{d=1}^{d_{\text{max}}} \int_{A_U \in C} p_{n,A_U} \pi((U^{(k)})_{k=1}^{K}|d) \pi(d) dU^{(1)} \cdots dU^{(K)} \big| D_n)}{\sum_{d=1}^{d_{\text{max}}} \int p_{n,A_U} \pi((U^{(k)})_{k=1}^{K}|d) \pi(d) dU^{(1)} \cdots dU^{(K)}},
\]
where $C \subseteq \mathbb{R}^{M_1 \times \cdots \times M_K}$. It is noteworthy that the posterior distribution of $U^{(k)}$ conditioned by $d$ and $U^{(k')}$ ($k' \neq k$) is a Gaussian distribution, because the prior is conjugate to Gaussian distributions. Therefore, the posterior mean $\int_A f(A)\Pi(dA|D_n)$ of a function $f : \mathbb{R}^{M_1 \times \cdots \times M_K} \to \mathbb{R}$ can be computed by an MCMC method, such as Gibbs sampling (as for the Bayes tensor estimator, see [10–12]). In this article, we consider the posterior mean estimator $\hat{A} = \int A\Pi(dA|D_n)$ which is the Bayes estimator corresponding to the square loss.

4. Convergence rate analysis of linear tensor estimator

In this section, we present the statistical convergence rate of the Bayes estimator of the linear tensor model (1). Before we give the convergence rate, we define some quantities and provide the assumptions.

We define the max-norm of $A^*$ as
\[
||A^*||_{\text{max},2} := \min_{\{U^{(k)}\}} \{ \max_{i,k} \| U^{(k)} \|_{2} \ | \ A^* = \{ [U^{(1)}, \ldots, U^{(K)}] \}, \ U^{(k)} \in \mathbb{R}^{d^* \times M_k} \}.
\]

The $\ell_p$-norm of a tensor $A$ is given by $\|A\|_p := \langle \sum_{j_1,\ldots,j_K} |A_{j_1,\ldots,j_K}|^p \rangle^{\frac{1}{p}}$. The prior mass around the true tensor $A^*$ is a key quantity for characterizing the convergence rate, which is denoted by $\Xi$:
\[
\Xi(\delta) := - \log(\Pi(A : \|A - A^*\|_n < \delta)),
\]
where $\delta > 0$. Small $\Xi(\delta)$ means that the prior is well concentrated around the truth. Thus, it is natural to consider that, if $\Xi$ is large, the Bayes estimator could be close to the truth. However, clearly we do not know beforehand the location of the truth. Thus, it is not beneficial to place too much prior mass around one specific point. Instead, the prior mass should cover a wide range of possibilities of $A^*$. The balance between concentration and dispersion has a similar meaning to that of the bias-variance trade-off.

To normalize the scale, we assume that the $\ell_1$-norm of $X_i$ is bounded\(^1\).

**Assumption 1.** We assume that the $\ell_1$-norm of $X_i$ is bounded by $1$: $\|X_i\|_1 \leq 1, \ a.s...$

Finally, we define the technical quantities $C_{n,K} := 3K\sqrt{n} \left( \frac{4\sigma^2 \Xi(\frac{1}{\sqrt{n}})}{d^*} \right)^{\frac{K}{2}}$ and $c_\xi := \min\{ |\log(\xi)| / \log(C_{n,K}), 1 \}/4$.

\(^1\) $\ell_1$-norm could be replaced by another norm such as $\ell_2$. This difference affects the analysis of out-of-sample accuracies, but rejecting samples with $\max_X |\langle X, A \rangle| \leq R$ gives an analogous result for other norms.
4.1. In-sample predictive accuracy

We now give the convergence rate of the in-sample predictive accuracy. We suppose the inputs \( \{X_i\}_{i=1}^n \) are fixed (not random). The in-sample predictive accuracy conditioned by \( \{X_i\}_{i=1}^n \) is given as follows.

**Theorem 1.** Under Assumption 1, there exists a universal constant \( C \) such that the posterior mean of the in-sample accuracy is upper bounded by

\[
E \left[ \int \| A - A^* \|_n^2 d\Pi(A|Y_{1:n}) \right] \leq \frac{C}{n} \left( d^* \left( \sum_k M_k + \frac{1}{\log(\xi)} \right) \log(C_{n,K}) + \frac{\Xi(\sqrt{\xi}/n)}{c_\xi} + \log(d_{\max}) + K + 8^K (K + 1) \right) . \tag{5}
\]

The proof is given by [13]. This theorem provides the speed at which the posterior mass concentrates around the true \( A^* \). It should be noted that the integral in the LHS of Eq. (5) is taken outside \( \| A - A^* \|_n^2 \). This gives not only information about the posterior concentration but also the convergence rate of the posterior mean estimator. This can be shown as follows. By Jensen’s inequality, we have

\[
E \left[ \int \| A - A^* \|_n^2 d\Pi(A|Y_{1:n}) \right] \leq E \left[ \int \| A - A^* \|_n^2 d\Pi(A|Y_{1:n}) \right] .
\]

Therefore, Theorem 1 gives a much stronger claim on the posterior than just stating the convergence rate of the posterior mean estimator.

Since the rate (5) is rather complicated, we give a simplified bound. By assuming \( \log(d_{\max}) \) and \( K! \) are smaller than \( d^* (\sum_k M_k) \), we rearrange it as

\[
E \left[ \int \| A - A^* \|_n^2 d\Pi(A|Y_{1:n}) \right] = O \left( \frac{d^*(M_1 + \cdots + M_K)}{n} \log \left( K \sqrt{n(\sum_{k=1}^K M_k)^2} \frac{8^K}{\xi} \right) \right) .
\]

Inside the \( O(\cdot) \) symbol, a constant factor depending on \( K, \| A^* \|_{\max,2}, \sigma_p, \xi \) is hidden. This bound means that the convergence rate is characterized by the actual degree of freedom up to a log term. That is, since the true tensor has rank \( d^* \) and thus has a decomposition (2), the number of unknown parameters is bounded by \( d^*(M_1 + \cdots + M_K) \). Thus, the rate is basically \( O(\text{degree of freedom}) \) (up to log order), which is optimal (see Section 5 for more precise argument). Here, we would like to emphasize that the true rank \( d^* \) is unknown, but by placing a prior distribution on a rank the Bayes estimator can appropriately estimate the rank and gives an almost optimal rate depending on the true rank. In this sense, the Bayes estimator has adaptivity to the true rank.

More importantly, we do not assume any strong convexity on the design. Usually, to derive a fast convergence rate of sparse estimators, such as Lasso and the trace norm regularization estimator, we assume a variant of strong convexity, such as a restricted eigenvalue condition [14] and restricted strong convexity [15]. It is difficult to check the strong convexity condition in practice. However, our convergence rate does not require such conditions. One reason why this is possible is that we are interested in the predictive accuracies rather than the actual distance between the tensors \( \| A - A^* \|_2^2 \) (parameter estimation accuracy). It is known that this phenomenon occurs also in high dimensional regression of vectors, see [22] for example.

4.2. Out-of-sample predictive accuracy

Next, we turn to the convergence rate of the out-of-sample predictive accuracy. In this setting, the input sequence \( \{X_i\}_{i=1}^n \) is not fixed, but an i.i.d. random variable generated by a distribution \( P(X) \).

To obtain fast convergence of the out-of-sample accuracy, we need to bound the difference between the empirical and population \( L_2 \)-errors: \( \| A - A^* \|_n^2 - \| A - A^* \|_{L_2(P(X))}^2 \). To ensure that this quantity is small using Bernstein’s inequality, the sup-norm \( \max_X \|X, A\| \) should be bounded. However, the sup-norm of the posterior mean could be large in tensor estimation. This difficulty can be avoided by rejecting posterior sample \( A \) with a large sup-norm, say \( R \). That is, we accept only a sample \( A_U \) that satisfies \( U \in \{ (U^{(1)}, \ldots, U^{(K)}) \mid \|U^{(k)}_{:,j}\| \leq R (1 \leq k \leq K, 1 \leq j \leq M_k) \} =: U_R \). If \( \| A \|_{\max,2} \leq R \),
then under Assumption 1, we have $|⟨X, A⟩| \leq R^K$. Therefore, by restricting the max-norm of the posterior sample, we can control the sup-norm. The estimation with this posterior can be implemented merely by rejecting the posterior samples with a sup-norm larger than a threshold $R$ during the sampling scheme.

Here, we assume that the max-norm $∥A^∗∥_{max,2}$ of the true tensor is approximately known, that is, we know $R > 0$ such that $2∥A^∗∥_{max,2} < R$. Otherwise, we may apply cross validation. We employ this $R$ for the posterior rejection threshold.

The resultant posterior distribution is expressed as the conditional posterior distribution $Π(⟨|∥X, A⟩| ≤ R, D_n)$.

**Theorem 2.** Under Assumption 1 and $∥A^∗∥_{max,2} + σ_p < R$, we have

$$E_{D_n} \left[ ∫ ∥A_U − A^∗∥_{L_2(P(X))}^2 dΠ(A_U | U ∈ U_R, D_n) \right] \leq C \left( \frac{d^4(\sum_{k=1}^K M_k)}{n} (1 ∨ R^{2K}) \log \left( K \sqrt{n} R^2 \frac{σ_p}{ξ} \right) \right),$$

where $C$ is a constant depending on $K, \log(d_{max}), σ_p$.

It should be noted that the optimal rate is achieved, although we do not assume any strong convexity on the distribution $L_2(Π)$. This can be so because we are not analyzing the actual $L_2$-norm $∥A − A^∗∥_2$. If we do not assume strong convexity like $∥A − A^∗∥_2 ≤ C∥A − A^∗∥_{L_2(P(X))}$, it is impossible to derive fast convergence of $∥A − A^∗∥_2$. The trick is that we focus on the “weighted” $L_2$-norm $∥A − A^∗∥_{L_2(P(X))}$ instead of $∥A − A^∗∥_2$.

Finally, it is remarked that, if $X_1$ is the uniform at random observation in the tensor completion problem, then $∥A − A^∗∥_{L_2(P(X))}^2 = \prod_{k=1}^K M_k ∥A − A^∗∥_2^2$ (note that in this setting $∥X_i∥_1 = 1$). Thus, our analysis yields fast convergence of the tensor recovery. If $K = 2$, the analysis recovers the well known rate of matrix completion problems up to a $\log(nM_1M_2)$ term [15, 23, 24]:

$$\frac{1}{M_1M_2} ∥\hat{A} − A^∗∥_2^2 = O_p \left( \frac{d^4(M_1 + M_2)}{n} \log(nM_1M_2) \right).$$

5. Minimax optimality

In this section, the learning rates derived above are actually minimax optimal up to log terms. To prove this, we specify the $L_2(P(X))$ norm. We take $L_2(P(X))$ as a uniform observation of the entries. That is, $⟨X, A⟩ = A_{i_1, i_2, \ldots, i_K}$ for some $i_k ∈ \{1, \ldots, M_k\}$ ($k = 1, \ldots, K$), and the choice of $(i_1, i_2, \ldots, i_K)$ is uniform. The hypothesis space is given by

$$T_R = \{[[U^{(1)}, \ldots, U^{(K)}]] | U^{(k)} ∈ \mathbb{R}^{d^k×M_k}, ∥U^{(k)}∥_∞ ≤ R (1 ≤ k ≤ K, 1 ≤ j ≤ M_k)\},$$

the set of tensors with rank $d^*$ and the max norm not more than $R$. The minimax optimal rate is the convergence rate that can not be improved by any estimator: for any estimator, there is a tensor $A^∗ ∈ T_R$ such that the predictive accuracy corresponding to the true tensor $A^∗$ is lower bounded by the minimax optimal risk.

**Theorem 3.** The minimax learning rate of the tensor estimation is lower bounded as follows. Suppose that $R ≥ 1$, $M_k > 4 (∀k < K)$, $M_K/d^* > 4$ and $M_K/d^*$ is integer. Then there exists a constant $C$ such that

$$\inf_{\hat{A}} \sup_{A^∗ \in T_R} E[∥\hat{A} − A^∗∥_{L_2(P(X))}] ≥ C \min \left\{ σ^2 \left( \frac{d^4(\sum_{k=1}^K M_k)}{n} \right), (R^2/d^*)^{K} \right\},$$

where $\inf_{\hat{A}}$ is taken over all estimator and the expectation is taken for the training samples.
The proof is given by [13]. The theorem is proven by using the information theoretic argument developed by [25] to derive the minimax optimal rate. The assumption, \( M_K/d^* \) is integer, is just a technical assumption and is not essential. We can see that the convergence rates given in Theorems 2 are minimax optimal up to \( \log R^{2K} \) terms.

6. Extension to nonlinear tensor estimator

So far, we have considered a linear relation. However, the model can be easily extended to capture nonlinear relations.

We consider the input \( X_i \) as a concatenation of \( K \) variables, i.e., \( X_i = (x_i^{(1)}, \ldots, x_i^{(K)}) \) where each \( x_i^{(k)} \) is an element of a set \( \mathcal{X}_k \). Here, we don’t assume that \( x^{(k)} \) is a real vector. That can be of any data type such as a graph as long as we can define a positive definite kernel function on it. We consider the regression problem where these samples are generated according to the nonparametric model [16]:

\[
Y_i = \sum_{r=1}^{d^*} \prod_{k=1}^{K} f_r^{*(k)}(x_i^{(k)}) + \epsilon_i, \tag{6}
\]

where \( \{\epsilon_i\}_{i=1}^{n} \) are i.i.d. zero-mean random variables. In this regression problem, our objective is to estimate the true function \( f^*(x^{(1)}, \ldots, x^{(K)}) = \sum_{r=1}^{d^*} \prod_{k=1}^{K} f_r^{*(k)}(x_i^{(k)}) \) from the data \( D_n = \{(X_i, Y_i)\}_{i=1}^{n} \).

This model captures the effect of non-linear higher order interactions among the input components \( \{x^{(k)}\}_{k=1}^{K} \) to the output \( y \), thus is useful for a regression problem where the output is determined by complex relations between the input components. That kind of regression problem appears in several applications such as multi-task learning, recommendation system, spatiotemporal data analysis [19, 2, 7].

To understand the model (6), it is helpful to consider the linear model (1) as a special case. If we assume that \( X_i \) is rank-1, i.e., \( X_i = x_i^{(1)} \odot \cdots \odot x_i^{(K)} \), then the inner product between the tensors \( X_i \) and \( A^* \) is written as:

\[
\langle A^*, X_i \rangle = \sum_{r=1}^{d^*} \prod_{k=1}^{K} \langle U_{r,i}^{*(k)}, x_i^{(k)} \rangle.
\]

This is equivalent to the case where we limit \( f_r^{*(k)} \) in Eq. (6) to the linear function \( \langle U_{r,i}^{*(k)}, x_i^{(k)} \rangle \). Hence, the linear model based on CP-decomposition can be understood as a special case of the nonlinear model (6).

6.1. The prior and posterior distributions for the nonlinear tensor estimation

We put a Gaussian process prior \( \text{GP}_{r,k} \) on each component \( f_r^{*(k)} \). A (zero-mean) Gaussian process \( f = (f(x) : x \in \mathcal{X}) \) on some input space \( \mathcal{X} \) is a set of random variables \( f(x) \) indexed by \( \mathcal{X} \) and defined on a common probability space \( (\Omega, \mathcal{F}, P) \) such that each finite subset \( (f(x_1), \ldots, f(x_j)) (j = 1, 2, \ldots) \) obeys a zero-mean multivariate normal distribution. Corresponding to the Gaussian process, we can define the kernel function \( k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \) as the covariance function defined by \( k(x, x') := \mathbb{E}[f(x)f(x')] \). We denote by \( k_{r,k} \) the kernel function corresponding to the Gaussian process \( \text{GP}_{r,k} \).

The number \( d \) of components (in other word, rank) can be set by cross validation. Otherwise, we put a prior distribution on the rank \( 1 \leq d \leq d_{\text{max}} \) as

\[
\pi(d) = \frac{\xi^d}{\sum_{d'=1}^{d_{\text{max}}} \xi^{d'}},
\]

where \( 0 < \xi < 1 \) is some positive real number and \( d_{\text{max}} \) is taken so that it is sufficiently larger than the supposed true rank \( d^* \). Given a rank \( d \), let \( F = (f_r^{(k)})_{r=1, \ldots, d, k=1, \ldots, K} \) be a concatenation of
functions \( \{f_r^{(k)}\}_{r=1,\ldots,d, k=1,\ldots,K} \) and we consider the following prior distribution on the product space \( d\mathcal{F} = (df_r^{(k)})_{r=1,\ldots,d, k=1,\ldots,K} \):
\[
\Pi(d\mathcal{F}|d) = \Pi_{r=1}^d \Pi_{k=1}^K \text{GP}_{r,k}(df_r^{(k)}).
\]

Based on the prior distribution introduced above, we provide the posterior distribution and the corresponding Bayesian estimator. The posterior measure is constructed as:
\[
\Pi(d\mathcal{F}|D_n) \propto \sum_{d=1}^{d_{\text{max}}} \sum_{d'=1}^{d_{\text{max}}} \Pi(D_n|\mathcal{F}, d) \Pi(d\mathcal{F}|d') \pi(d),
\]
where \( \Pi(D_n|\mathcal{F}) \) is a quasi likelihood defined by
\[
\Pi(D_n|\mathcal{F}) = \exp \left\{ -\beta \sum_{i=1}^n \left( Y_i - \sum_{r=1}^d \sum_{k=1}^K f_r^{(k)}(x_i^{(k)}) \right)^2 \right\}
\]
with an inverse temperature parameter \( \beta > 0 \). Corresponding to the posterior, we have the posterior mean estimator \( \hat{f} = \int f \Pi(df|D_n) \).

### 6.2. Posterior sampling

Here, we present how to obtain samples from the posterior distribution. Let \( f_r^{(k)} \) be the function values of \( f_r \) on the input data points: \( f_r(x_1^{(k)}), \ldots, f_r(x_n^{(k)}) \in \mathbb{R}^n \). Since the distribution of \( f_r^{(k)} \) for some \((r, k)\) given \( d \) and \( f_{r'}^{(k')} \) for \((r', k') \neq (r, k)\) is a Gaussian distribution, we use Gibbs sampling for the computation of the mean estimator. Let \( K_r^{(k)} \) be the Gram matrix corresponding to the kernel function \( k_{r,k} \): \( K_r^{(k)} = (k_{r,k}(x_i, x_j))_{i=1,j=1}^{n,n} \). The conditional distribution of \( f_r^{(k)} \) is
\[
\pi(f_r^{(k)}|f_{-r}^{(k)}, D_n) \sim N(\mu_r^{(k)}, \Sigma_r^{(k)}),
\]
\[
\mu_r^{(k)} = \Sigma_r^{(k)} (K_r^{(k)} - \text{diag}(a_1^2, \ldots, a_n^2))^{-1} f_r,
\]
\[
\Sigma_r^{(k)} = K_r^{(k)} - \text{diag}(a_1^2, \ldots, a_n^2).
\]

where the symbol \( * \) is the Hadamard product, \( N(\mu, \Sigma) \) is the multivariate normal distribution with mean \( \mu \) and variance-covariance \( \Sigma \), \( f_{-r}^{(k)} = \{f_{r'}^{(k')}(x_i^{(k')})\}_{r' \neq r, k'}_{i=1}^{n} \), \( y = (y_i)_{i=1}^n \), \( a = (\prod_{k' \neq k} f_{r'}^{(k')} (x_i^{(k')})_{i=1}^n \), and \( b = (\sum_{r' \neq r} \prod_{k=1}^K f_{r'}^{(k)}(x_i^{(k)})_{i=1}^n)_{i=1}^n \). Therefore, we can sample the function value \( f_r^{(k)} \) for \((r, k) \in [d] \times [K] \) from the posterior using the Gibbs sampling technique. That is, we sample \( f_r^{(k)} \) recursively one after another. We denote the \( j \)-th sample by \( f_r^{(k)} \) for \((r, k) \in [d] \times [K] \).

Based on the posterior samples generated from the above formula, we can predict the outcome on the input \( x \) through the following distribution:
\[
\pi(f_r(x)|f_r^{(k)}, f_{-r}^{(k)}, D_n) \sim N(\mu, \sigma^2),
\]
\[
\mu = k_{x(k)}^\top (K_r^{(k)})^{-1} f_r,
\]
\[
\sigma^2 = k(x(k), x(k)) - k_{x(k)}^\top (K_r^{(k)})^{-1} k_{x(k)}.
\]
where \( k_{x(k)} := (k(x_1^{(k)}, x_1^{(k)}), \ldots, k(x_n^{(k)}, x_n^{(k)}))^\top \). Along with the posterior sample \( \{f_r^{(k)}\}_{r\in[j]}(r, k) \in [d] \times [K] \), the prediction on the new input point \( x = (x^{(1)}, \ldots, x^{(K)}) \) is given by
\[
\hat{f}(x) = \frac{1}{J} \sum_{j=1}^J \sum_{r=1}^d \sum_{k=1}^K k_{x(k)}^\top (K_r^{(k)})^{-1} f_{r,j}^{(k)}(r, k),
\]
7. Numerical experiments

We now present numerical experiments to show how the Bayes estimator works in practice. The numerical experiments are performed on both linear and nonlinear models.

7.1. Linear tensor estimation

We compare the convex optimization approach and the Bayes method for linear tensor estimation on artificial data. The problem is the tensor completion problem where each observation is a random selection of one element of $A^*$ with observational noise $N(0, 1)$ (see Example 1). The true tensor $A^*$ was randomly generated such that each element of $U^{(k)}$ ($k = 1, \ldots, K$) was uniformly distributed on $[-1, 1]$. $\sigma_p$ was set at 5, and the true tensor was estimated by the posterior mean obtained by the rejection sampling scheme with $R = 10$. $d_{\text{max}}$ and $\xi$ were set at 10 and 0.5. The posterior sampling was terminated after 500 iterations. The experiments were executed in five different settings, called settings 1 to 5: 

$$
\left\{ (M_1, \ldots, M_K), d^* \right\} = \left\{ (10, 10, 10), 4 \right\}, \left\{ (10, 10, 40), 5 \right\}, \left\{ (20, 20, 30), 8 \right\}, \left\{ (20, 30, 40), 5 \right\}, \left\{ (30, 30, 40), 6 \right\}.
$$

For each setting, we repeated the experiments five times and computed the average of the in-sample predictive accuracy and out-of-sample accuracy over all five repetitions. The number of samples was chosen as $n = n_s \prod_k M_k$, where $n_s$ varied from 0.3 to 0.9.

We compare the Bayes estimator with the overlapped Schatten 1-norm regularization approach [6]. The comparison is executed in the settings 2 and 5. As for the regularization parameter of the convex regularized approach, we have chosen the best parameter at each sample size and each problem setting. The dashed lines correspond to the convex approach, and the solid line correspond to the Bayes approach. The accuracies of both methods are improved as the sample size increases. It can be seen that the Bayes approach much outperforms the convex approach in terms of both in-sample and out-of-sample accuracies.

![Figure 1: In-sample and out-of-sample accuracy comparison between the convex regularization approach and the Bayes approach, averaged over five repetitions.](image)

72. Nonlinear tensor estimation

Here, we apply the nonlinear tensor estimator to the multi-task learning problem on the school dataset. The school dataset is from the inner London Education Authority (ILEA) and consists of examination records of 15362 students at 139 secondary schools in years 1985, 1986, and 1987. We modeled this data as a MLMTL problem in which the task is to predict exam scores for students at schools in certain years based on student and school dependent inputs. Following [26] for the preprocessing of the categorical attributes, we obtained 44 features. Therefore, in this setting, $M_1 = 139$ (number of school), $M_2 = 3$ (number of years), and $M_3 = 44$ (number of features).
We put a Gaussian kernel on the 44-dimensional feature space, and put the following kernel on the year and the school (we consider three settings (a), (b) and (c)):

(a) School: $k(p, p') = \delta_{p, p'}$, Year: $k(q, q') = \delta_{q, q'}$,
(b) School: $k(p, p') = \delta_{p, p'} + 0.5 \cdot (1 - \delta_{p, p'})$, Year: $k(q, q') = \delta_{q, q'} + 0.8 \cdot (1 - \delta_{q, q'})$,
(c) School: $k(p, p') = \delta_{p, p'} + 0.8 \cdot (1 - \delta_{p, p'})$, Year: $k(q, q') = \delta_{q, q'} + 0.8 \cdot (1 - \delta_{q, q'})$,

where $p, p'$ are indexes of schools and $q, q'$ are those of years, and $\delta_{i, j}$ is the delta function that is 1 if $i = j$, and 0 otherwise. We fixed $d = 3$.

We compare the Bayes estimator and the regularization method proposed by [27]. The regularization parameter for [27] was chosen from $[100, 900]$ so that it achieves the optimal test MSE. As in [27], we used the percentage of explained variance, $100 \cdot (\text{test MSE})/(\text{variance of } y)$, as the evaluation metric. The result is shown in Figure 3a. We can see the nonlinear Bayes estimator much improves the accuracy.

![Figure 2: Results for the school data set](image)

(a) Comparison between the Bayesian nonparametric method and scaled latent norm regularization [27]  
(b) Comparison between different kernels on the tasks

We also compared different settings of kernels on tasks (Figure 4b). The two settings (b) and (c) that incorporate similarities between tasks performed better than the setting (a).

8. Conclusion and discussions

We presented the statistical convergence rate of a Bayesian low rank tensor estimator. The notion of a tensor’s rank in this article was based on CP-rank. It is noteworthy that the predictive accuracy was derived without any strong convexity assumption. Moreover, we presented the minimax optimal rate of the out-of-sample predictive accuracy. Moreover, we presented a nonparametric method to estimate a nonlinear tensor function using a Gaussian process technique. Numerical experiments showed effectiveness of the Bayes estimators.

Acknowledgment

This work was partially supported by MEXT Kakenhi (25730013, 25120012, 26280009, 15H01678 and 15H05707), JST-PRESTO and JST-CREST.

[1] Karatzoglou A, Amatriain X, Baltrunas L and Oliver N 2010 Proceedings of the 4th ACM Conference on Recommender Systems 2010 pp 79–86
[2] Kolda T G and Bader B W 2009 SIAM Review 51 455–500
[3] Liu J, Musialski P, Wonka P and Ye J 2009 Proceedings of the 12th International Conference on Computer Vision (ICCV) pp 2114–2121
[4] Signoretto M, Lathauwer L D and Suykens J 2010 Nuclear norms for tensors and their use for convex multilinear estimation Tech. Rep. 10-186 ESAT-SISTA K.U.Leuven
[5] Gandy S, Recht B and Yamada I 2011 *Inverse Problems* **27** 025010

[6] Tomioka R, Suzuki T, Hayashi K and Kashima H 2011 *Advances in Neural Information Processing Systems* **24** pp 972–980 nIPS2011

[7] Tomioka R and Suzuki T 2013 *Advances in Neural Information Processing Systems* **26** pp 1331–1339 nIPS2013

[8] Mu C, Huang B, Wright J and Goldfarb D 2014 *Proceedings of the 31th International Conference on Machine Learning* pp 73–81

[9] Chu W and Ghahramani Z 2009 *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)* (JMLR Workshop and Conference Proceedings vol 5)

[10] Xu Z, Yan F and Qi Y 2013 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **99** 1 prePrints

[11] Xiong L, Chen X, Huang T K, Schneider J and Carbonell J G 2010 *Proceedings of SIAM Data Mining* pp 211–222

[12] Bai P, Wang Y, Guo S, Chen G, Dunson D and Carin L 2014 *Proceedings of the 31th International Conference on Machine Learning* (JMLR Workshop and Conference Proceedings vol 32) pp 1800–1808

[13] Suzuki T 2015 *Proceedings of the 32nd International Conference on Machine Learning (ICML2015)* ed Blei D and Bach F (JMLR Workshop and Conference Proceedings) pp 1273–1282

[14] Bickel P J, Ritov Y and Tsybakov A B 2009 *The Annals of Statistics* **37** 1705–1732

[15] Negahban S, Ravikumar P, Wainwright M J and Yu B 2012 *Statistical Science* **27** 538–557

[16] Signoretto M, De Lathauwer L and Suykens J A 2013 Learning tensors in reproducing kernel hilbert spaces with multilinear spectral penalties arXiv:1310.4977

[17] Hitchcock F L 1927 *Journal of Mathematics and Physics* **6** 164–189

[18] Hitchcock F L 1927 *Journal of Mathematics and Physics* **7** 39–79

[19] Romera-Paredes B, Aung H, Bianchi-Berthouze N and Pontil M 2013 *Proceedings of the 30th International Conference on Machine Learning* (JMLR Workshop and Conference Proceedings vol 28) pp 1444–1452

[20] Srebro N, Rennie J and Jaakkola T 2005 *Advances in Neural Information Processing Systems* **17** (MIT Press) pp 1329–1336

[21] Hillar C J and Lim L H 2013 *Journal of the ACM* **60** 45:1–45:39

[22] Dalalyan A S and Tsybakov A B 2008 *Machine Learning* **72** 39–61

[23] Negahban S and Wainwright M J 2012 *Journal of Machine Learning Research* **13** 1665–1697

[24] Rohde A and Tsybakov A B 2011 *The Annals of Statistics* **39** 887–930

[25] Yang Y and Barron A 1999 *The Annals of Statistics* **27** 1564–1599

[26] Bakker B and Heskes T 2003 *Journal of Machine Learning Research* **4** 2003

[27] Wimalawarne K, Sugiyama M and Tomioka R 2014 *Advances in Neural Information Processing Systems* **27** (Curran Associates, Inc.) pp 2825–2833