Structural tensor-on-tensor regression with interaction effects and its application to a hot rolling process

Huihui Miao, Andi Wang, Bing Li & Jianjun Shi

To cite this article: Huihui Miao, Andi Wang, Bing Li & Jianjun Shi (2021): Structural tensor-on-tensor regression with interaction effects and its application to a hot rolling process, Journal of Quality Technology, DOI: 10.1080/00224065.2021.1973931

To link to this article: https://doi.org/10.1080/00224065.2021.1973931

Published online: 30 Sep 2021.
Structural tensor-on-tensor regression with interaction effects and its application to a hot rolling process

Huihui Miao\textsuperscript{a}, Andi Wang\textsuperscript{b}, Bing Li\textsuperscript{a}, and Jianjun Shi\textsuperscript{b}

\textsuperscript{a}State Key Lab for Manufacturing Systems Engineering, Xi’an Jiaotong University, Xi’an, China; \textsuperscript{b}H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia

ABSTRACT

This paper proposes a method of Structural Tensor-On-Tensor regression considering the Interaction effects (STOTI). To alleviate the curse of dimensionality and resolve computational challenge, the STOTI method describes the specific structure of the main and interaction effect tensors indicated by the prior knowledge of the data using corresponding regularization terms on their appropriate modes. We designed an ADMM consensus algorithm to estimate these coefficient tensors. Extensive simulations and a real case study of the hot rolling process verified the superiority of the proposed method in terms of estimation and prediction accuracy.

KEYWORDS

interaction effect; regularization; tensor-on-tensor regression

1. Introduction

A multistage manufacturing system (MMS) is a complex system with multiple variables in each station characterizing the manufacturing process. Tensor data of huge sizes are typically generated as either the process variables or the quality variables are high dimensional in an MMS. For example, in a hot cascade steel rolling process (Figure 1), a billet is heated in the furnace and then passes dozens of stands to form its shape. As one rolling bar passes each stand, multiple sensors record the temperature, current, torque, speed at an equal time interval (Figure 1), generating multiple signals of the same length. Such a specific sensor layout generates a three-mode tensor for each product during its manufacturing process, where these modes denote the measurements along a billet, specific type of sensors, and stand numbers, respectively. After the rolling process, a laser caliper measures the diameters of the rolling bar along six orientations and thus generates a two-mode tensor (i.e., matrix) as the quality output, whose rows correspond to the six directions, while the columns are the measurements along every direction.

Given the high complexity of an MMS, it is typical that both main effects of process variables from each stage and interactions among them relate to the end products’ quality. Process variables at a manufacturing stage may impact other process variables in the current stage and the down-stream stages, and further impact the product quality. For example, when the reheating process results in a high rolling temperature, the rolling speed, and the torque will have more significant influences on the products’ dimension due to the steel billet’s decreased hardness during the rolling process. Therefore, it is crucial to consider the interaction effects when modeling the relationship between the process variables measured from stages and the final products’ quality in the last stage.

In literature, the tensor-on-tensor regression model has been developed for high-dimensional measurements. However, the existing models only take the main effects into consideration while ignoring the interaction effects in the model formulation (Gahrooei et al. 2021). This paper is the first attempt to build a general model of tensor-on-tensor regression with the tensor interaction effects. Specifically, introducing the tensor interaction effects into the tensor-on-tensor regression model leads to two challenges below.

- First, the interaction effects of tensor inputs are represented by a tensor coefficient of a much higher order. For example, consider a three-order input tensor of size $10 \times 8 \times 60$, and a two-order output tensor of size $6 \times 10$. An eight-order tensor of size...
represents the regression interaction effects between these two tensors, which contains over a billion elements. The extra high dimensionality will result in difficulties in both estimation accuracy and computation.

- Second, in tensor regression problems, we usually have some knowledge of the relationship between the input tensor and the output sensor. For example, as shown in the rolling example, we know that only certain process variables affect the shape of the rolling bar, and that the variation patterns of the output driven by the inputs are of low dimension. This knowledge restricts the structure of both the main effect tensor and the interaction effect tensor. How to employ this knowledge in estimating the main effect tensor and the interaction effect tensor to alleviate the curse of dimensionality is another challenge.

The above challenges require us to appropriately incorporate the low dimensional structure in the coefficient tensors according to our understanding of the input-output relationship in the estimation process. As will be seen from the literature review, the existing method of tensor-on-tensor regression is to apply low-order parametric forms for the tensor coefficient. However, they do not fully describe the understanding of the relationship between the input tensor and the output tensor. In our method, we use appropriate regularization terms to represent the sparse and low-rank structure at certain modes of the tensor coefficients to represent both the main and the interaction effect tensor coefficient, and thereby we call our method Structural Tensor-On-Tensor regression with Interactions (STOTI). Similar representations of the structures at different modes of the tensor have been utilized for additive tensor decomposition (Mou et al. 2020). However, this paper uses a similar technique to describe both the main and interactive coefficient tensors that specify the relationship between the tensor input and tensor output and to alleviate the curse of dimensionality. As a result, we can accurately identify the effective main and interaction effects, and achieve higher estimation and prediction accuracy. Our problem is solved by an alternating direction method of
multipliers (ADMM) (Parikh and Boyd 2014), which is highly parallelizable and suitable for high-dimensional problems. The superiority of the STOTI method has been verified by various data scenarios in the simulation study and a real case study.

The remainder of this paper is organized as follows. Section 2 reviews the related works regarding the tensor regression. Section 3 formulates the STOTI model, elaborates the solution and an ADMM algorithm of the model. Section 4 compares the performance of the proposed STOTI method with multiple benchmark methods through a simulation study under different data scenarios. Section 5 further validates the proposed method through a case study from a hot rolling process. Finally, we conclude this article in Section 6.

2. Literature review

The tensor regression analysis is a technique to model the relationships between the tensor predictors and tensor responses. It has broad applications on different types of data, including neuroimaging (Zhou, Li, and Zhu 2013), MRI image (Li et al. 2016), longitudinal data (Hoff 2015), structured point data cloud (Yan, Paynabar, and Pacella 2019), etc.


For representing the interaction effects in the regression model, Lin et al. (2016) introduced high-order feature interaction for multitask learning. Wu, Huang, and Ma (2018) aim at identifying gene-gene interactions among vector inputs. However, no existing literature analyzes the interaction effects under the setup of tensor-on-tensor regression.

To sum up, all existing tensor-on-tensor regression models assume that the coefficient tensor can be described by a small number of parameters, which specifies a low dimensional structure of the input-output relationship. Also, all the above models do not address the interaction effects in the tensor regression model. However, these forms cannot describe the structures of the coefficient tensor according to the sparsity of effective input tensor slices and the output tensor variation patterns suggested in the case of rolling process modeling as described in Section 1. To tackle these problems, we propose to describe the main effect tensor and interaction effect tensors’ unique characteristics by assigning appropriate regularization terms on their different modes and forms of matricization (Mou et al. 2020).

3. Formulation of tensor-on-tensor regression with tensor interactions

In this section, we will first introduce the tensor-on-tensor regression model formulation with interaction effects. Then we present the proposed solution procedure. Finally, we validate the proposed method using a numerical example.

In this paper, we use a lowercase letter like a to represent a scalar, use a boldface lowercase letter like a to represent a vector, and use a boldface uppercase letter like A to represent a matrix. An order-d tensor is denoted by a calligraphic letter like $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$, where $I_j$ is the dimension of its $j$-th mode. $\mathcal{A}(i_1, ..., i_d)$ denotes the element $(i_1, ..., i_d)$ of tensor $\mathcal{A}$. $\mathcal{A}(i_1, ..., i_d ; i_{d+1}, ..., i_k)$ denotes the $(i_{d+1}, ..., i_k)$ slice of a tensor $\mathcal{A}$ at mode $(d_1, ..., d_k)$ by fixing $k$ indices, where $i_d \in \{1, ..., I_d\}$ represents the elements at mode $d_j$, where $j \in \{1, 2, ..., k\}$. The mode-$j$ matricization of $\mathcal{A}$ is a matrix $\mathcal{A}_{i_1, ..., i_d} [q_1, q_2, ..., q_d] \in \mathbb{R}^{I_{d+1} \times I_{d+2} \times \cdots \times I_d}$ where $(q_1, ..., q_d)$ is a permutation of $1, ..., d$. Specifically, the mode-$j$ matricization of tensor $\mathcal{A}^{i_1, ..., i_d}$ is denoted as $\mathcal{A}_{i_d} [q_d] \in \mathbb{R}^{I_{d+1} \times I_d}$, whose columns are the mode-$j$ fibers of the corresponding tensor $\mathcal{A}$, and $I_j = \prod_{i \in \{1, 2, ..., j-1, j+1, ..., d\}} I_i$. Given a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$, we propose to describe the main effect tensor and interaction effect tensors’ unique characteristics by assigning appropriate regularization terms on their different modes and forms of matricization (Mou et al. 2020).
The tensor product between tensor \( A \) and tensor \( B \) is defined as \( \mathcal{F} = A \circ B \in \mathbb{R}^{R \times R} \) where \( \mathcal{F}(r_1, ..., r_t) = \sum_{i_1=1}^{l_1} \cdots \sum_{i_t=1}^{l_t} A(i_1, ..., i_t) B(i_1, ..., i_d) \). As a special case, the mode-\( j \) product of a tensor \( A \) with a matrix \( G \in \mathbb{R}^{P \times l_j} \) is a tensor in \( \mathbb{R}^{l_1 \times \cdots \times l_{j-1} \times P \times l_{j+1} \times \cdots \times l_t} \) defined as \( (A \circ_j G)(i_1, ..., i_{j-1}, p, i_{j+1}, ..., i_t) = \sum_{j=1}^{l_j} A(i_1, ..., i_{j-1}, i_j, i_{j+1}, ..., i_t) G_{p, i_j} \). The outer product between \( A \) and \( B \) gives the tensor \( C = A \otimes B, C \in \mathbb{R}^{l_1 \times \cdots \times l_t} \) where \( \mathcal{C}(i_1, ..., i_t, r_1, ..., r_t) = A(i_1, ..., i_t) \cdot B(r_1, ..., r_t) \). The operator \( T_{u_1, ..., u_d} \) is an extension of matrix transpose in tensor object, and it permutes the modes of a tensor \( X \) such that \( T_{u_1, ..., u_d}(X)(i_1, ..., i_d) = X(i_{u_1}, ..., i_{u_d}) \), where \( u_1, ..., u_d \) is a permutation of \( 1, ..., d \).

### 3.1. Problem description

We introduce our methodology in the context of an MMS. For notational simplicity, we assume that the input tensor has two modes and the output tensor has one mode, and the readers shall see that the setup can be extended to tensors of higher orders. Assume that we have a set of data set with \( N \) samples available. The \( D \)-dimensional quality output is denoted as \( Y_i \in \mathbb{R}^D, i = 1, 2, ..., N \). For each input, we record the measurements of \( M \) process variables at stage \( 1, ..., K \), and its corresponding input is denoted as \( X_i \in \mathbb{R}^{K \times M}, i = 1, 2, ..., N \). The tensor-on-tensor regression model [1] with interaction effects is to associate the output \( Y_i \) not only with the input variables in \( X_i \), but also with the interactions among these input variables. Specifically, we describe the relationship between \( Y_i \) and \( X_i \) through a main effect tensor \( B_1 \in \mathbb{R}^{K \times M \times D} \) and an interaction effect tensor \( B_2 \in \mathbb{R}^{K \times M \times K \times M \times D} \):

\[
Y_i = X_i \ast B_1 + \mathcal{H}_i \ast B_2 + E_i
\]  

[1]

In this equation, \( \mathcal{H}_i = X_i \otimes X_i \in \mathbb{R}^{K \times M \times K \times M} \) represents the interaction of the input tensors and \( E_i \in \mathbb{R}^D \) represents the random noise. To make \( B_2 \) identifiable in the estimation process, we let \( B_2(k_1, m_1, k_2, m_2, d) = B_2(k_2, m_2, k_1, m_1, d), k_1, k_2 = 1, ..., K; m_1, m_2 = 1, ..., M; d = 1, ..., D \).

To put [1] in a compact form, the tensors \( Y_i, X_i, \mathcal{H}_i \), and \( E_i \) corresponding to all samples \( i = 1, ..., N \) can be represented by data tensors \( Y \in \mathbb{R}^{N \times D}, X \in \mathbb{R}^{N \times K \times M}, \mathcal{H} \in \mathbb{R}^{N \times K \times M \times K \times M}, E \in \mathbb{R}^{N \times D} \), respectively, and thereby we have:

\[
Y = X \ast B_1 + \mathcal{H} \ast B_2 + E
\]  

[2]

This model [2] describes both the effects of the process variables themselves and the second-order interaction effects among process variables, through main effect tensor \( B_1 \) and the interaction effect tensor \( B_2 \). In our work, the estimation of \( B_1 \) and \( B_2 \) is achieved by solving an optimization problem that minimizes the sum of the prediction error of the outputs and the regularization terms specifying the structural properties of \( B_1 \) and \( B_2 \). We describe the objective function of this optimization problem in detail below.

#### The loss of the prediction error

The prediction error \( E \) in model [2] for the process outputs can be represented as \( E = Y - X \ast B_1 - \mathcal{H} \ast B_2 \). Then, the prediction loss can be represented as

\[
L(B_1, B_2) = ||Y - X \ast B_1 - \mathcal{H} \ast B_2||_F^2
\]  

[3]

where \( \cdot || \cdot ||_F \) is the Frobenius norm.

#### The regularization for the structural sparsity

One prior knowledge we have on the relationship between the input and output is that among the \( M \) process variables collected from \( K \) stages, only a few of them have impacts on the product quality. Therefore, either the process variables of the same type measured at \( K \) different stages all affect the process outputs, or none of them within the same type affect the process outputs, which means that only some mode-(2) slices of the tensor coefficient \( B_1 \) contain non-zero elements. Similarly, only some inputs of type \( m \) and \( m' \) have interaction effects on the output, \( m, m' \in \{1, ..., M\} \), and thus only some mode-(2, 4) slices of the tensor coefficient \( B_2 \) contain non-zero elements. For the above reasons, we apply the following group lasso (Yuan and Lin 2006) penalties on \( B_1 \) and \( B_2 \), respectively:

\[
p_1(B_1) = \lambda_1 \sum_{m=1}^{M} ||\text{vec}(B_1(:, m, :))||_2 = \lambda_1 ||B_{1[2]}||_{2,1},
\]

\[
p_2(B_2) = \lambda_2 \sum_{m_2=1}^{M} \sum_{m_1=1}^{M} ||\text{vec}(B_2(:, m_1, : m_2, :))||_2 = \lambda_2 ||B_{2[2,4]}||_{2,1},
\]

where \( \lambda_1 \) and \( \lambda_2 \) are the tuning parameters.

#### The regularization for structural variation patterns

Another prior knowledge on the model is related to the output variation patterns of the tensor output. In many applications, the number of the variation patterns of the tensor output driven by the input tensor is limited, because the inputs and outputs are associated with latent factors of low dimension. Therefore, if we reshape the tensors \( B_1 \) and \( B_2 \) into two matrices whose rows and columns correspond to the input and the output, these matrices should be low-rank. This consideration has been applied extensively in multivariate regression, such as partial least square and reduced-rank regression. For the above reason, the mode-(3) matricization of \( B_1 \) and the mode-(5) matricization of \( B_2 \) should be of low
rank. For this reason, we follow the method of Yuan et al. (2007) and apply the penalizations $p_3(B_1) = \lambda_3 \| B_{1(i)} \|_s$ and $p_4(B_2) = \lambda_4 \| B_{2(j)} \|_s$, where $\| \cdot \|_s$ refers to the nuclear norm.

Put all aforementioned terms together, we derive the objective function of the optimization problem as the sum of the prediction error of the outputs $L(B_1, B_2)$ and the four regularization terms of $p_1(B_1)$, $p_2(B_2), p_3(B_1)$, and $p_4(B_2)$. The optimization problem to estimate the tensor coefficients $B_1$ and $B_2$ are formulated as follows:

$$\min_{B_1, B_2} \ L(B_1, B_2) + p_1(B_1) + p_2(B_2) + p_3(B_1) + p_4(B_2)$$  \[4\]

### 3.2. Problem solution

The formulation [4] is a convex problem bounded below by zero. Thus, it has an optimal solution. However, computational efficiency is a critical issue due to the decision variables’ high dimensionality in the objective function. To address this challenge, we adopt an ADMM consensus algorithm to solve it. Similar frameworks were adopted in Wang and Shi (2021) and Mou et al. (2020). To map the formulation [4] into it, three copies of both parameters $B_1$ and $B_2$ are introduced as $\tilde{B}_1 = \left( B_1^{(1)}, B_1^{(2)}, B_1^{(3)} \right)$ and $\tilde{B}_2 = \left( B_2^{(1)}, B_2^{(2)}, B_2^{(3)} \right)$ respectively.

Let $\tilde{B} = \left( B_1^{(1)}, B_1^{(2)}, B_1^{(3)}, B_2^{(1)}, B_2^{(2)}, B_2^{(3)} \right)$, then the formulation [4] is transformed into

$$\min \ f(\tilde{B}) + g(\tilde{B})$$  \[5\]

where $f(\tilde{B}) = L(B_1^{(1)}, B_2^{(1)}) + p_1(B_1^{(2)}) + p_2(B_2^{(2)}) + p_3(B_2^{(3)}) + p_4(B_2^{(3)})$ and $g(\tilde{B}) = I_C(\tilde{B})$. The function $g(\tilde{B}) = I_{B_1^{(1)}=B_2^{(1)}=B_1^{(2)}=B_2^{(2)}=B_1^{(3)}=B_2^{(3)}}(\tilde{B}) \cdot I(B_2^{(3)} \in A; B_2^{(2)} \in A; B_2^{(3)} \in A)$ is to specify the copies of parameters that are of the same values and that all replicates of $B_2$ satisfy the constraint, where $I_a(x) = \begin{cases} 0, & \text{if } x \in \Omega \\ +\infty, & \text{else} \end{cases}$ is an indicator function and $A = \{ B_2 : B_2(k_1, m_1, k_2, m_2, d) = B_2(k_2, m_1, k_1, m_2, d), \forall k_1, m_1, k_2, m_2, d \}$. The formulation [5] can be solved using the general framework of ADMM as presented in Algorithm 1.

---

**Algorithm 1. ADMM Algorithm**

Initialize $\tilde{Z}$ and $\tilde{U}$ as the same structures as $\tilde{B}$. Set all their elements to be 0.

Do:

Step 1. Set $\tilde{B}_{\text{prev}} \leftarrow \tilde{B}$, $\tilde{Z}_{\text{prev}} \leftarrow \tilde{Z}$ and $\tilde{U}_{\text{prev}} \leftarrow \tilde{U}$

Step 2. $\tilde{B} \leftarrow \text{prox}_{g}(\tilde{Z} - \tilde{U})$

Step 3. $\tilde{Z} \leftarrow \text{prox}_{f}(\tilde{B} + \tilde{U})$

Step 4. $\tilde{U} \leftarrow \tilde{U} + \tilde{B} - \tilde{Z}$

Until $\| \tilde{U} - \tilde{U}_{\text{prev}} \| < \epsilon$ and $\| \tilde{Z} - \tilde{Z}_{\text{prev}} \| < \epsilon$

---

In Algorithm 1, the parameter $\eta$ is to specify the step size, and $\text{prox}_{g}(\cdot)$ is the proximal operator defined as

$$\text{prox}_{g}(\lambda) = \text{argmin}_{\lambda} \ h(\lambda) + \frac{1}{2\eta} \| \text{vec}(\lambda - Y) \|_2^2$$

where vec(·) is the vectorization operation to transform a tensor into a vector. The key steps of Algorithm 1 are to evaluate the proximal operators of $n_f$ and $n_g$ in Step 2 and Step 3.

As $f(\tilde{B})$ is the sum of multiple terms involving disjoint sets of variables, we can evaluate the proximal operator of each component in $f(\tilde{B})$ individually as below.

**Evaluation of the proximal operator of $L(B_1^{(1)}, B_2^{(1)})$.** The first term $L(B_1^{(1)}, B_2^{(1)})$ can be presented as

$$L(B_1^{(1)}, B_2^{(1)}) = \| Y - \lambda \ast B_1^{(1)} - \mathcal{H} + B_2^{(1)} \|_F^2$$  \[6\]

which can be transformed into [7], according to Appendix A.

$$L(B_1^{(1)}, B_2^{(1)}) = \| Y - [ \lambda_{(1)} \ast \mathcal{H}(1) \left[ B_1^{(1)} \right] ]_2 \|_2$$  \[7\]

We see that $L(B_1^{(1)}, B_2^{(1)})$ is a quadratic function of each element in $B_1^{(1)}$ and $B_2^{(1)}$. In this way, the proximal operator of $L(B_1^{(1)}, B_2^{(1)})$ can be evaluated according to the proximal operator of quadratic functions as detailed in Appendix B.

**Evaluation of the proximal operator of $p_1(B_2^{(1)})$ and $p_2(B_2^{(2)})$.** The proximal operator of $p_1(B_2^{(2)})$ can be evaluated using

$$\text{prox}_{\lambda \| \cdot \|_2^2}(\lambda) = S_{\lambda}(\cdot) := \left( 1 - \frac{\lambda}{\| \cdot \|_2^2} \right)_{+} \lambda$$  \[8\]
that the proximal operator in Step 3, presented in Algorithm 2. As the steps in (2a-2f) involve the same data structure as \( B \), those steps can be performed in parallel. Therefore, this algorithm is highly parallelizable and efficient for the computation of tensors of high-dimensionality.

Algorithm 2: the complete optimization procedure

Initialize \( \tilde{U} = (U^{(1)}, U^{(2)}, U^{(3)}, U^{(1)}T U^{(2)}, U^{(1)}T U^{(3)}) \) as the same data structure as \( \tilde{B} = (B^{(1)}, B^{(2)}, B^{(3)}), \) with all elements being 0.

Do:

1. Save \( \tilde{B}_{1, \text{prev}} \leftarrow \tilde{B}_1, \tilde{B}_{2, \text{prev}} \leftarrow \tilde{B}_2. \)
2. Perform the following calculations in parallel:

\[
(2a) \quad \begin{bmatrix} B^{(1)}_1 \& B^{(1)}_2 \end{bmatrix} = \left( I + \eta [X(1) \& H(1)] \right)^{-1} \begin{bmatrix} \mathcal{B}^{(1)}_1 \& \mathcal{B}^{(1)}_2 \end{bmatrix}
\]

\[
(2b) \quad \text{vec}( B^{(2)}_{j, \text{prev}} ) \leftarrow S_{\mu j} \left( \text{vec}( B^{(2)}_{j, \text{prev}} - U^{(2)}_{j, \text{prev}} ) \right) \]

\[
(2c) \quad B^{(3)}_1 = \Sigma_i (\sigma_i - \eta \lambda_i) \cdot u_i v_i^T, \text{ where } \Sigma_i \sigma_i u_i v_i^T \text{ is the singular value decomposition of } (\mathcal{B}_1 - U^{(1)}) (3) ;
\]

(2d) For \( m_1 = 1, \ldots, M, m_2 = 1, \ldots, M, \) update

\[
\text{vec}( B^{(2)}_{j, \text{prev}} ; m_1, \ldots, m_2 ) \leftarrow S_{\mu j} \left( \text{vec}( B^{(2)}_{j, \text{prev}} ; m_1, \ldots, m_2 ) - U^{(2)}_{j, \text{prev}} ; m_1, \ldots, m_2 \right) \]

(2e) \quad B^{(3)}_2 = \Sigma_i (\sigma_i - \eta \lambda_i) \cdot u_i v_i^T, \text{ where } \Sigma_i \sigma_i u_i v_i^T \text{ is the singular value decomposition of } (\mathcal{B}_2 - U^{(2)}) (5).

3. Update \( \tilde{B}_1 \) and \( \tilde{B}_2 \) via

\[
\tilde{B}_j \leftarrow \frac{1}{\gamma} \left( B_j^{(1)} + B_j^{(2)} + B_j^{(3)} \right), \quad \tilde{B}_2 \leftarrow \frac{1}{\gamma} \left( B_j^{(1)} + B_j^{(2)} + B_j^{(3)} + T_34125 ( B_j^{(2)} + B_j^{(3)} ) \right)
\]

4. Update \( U^{(1)}_i \leftarrow U^{(1)}_i - B^{(1)}_i, \quad U^{(2)}_i \leftarrow U^{(2)}_i + B^{(2)}_i, \quad U^{(3)}_i \leftarrow U^{(3)}_i + B^{(3)}_j - \tilde{B}_j \) for \( i = 1, 2, 3. \)

Until \( \max_{i=1,2,3} \| B^{(1)}_i - B^{(1)}_i \| < \epsilon, \max_{i=1,\ldots,4} \| B^{(2)}_i - B^{(2)}_i \| < \epsilon, \| \tilde{B}_i - \tilde{B}_i_{\text{prev}} \| < \epsilon, \| \tilde{B}_j - \tilde{B}_j_{\text{prev}} \| < \epsilon \)

3.3. Discussions

As we have seen in the Introduction, the model complexity increases dramatically when the interaction effect is considered in the tensor-on-tensor regression model. In the STOTI method, the assumptions on the prior knowledge motivated multiple regularization terms applied on slices of the coefficient tensors, which helps to decrease the model complexity. In Section 3.1, the problem formulation is based on an order-2 input tensor and an order-1 output tensor motivated by the engineering problem. However, the readers should note that the method of regularizing certain modes of the coefficient tensors can be extended to higher-order tensors and a larger variety of the structural properties between input and output tensor in general. First, if the input tensor is an order-\( d_1 \) tensor of shape \( S_1 = p_1 \times \cdots \times p_{d_1} \), the output tensor...
is of shape $S_2 = q_1 \times \cdots \times q_d$, the shape of the main effect coefficient is of shape $p_1 \times \cdots \times p_d \times q_1 \times \cdots \times q_d$, and the interaction effect tensor is of shape $p_1 \times \cdots \times p_d \times p_1 \times \cdots \times p_d \times q_1 \times \cdots \times q_d$. Second, the sparsity of effective slices in the tensor $\mathcal{X}$ can be defined for multiple modes. The optimization problem can be solved by an ADMM consensus algorithm with a similar procedure as detailed in Section 3.2.

When the sample size is small or when collinearity between the input tensors exists, the structural information is insufficient to determine a unique regression coefficient tensor $B_1, B_2$. When this happens, we propose to include a regularization term $\epsilon_j \|B_j\|_F, j = 1, 2$ into the objective function, where $\epsilon_j, j = 1, 2$ are small positive real numbers. With this term, the objective function becomes strong convex and thus it has a unique solution. The solution procedure based on ADMM is similar and thus omitted.

4. Simulation studies for performance assessment

In this section, we evaluate the methodology proposed in Section 3 in estimating the main effect and the interaction effect tensors, and in predicting the outputs under four data scenarios. In these four scenarios, we consider two magnitudes of the errors measured by the signal to noise ratios $\text{SNR} = \frac{|\mathcal{X} + B_1 + \mathcal{N} + B_2|}{|\mathcal{N}|}$. We also consider the situations where the output $\mathcal{Y}$ is related to, or not related to, the interaction of the elements in $\mathcal{X}$. The four cases are summarized below.

- Case 1. The elements in the input tensor $\mathcal{X}$ have interaction effects and $\text{SNR} = 5$.
- Case 2. The elements in the input tensor $\mathcal{X}$ have interaction effects and $\text{SNR} = 10$.
- Case 3. The elements in the input tensor $\mathcal{X}$ do not have interaction effects and $\text{SNR} = 5$.
- Case 4. The elements in the input tensor $\mathcal{X}$ do not have interaction effects and $\text{SNR} = 10$.

For each case, the input tensor is of shape $4 \times 6$, denoting the six process variables measured from each of the four stages. The output is of size $6$. The coefficient tensors are $B_1 \in \mathbb{R}^{4 \times 6 \times 6}$ and $B_2 \in \mathbb{R}^{4 \times 6 \times 4 \times 6 \times 6}$. A dataset of $N = 500$ samples are generated. The detailed procedure of data generation is introduced in Section 4.1.

4.1. Data generation

To generate the input tensor $\mathcal{X}$ and the output tensor $\mathcal{Y}$ under each case, we first generate $B_1$ and $B_2$ that meet our understanding of the dataset. Considering that the elements in $\mathcal{X}$ do not have interaction effects in Case 3 and Case 4, we let $B_2 = 0$ under these cases. We also let the output variation patterns be limited by restricting the mode-3 matricization of $B_1$ low-rank, and restricting mode-5 matricization of $B_2$ low-rank. As sparse slices of the input tensor $\mathcal{X}$ along mode-3 affect the output tensor, we let a small number of mode-2 slices of the regression coefficient tensor $B_1$ be non-zero, and a small number of mode-(2, 4) slices of the tensor $B_2$ be non-zero.

To satisfy the above requirements, the detailed data generation procedure is as follows. We generate $B_1$ and $B_2$ by using Eqs. [10] and [11], respectively, to guarantee that the slices of $B_1$ in mode-3 is of low rank and the slices of $B_2$ in mode-5 is of low rank.

$$B_1 = C_1 \times_3 V_{B_1} \quad [10]$$

$$B_2 = C_2 \times_5 V_{B_2} \quad [11]$$

where $V_{B_1} \in \mathbb{R}^{4 \times 6}$, $V_{B_2} \in \mathbb{R}^{5 \times 6}$, $C_1 \in \mathbb{R}^{4 \times 6 \times 3}$, and $C_2 \in \mathbb{R}^{4 \times 6 \times 4 \times 6 \times 5}$. The columns of $V_{B_1}$ and $V_{B_2}$ are generated as random orthonormal vectors. Below we detail the generation of $C_1$ and $C_2$. The elements in both $C_1$ and $C_2$ are initialized as zero values.

To generate $C_1$, we first generate a basis $f_{1, 1, \ldots, 5} \in \mathbb{R}^4$ where

$$f_{1,i} = \left[ \begin{array}{c} \cos \frac{2\pi}{5} (i - 1) \\ \sin \frac{2\pi}{5} (i - 1) \\ \cos \frac{4\pi}{5} (i - 1) \\ \sin \frac{4\pi}{5} (i - 1) \end{array} \right], i = 1, \ldots, 5.$$

Then we assign each $f_{1,i}$ to the 2nd or the 4th columns of the mode-(3) slices of $C_1$ according to [12].

$$C_1(:, 2, j) = f_{1, 2j - 1}, C_1(:, 4, j) = f_{1, 2j}, j = 1, 2, 3. \quad [12]$$

The above basis are adopted so that the non-zero columns on all the slices at mode-(1, 2) are orthogonal.

We generate another basis matrix $F_2 \in \mathbb{R}^{4 \times 20}$ with its each column $i$ represented by $f_{2, i}:

$$f_{2,i} = \left[ \begin{array}{c} \cos \frac{2\pi}{19} (i - 1) \\ \sin \frac{2\pi}{19} (i - 1) \\ \cos \frac{4\pi}{19} (i - 1) \\ \sin \frac{4\pi}{19} (i - 1) \end{array} \right], i = 1, \ldots, 20.$$

and we reshape $F_2$ into a multiaarray $\mathcal{F}_2 \in \mathbb{R}^{4 \times 4 \times 5}$. We assigned it to the (2, 4) slice of the tensor $C_2$ at mode-(2, 4) according to [13].

$$C_2(:, 2, :, 4, :) = \mathcal{F}_2 \quad [13]$$

Considering the symmetricity of interaction between variables, we leave the values in
C_2(4, 2, :) to be zero. With the generated C_1 and C_2, we can obtain B_1 and B_2 according to [10] and [11], respectively. Elements in the input tensor \( X' \) and the error tensor \( E \) are independently generated from Gaussian distributions of \( N(0, 1) \) and \( N(0, \sigma^3) \), respectively, where the value of \( \sigma \) is adjusted to achieve the desired SNR in each case.

4.2. Benchmark methods and performance measure

Besides STOTI, the benchmark methods used in our simulation study are based on the ordinary least square (OLS), the partial least square (PLS), the CP decomposition-based tensor-on-tensor regression (CP-TOT) of Lock (2018), and the Tucker decomposition-based multiple tensor-on-tensor regression (Tucker-MTOT) of Gahrooei et al. (2021). To understand the necessity of integrating interaction effects into the model, we consider two variations of each benchmark method mentioned above: one variation integrates the interaction effects of the input tensor into the model, and the other does not consider the interaction effects. Therefore, we compare ten predictive methods in this simulation study, listed as follows.

- Without interactions: OLS, PLS, CP-TOT, Tucker-MTOT, and STOT.
- With interactions: OLSI, PLSI, CP-TOTI, Tucker-MTOTI, and STOTI.

Among them, OLSI, PLSI, CP-TOTI and Tucker-MTOTI integrate the interaction effects by naturally regard the interaction effect tensor \( H = X' \times X' \) as an additional input tensor. By removing the term \( H \times B_2 \) from the model of the STOTI method, we have the STOT method, where the coefficient tensor \( B_1 \) is estimated based on the regularized least-square method. The detailed description and the implementation of these benchmark methods are provided in the supplementary material.

Performance measure We use the mean squared prediction error \( MSPE = \frac{1}{100} \| Y - \hat{Y} \|_F^2 \) to evaluate the prediction accuracy of the methods in the comparison study. We use the relative estimation error \( REE = \frac{\| B_i - \hat{B}_i \|_F}{\| B_i \|_F} \) to evaluate the estimation accuracy for Case 1 and Case 2, and we use the absolute estimation error \( AEE = \| B_i - \hat{B}_i \|_F \) to evaluate the estimation accuracy for Case 3 and Case 4, as \( B_2 = 0 \) in both cases and thus the REE is not defined.

4.3. Results analysis

We randomly divide the \( N = 500 \) samples generated into a set of 400 training samples and a set of 100 testing samples in each case. Then we calculate the MSPE and REE (or AEE) for the proposed STOTI method and the benchmark methods based on the training data and the testing data. The above procedure is repeated 50 times to estimate the mean and standard deviation of the MSPE and REE (or AEE) for each case.

Figure 2 (a)-(d) presents the boxplots of the estimation accuracy of the regression coefficient tensors \( B_1 \) using ten methods for the four cases, and Figure 3 (a)-(d) similarly shows the estimation accuracy of \( B_2 \) using the five methods that take the interaction of input tensors into consideration for the four cases. The estimation accuracy of the regression coefficient tensor \( B_2 \) for the methods that do not consider the interaction effect is not reported. Figure 4 presents the boxplots of the mean squared prediction error by each comparison method in the four cases. The detailed results of the estimation and prediction by each method in each case are listed in the tables in the supplementary material. The tuning parameters of the proposed STOTI method are determined as \( \eta = 2, \lambda_1 = 250, \lambda_2 = 500, \lambda_3 = 4, \lambda_4 = 10 \) by the 5-fold cross-validation minimizing the mean squared prediction error.

Comparing all those methods in all the four cases, the proposed STOTI method achieves the best performance in estimating the coefficient tensors of \( B_1 \) and \( B_2 \) for Case 1 and Case 2 when the interactions of certain elements in the input tensor indeed correlate with the output tensor data. This result can be
seen from Figures 2 and 3. The STOT achieves the best performance of estimation accuracy of $B_1$ in Case 3 and Case 4, whereas the performance of the STOTI method is very close to this best performance. The methods CP-TOT, CP-TOTI, Tucker-MTOT and Tucker-MTOTI are not as effective as our proposed method. This is because they restrict the regression coefficient tensors in the form of low-order CP or Tucker decomposition, which is not compatible with the true relationship between the input and the output. The good performance of STOTI in estimation and prediction is thus validated.

To obtain further insights into the behavior of the STOTI method, we give an intuitive illustration of the tensor $^\ast B_1$ and $^\ast B_2$ estimated from the STOTI method and compare them with the true values of $B_1$, $B_2$ of Case 2 in Figure 5(a1) - (a2) and (b1) - (b2). Specifically, Figure 5(a1) - (a3) shows the true value of $B_1^{(3)}$, its estimation $B_1^{(3)}$ obtained from the STOTI method, and their difference; and Figure 5(b1) - (b3) shows the true value of $B_2^{(1,2)}$, the estimation $B_2^{(1,2)}$ from STOTI, and their difference. These figures show that the proposed method accurately identifies the effective variables in both the main effect coefficient tensor and the interaction effect coefficient tensor. The elements values in the main effect coefficient tensor are also correctly estimated. Figure 5(b3) shows that the estimation error of the effective elements in the interaction coefficient tensor is farther away from zero compared with other values. The reason is that the regularization introduces bias to the estimation of these coefficients when it simultaneously decreases the variability of the estimation for those zero elements in the coefficient tensor and achieves minimal error of cross-validation. However, this bias has little impact on the effective variable identification. In general, we can see that STOTI not only gives an accurate estimation, visualizing the values of its elements also leads to good interpretability on the relationship between the input and output tensor.

5. Case study

In this section, we apply the proposed framework to the data obtained from the real production of hot rolling processes. Specifically, we model the relationship between the process variables and the rolling bar’s output dimensions and test the prediction accuracy of the proposed STOTI method.

The production data of 300 billets are collected. These products contain the same type of material and have the same size, and they went through 10 rolling stands in the hot rolling mill. For each billet, the process variables of speed, roll gap, effective diameter, and rod area were measured. After these 10 mills, the average rod diameters are obtained along six directions $y_1, \ldots, y_6$, as illustrated in Figure 1. We scale the range of the responses to [0,1]. In the end, we obtain an input tensor $X \in \mathbb{R}^{N \times K \times M}$ and an output tensor $Y \in \mathbb{R}^{N \times D}$ with $N = 300, K = 10, M = 5$ and $D = 6$.

We apply the STOTI method to model the rolling data with the following configurations. As suggested in Section 3.3, two terms of $\epsilon_1 \| B_1 \|_F^2$ and $\epsilon_2 \| B_2 \|_F^2$ are included in the objective function. With tuning
parameters $\lambda_1 = 0.001, \lambda_2 = 0.01, \lambda_3 = 0.003, \lambda_4 = 0.4$, and $\epsilon_1 = \epsilon_2 = 1 \times 10^{-4}$, we solved this problem using the ADMM algorithm.

We randomly divide the samples into training (80%) and testing (20%) data set ten times. In each time, we develop a predictive model based on the training data set, predict the output tensor for the testing data set with this model, and evaluate the prediction accuracy by calculating the MSPE.

Table 1 lists the mean and standard deviation (in the bracket) of the MSPEs for the test samples by all five methods considering interaction effects: OLSI, PLSI, CP-TOTI, Tucker-MTOTI, and STOTI. The proposed method STOTI achieves the best performance compared with the benchmark methods. The OLSI has the worst performance as estimating massive independent parameters within both main and interaction effect tensor causes overfitting. Both CP-TOTI and Tucker-MTOTI do not perform well, because they implicitly assume inappropriate structures of the tensor coefficients. The performance of PLS is also inferior as it misses the sparsity structure of the coefficient tensor.

Furthermore, Figure 6(a) and (c) illustrates the matricized coefficient tensor $B_1$ along mode-(1, 2) and $B_2$ along mode-(1, 2), respectively. Both $B_1$ and $B_2$ present repetitive patterns in all output dimensions, indicating that these matrices have low rank. This characteristic is due to the limited variation patterns of the product dimensions. From Figure 6(a), we observe the speed of stands 2, 9, 10 and the roll gap of stands 3-9 have important effects on the dimension quality. Figure 6(b) presents an enlarged view of the interaction coefficients marked in the last column in Figure 6(c), corresponding to the interaction coefficients regressed on $y_6$. They indicate that the interactions between speed and rod area, between speed and effective diameter are of great importance on the dimension quality.

Table 1. Comparison between the proposed methods and the benchmarks on the rolling process.

<table>
<thead>
<tr>
<th>OLSI</th>
<th>PLSI</th>
<th>CP-TOTI</th>
<th>Tucker-MTOTI</th>
<th>STOTI</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSPE</td>
<td>0.0976</td>
<td>4.18e-04</td>
<td>7.10e-04</td>
<td>7.50e-04</td>
</tr>
<tr>
<td></td>
<td>(0.0397)</td>
<td>(5.14e-05)</td>
<td>(6.28e-05)</td>
<td>(1.15e-04)</td>
</tr>
</tbody>
</table>

Figure 5. Comparison between the ground truth and the estimation of $B_1$ and $B_2$ by STOTI on case 2.
6. Conclusion

In an MMS, sensors are widely applied to manufacturing stations to measure the product quality and the process variables. These process inputs and outputs are usually in tensor formats. In a tensor regression problem, the key consideration is exploiting the low-dimensional structure of high dimensional coefficient tensor, to alleviate the curse of dimensionality and resolve the computational challenges. In this paper, we extend the tensor regression model to include the tensor interaction effect. As it involves interaction effect tensor that has even higher order, the curse of dimensionality becomes more challenging. To address this challenge, the method presented in this paper represents the prior knowledge on the tensor inputs and output with specific structure of the coefficient tensor, and these structures are typically associated with multiple slices along with certain modes of the tensor, or with specific reshape of the tensor. Finally, these structures are expressed using multiple regularization terms in the optimization problem. Compared with the existing and widely applied methods of lower-order tensor

Figure 6. Estimated coefficient tensor $B_1$ and $B_2$ for the rolling process.

Figure 7. Scatter plots to illustrate relationships among variables with/without interactions.
decomposition, our method can represent more versatile structures of the coefficient tensors and thus is applicable to a wider variety of engineering problems.

Our method is called Structural Tensor-On-Tensor Regression with the Interaction effects (STOTI). It solves a penalized least square problem representing the main effects and interaction effects tensors’ structural information. An ADMM algorithm is applied to estimate the regression coefficient tensors. A numerical example containing different data scenarios validates the accuracy and effectiveness in estimation and prediction of the proposed STOTI method. Furthermore, the proposed STOTI method has been further validated on the real production dataset obtained in a hot rolling process. In the practice of modeling the relationship between tensor input and output, we would like to point out that a suitable modeling approach should be to first model the data using all plausible alternative methods including the STOT method and the STOTI method, if the model assumptions detailed in the paper are satisfied. Then, one should pick the model with the smallest magnitude of prediction error. In this sense, our newly developed STOTI method extends the capability to model the system by considering the interaction of the input tensor.

The proposed STOTI method is flexible and expandable. Its essential idea is to address the prior knowledge on the tensor regression relationship through regularizing the reshaped coefficient tensor. This idea can be easily extended to other scenarios of tensor regression. For example, when there are multiple input tensors $X_1, ..., X_p$ of different shapes, we can similarly represent the second-order interaction of two inputs by adding the terms of shape $B_{ij} \ast (X_i \ast X_j)$ into the model, where the shape of $B_{ij}$ is the product of $X_i$, $X_j$ and $Y$'s shapes. Also, we can define the relationship between smooth input tensor and smooth output tensor, through smoothness regularizations on the coefficient tensors, according to the description of these tensor structures by Mou et al. (2020). When these regularizations reflect the reality of the relationship between input tensors and output tensors, they will alleviate the curse of dimensionality and have the potential to reduce the estimation errors. The practitioners may tailor the problem formulation based on the knowledge of the relationship between input and output tensors, by incorporating appropriate regularizations on the coefficient tensors and adapting the ADMM algorithm accordingly.

About the authors

Ms. Huihui Miao received her BS and MS degrees in mechanical engineering from Xi’an Jiaotong University, Xi’an, China, in 2011 and 2014, respectively. She received an Engineering Degree from Ecole Nationale Supérieure d’Arts et Métiers ParisTech in 2014 for a joint degree program. From September 2019 to September 2021, she was a Visiting Scholar with the H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology. She is a Ph.D. Candidate with the School of Mechanical Engineering at Xi’an Jiaotong University. Her research interest lies in machine learning for modeling, monitoring, and diagnosis of manufacturing and machinery systems.

Dr. Andi Wang received his B.S. in Statistics from Peking University in 2012 and his Ph.D. in Industrial Engineering from Hong Kong University of Science and Technology in 2016. He also received his M.S. in Computer Science and Engineering and another Ph.D. in Industrial Engineering (System Informatics and Control) from Georgia Institute of Technology in 2021. He is currently an assistant professor in the School of Manufacturing Systems and Networks in Ira A. Fulton Schools of Engineering, Arizona State University. Andi Wang’s research focuses on the intersection of data science and manufacturing systems. His research involves applying machine learning, high-dimensional statistics, and advanced optimization techniques to solve the challenges in manufacturing systems, and perform root-cause diagnostics, monitoring, design optimization, prediction for complex, interconnected, and intelligent systems. He is a recipient of Wayne Kay Scholarship from SME, a recipient of INFORMS 2019 Data Mining Best Paper Finalist Award, INFORMS 2020 Quality Reliability and Statistics Best Paper Finalist Award, IIEE QCRE 2021 Best Paper Finalist Award.

Dr. Bing Li, received his BS degree in Thermal Engines and the MS degree in Marine Engineering at Northwestern Polytechnical University, Xi’an, in 1999 and 2002, respectively, and the Ph.D. degree in Mechanical Engineering at Xi’an Jiaotong University, Xi’an, in 2005. From November 2005 to April 2007, he worked as a senior engineer at ANSYS. He is currently a Professor of Mechanical Engineering with Xi’an Jiaotong University. His research interests include nonlinear dynamics, signal processing and analysis, fault diagnosis, and prognosis for machinery equipment.

Dr. Jianjun Shi received the B.S. and M.S. degrees in automation from the Beijing Institute of Technology in 1984 and 1987, respectively, and the Ph.D. degree in mechanical engineering from the University of Michigan in 1992. Currently, Dr. Shi is the Carolyn J. Stewart Chair and Professor at the Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology. His research interests include the fusion of advanced statistical and domain knowledge to develop methodologies for modeling, monitoring, diagnosis, and control for complex manufacturing systems. Dr. Shi is a Fellow of four professional societies, including ASME, IISE, INFORMS, and SME, an elected member of the International Statistics Institute, a life member of ASA, an Academician of the International Academy for Quality (IAQ), and a member of National Academy of Engineers (NAE).
References


Appendix A

This appendix provides the proof of derivation of Eq. [7] from Eq. [6].

\[ L(B_1, B_2) = \|Y - X*B_1 - H*B_2 \|^2_F \]

[A1]

Since,

\[ [X(1)B(13) ]_{n,k} = \sum_{i,j} X_{n,i,j} B^1_{1,k} \]

\[ [H(1)B(25) ]_{n,k} = \sum_{i,j} H_{n,i,j} B^2_{1,k} \]

Thus,

\[ L(B_1, B_2) = \sum_{n,k} \{ Y_{n,k} - \sum_{i,j} X_{n,i,j} B^1_{1,k} - \sum_{n,i,j} H_{n,i,j} B^2_{1,k} \}^2 \]

\[ = \sum_{n,k} \| Y_{n,k} - [X(1)H(1)]_{n,k} \|^2_F \]

\[ = \sum_{n,k} \| Y_{n,k} - [X(1)H(1)]_{n,k} B^1_{13}B^2_{25} \|^2_F \]

[A2]

Let \[ X = [X(1)H(1)] \], \[ B = [B^1_{13}B^2_{25}] \], then \[ L(B_1, B_2) \] can be rewritten into [A3],

\[ L(B) = \| Y - XB \|^2_F \]

[A3]
Appendix B

This appendix provides the proof of derivation of the proximal operator of [A3] in Appendix A.

$$\text{prox}_f(V) = \arg\min_B \left( \frac{1}{2} ||y - XB||^2_2 + \frac{1}{2\eta} ||B - V||^2_2 \right)$$

Let

$$h(B) = \text{tr} \left( (y - XB)^T (y - XB) \right)$$

$$= \text{tr} (y^T y) - \text{tr} (B^T X^T y) - \text{tr} (y^T XB) + \text{tr} (B^T X^T XB)$$

(B1)

$$= \text{tr} (y^T y) - 2 \text{tr} (B^T X^T y) + \text{tr} (B^T X^T XB)$$

Let

$$h = ||B - V||^2_2$$

$$= \text{tr} \left( (B - V)^T (B - V) \right)$$

Then,

$$\frac{\partial h}{\partial B} = -2X^T y + 2X^T XB$$

$$\frac{\partial h}{\partial B} = \frac{2}{\eta} (B - V)$$

Let,

$$\frac{\partial f(B)}{\partial B} = -X^T y + X^T XB + \frac{1}{\eta} (B - V)$$

$$= 0$$

Thus,

$$B = \left( \eta X^T X + I \right)^{-1} (V + \eta X^T y)$$

That is,

$$\text{prox}_f(V) = \left( \eta X^T X + I \right)^{-1} (V + \eta X^T y)$$