Tensorial Change Analysis Using Probabilistic Tensor Regression

Tsuyoshi Idé
IBM Research, Thomas J. Watson Research Center
tide@us.ibm.com

Abstract

This paper proposes a new method for change detection and analysis using tensor regression. Change detection in our setting is to detect changes in the relationship between the input tensor and the output scalar while change analysis is to compute the responsibility score of individual tensor modes and dimensions for the change detected. We develop a new probabilistic tensor regression method, which can be viewed as a probabilistic generalization of the alternating least squares algorithm. Thanks to the probabilistic formulation, the derived change scores have a clear information-theoretic interpretation. We apply our method to semiconductor manufacturing to demonstrate the utility. To the best of our knowledge, this is the first work of change analysis based on probabilistic tensor regression.

Introduction

Change detection in temporal data has a variety of applications across many industries. Depending on the specific type of data and changes expected, a number of different machine learning tasks can be defined. Of particular importance is change detection in the supervised setting, whose goal is to detect a change in the relationship between the input and output variables. By analyzing the nature of the detected change in terms of controllable input variables, we can obtain actionable insights into the system.

In the supervised setting, change detection has been treated typically as a regression problem. For a recent comprehensive review from an application perspective, see (Ge et al. 2017). As a natural extension of conventional vector-based regression approaches, condition-based monitoring based on *tensor regression* has attracted recent attention (Zhu, He, and Lawrence 2012; Fanaee-T and Gama 2016). As a real-world example, Fig. 1 illustrates the etching process in semiconductor manufacturing. One etching round consists of many, say 20, etching steps (chemical gas introduction, plasma exposure, etc.) and each of the steps is monitored with the same set of sensors (pressure, temperature, etc.), resulting in a two-way tensor $\boldsymbol{\mathcal{X}}$ as the input. Note that neither y nor $\boldsymbol{\mathcal{X}}$ is constant even under the normal condition due to different production recipes, random fluctuations, aging of the tool, etc. After etching, semiconductor wafers are sent to an inspection tool, which gives each of the wafers a scalar y representing the goodness of etching.

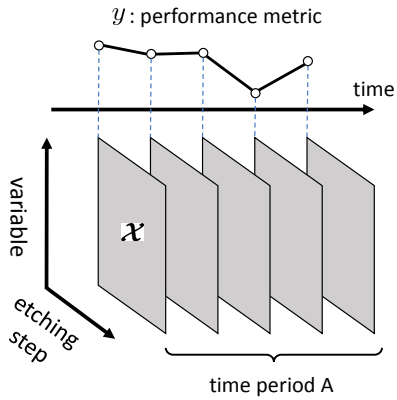

Figure 1: Illustration of industrial change analysis using tensor regression in semiconductor etching. A scalar y representing the goodness of etching is predicted as a function of etching trace data $\boldsymbol{\mathcal{X}}$ in a tensor format. How can we quantitatively compare the time period A with a "golden period"?

When monitoring the process, imagine that we observed an unusual trend in y for a certain time period ("time period A" in the figure). In order to get insights into how to fine-tune process parameters, we need to know *how* the current situation is different from a "golden period", in which everything looked in good shape, in terms of the relationship between $\boldsymbol{\mathcal{X}}$ and y. This is indeed a motivating example of the *regression-based tensorial change analysis*.

For practical change analysis, three major requirements should be met. *First*, obviously, a change analysis model must be able to quantitatively explain which tensor modes and dimensions contribute most to the changes detected. *Second*, it must be built on a probabilistic model between $\boldsymbol{\mathcal{X}}$ and y. The input tensor generally includes different physical quantities. To make them comparable on the same ground, change scores should be formalized information-theoretically. *Third*, the change scores must be efficiently computed in both the training and testing phases.

To meet these requirements, we propose a new change *analysis* framework based on a probabilistic tensor regres-

sion. Our tensor regression model can be viewed as a Bayesian re-formulation of the conventional alternating least squares (ALS) algorithm (Signoretto et al. 2014; Yu and Liu 2016; Zhou, Li, and Zhu 2013; Zhu, He, and Lawrence 2012). Although exact Bayesian inference is not possible, we derive an iterative inference algorithm using a variational expectation-maximization framework (Tzikas, Likas, and Galatsanos 2008). Our contribution is the first proposal of (1) a Bayesian extension of ALS for tensor regression and (2) information-theoretic tensorial change analysis using probabilistic tensor regression.

Related work

There are two categories of related work: tensorial change detection and probabilistic tensor regression models.

For the former, although much work has been done in sequential tensor tracking using tensor factorization (Sun et al. 2008; Dunlavy, Kolda, and Acar 2011; de Araujo, Ribeiro, and Faloutsos 2017), most of them are based on the unsupervised setting. Little is known about the supervised setting, especially information-theoretic approaches to performing both change detection and analysis.

For the latter, unlike tensor factorization, which has been a major research topic in machine learning for years, tensor regression is relatively new. For probabilistic tensor regression, two major approaches can be found in the literature: kernel methods and non-kernel methods. For the former, most of the existing studies attempt to extend Gaussian process regression (GPR) for tensors (Zhao et al. 2014; Hou, Wang, and Chaib-draa 2015; Suzuki 2015; Kanagawa et al. 2016; Imaizumi and Hayashi 2016). It may look straightforward to mechanically use the well-known formula of GPR (Rasmussen and Williams 2006), assuming that a kernel function between tensors is given. However, the tricky part is that naive distance metrics such as $\|\boldsymbol{\mathcal{X}}^{(n)} - \boldsymbol{\mathcal{X}}^{(n')}\|_{\mathrm{F}}^2$, where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, do not properly preserve structural information of tensors because they are reduced to the summation of the element-wise distance, giving exactly the same expression as the naive vectorized formulation.

To handle this issue, Zhao et al. (2014) and Ho et al. (2015) proposed to use a kernel function defined through mode-wise matricization. Kanagawa et al. (2016) considered GPR for the individual tensor modes $l = 1, \ldots, M$ and combine them for predicting y. A similar approach was also used in (Suzuki 2015; Imaizumi and Hayashi 2016). For real-world industrial applications, however, these methods have significant limitations because, unlike the proposed Bayesian ALS, they need either Monte Carlo (MC) sampling on training or expensive computation of tensor factorization on testing.

For the non-kernel methods, only a limited number of studies is found in the literature. One of the earliest studies is Goldsmith et al. (2014) but it is based on a few strong assumptions specific to 3D imaging. Recently, Guhaniyogi *et al.* (2017) proposed a fully Bayesian tensor regression model with variable selection, using the CP (canonical polyadic) decomposition assumption for the regression coefficients. As a result of its multi-layered hierarchical model with a fully Bayesian treatment, however, their model requires MC sampling for inference, making practical implementation hard especially in the context of change analysis. Also, mainly due to the complexity of the model, how it is related to the existing alternating least squares and other regression work is not necessarily clear. In contrast, in the proposed probabilistic model all the steps for inference have a simple analytic expression thanks to a variational approximation. To the best of our knowledge, this is the first work to use a variational inference approach for tensor regression.

Tensor notations

We follow Kolda and Bader (2009) for most of the tensor notations. We denote (column) vectors, matrices, and tensors by bold italic ($\boldsymbol{x}$, etc.), sanserif (A, etc.), and bold calligraphic ($\boldsymbol{\mathcal{X}}$, etc.) letters, respectively. The elements of them are typically denoted by corresponding non-bold letters with a subscript ($x_i, \mathsf{A}_{i,j}, \mathcal{X}_{i_1,\ldots,i_M}$, etc.). We may also use $[\cdot]$ as the operator to select a specific element ($[\boldsymbol{w}]_i \equiv w_i$, etc.), with $\equiv$ being used to define the left hand side (l.h.s.) by the right hand side (r.h.s.). To simplify the notation, we may use non-italic bold letters to collectively represent the indexes of the M modes as $\mathbf{i} = (i_1, \ldots, i_M)$. Superscripts are used to distinguish the tensor modes such as $\boldsymbol{a}^l$.

The inner product of two same-sized tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times \cdots \times d_M}$ is defined as

$$(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{A}}) \equiv \sum_{i_1,\ldots,i_M} \mathcal{X}_{i_1,\ldots,i_M} \mathcal{A}_{i_1,\ldots,i_M}. \quad (1)$$

The outer product between vectors is an operation to create a tensor from a set of vectors. For example, the outer product of $\boldsymbol{a}^1 \in \mathbb{R}^{d_1}, \boldsymbol{a}^2 \in \mathbb{R}^{d_3}, \boldsymbol{a}^3 \in \mathbb{R}^{d_3}$ makes a 3-mode tensor of $d_1 \times d_2 \times d_3$ dimension as

$$[\boldsymbol{a}^1 \circ \boldsymbol{a}^2 \circ \boldsymbol{a}^3]_{i_1,i_2,i_3} = a^1_{i_1} a^2_{i_2} a^3_{i_3}. \quad (2)$$

The inner product between a tensor and a rank-1 tensor plays a major role in this paper. For example,

$$(\boldsymbol{\mathcal{X}}, \boldsymbol{a}^1 \circ \cdots \circ \boldsymbol{a}^M) = \sum_{i_1,\ldots,i_M} \mathcal{X}_{i_1,\ldots,i_M} a^1_{i_1} \cdots a^M_{i_M}. \quad (3)$$

This can be viewed as a "convolution" of a tensor by a set of vectors. The m-mode product is an operation to convolute a tensor with a matrix as

$$[\boldsymbol{\mathcal{X}} \times_m \mathsf{S}]_{i_1,\ldots,j_m,\ldots,i_M} \equiv \sum_{i_m=1}^{d_m} \mathcal{X}_{i_1,\ldots,i_m,\ldots,i_M} \mathsf{S}_{j_m,i_m}. \quad (4)$$

Tensorial Change Analysis Framework

This section summarizes the problem setting and tensorial change detection framework at a high level.

Problem setting

We are given a training dataset D consisting of N pairs of a scalar target variable y and an input tensor $\boldsymbol{\mathcal{X}}$:

$$\mathrm{D} = \{(y^{(n)}, \boldsymbol{\mathcal{X}}^{(n)}) \mid n = 1, \ldots, N\}, \quad (5)$$

where the superscript in the round parenthesis is used to denote the n-th sample. Both the input tensor and the target variable are assumed to be *centered*. $\boldsymbol{\mathcal{X}}^{(n)}$'s have M modes in which the l-th mode has d_l dimensions.

We assume a linear relationship as $y \sim (\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{X}})$, in which the coefficient tensor follows the CP expansion of order R:

$$\boldsymbol{\mathcal{A}} = \sum_{r=1}^{R} \boldsymbol{a}^{1,r} \circ \boldsymbol{a}^{2,r} \circ \cdots \circ \boldsymbol{a}^{M,r}. \tag{6}$$

Using a probabilistic model described in the next subsection, our first goal is to obtain the predictive distribution $p(y \mid \boldsymbol{\mathcal{X}}, \mathrm{D})$ for an unseen sample $\boldsymbol{\mathcal{X}}$, based on the posterior distribution for $\{\boldsymbol{a}^{r,l}\}$ and the other model parameters learned from the training data.

Change analysis scores

We give the definition of change scores for three sub-tasks in tensorial change analysis: *Outlier detection, change detection*, and *change analysis*.

First, the outlier score is defined for a single pair of observation $(y, \boldsymbol{\mathcal{X}})$ to quantify how much uncommon they are in reference to the training data. Given the predictive distribution $p(y \mid \boldsymbol{\mathcal{X}}, \mathrm{D})$, we define the outlier score as the logarithmic loss (Yamanishi et al. 2004):

$$\begin{aligned} c(y, \boldsymbol{\mathcal{X}}) &\equiv -\ln p(y|\boldsymbol{\mathcal{X}}, \mathrm{D}) \\ &= \frac{\{y - \mu(\boldsymbol{\mathcal{X}})\}^2}{2\sigma^2(\boldsymbol{\mathcal{X}})} + \frac{1}{2}\ln\{2\pi\sigma^2(\boldsymbol{\mathcal{X}})\}, \end{aligned} \tag{7}$$

where the second line follows from the explicit from of $p(y \mid \boldsymbol{\mathcal{X}}, \mathrm{D})$ given later (Eq. (40)).

Second, we define the change-point score by averaging the outlier score over a set $\tilde{\mathrm{D}}$:

$$c(\tilde{\mathrm{D}}, \mathrm{D}) = \frac{1}{\tilde{N}} \sum_{n \in \tilde{\mathrm{D}}} c(y^{(n)}, \boldsymbol{\mathcal{X}}^{(n)}), \tag{8}$$

where $\tilde{N}$ is the sample size of $\tilde{\mathrm{D}}$. The set $\tilde{\mathrm{D}}$ is typically defined using a sliding window for temporal change-point detection.

Third, for change analysis, we leverage the posterior distribution of the CP-decomposed regression coefficient $\{\boldsymbol{a}^{l,r}\}$. As shown in the next section, the posterior distribution, denoted as $q_1^{l,r}(\boldsymbol{a}^{l,r})$, given by a Gaussian distribution:

$$q_1^{l,r}(\boldsymbol{a}^{l,r}) = \mathcal{N}(\boldsymbol{a}^{l,r} \mid \boldsymbol{\mu}^{l,r}, \boldsymbol{\Sigma}^{l,r}), \tag{9}$$

where the first and second arguments after the bar represents the mean and the covariance matrix, respectively. As explained in Fig. 1, the goal of change analysis is to quantify the contribution of each tensorial mode to the distributional difference between two datasets, say $\mathrm{D}, \tilde{\mathrm{D}}$. For the (l, r)-mode, this can be calculated as the Kullback-Leibler (KL) divergence of $q_1^{l,r}(a_i^{l,r}|\boldsymbol{a}_{-i}^{l,r})$, the conditional distribution for the i-th dimension given the rest:

$$\begin{aligned} c_i^l(\tilde{\mathrm{D}}, \mathrm{D}) &\equiv \frac{1}{R}\sum_{r=1}^{R} \int \mathrm{d}\boldsymbol{a}^{l,r} q_1^{l,r}(\boldsymbol{a}^{l,r}) \ln \frac{q_1^{l,r}(a_i^{l,r} \mid \boldsymbol{a}_{-i}^{l,r})}{\tilde{q}_1^{l,r}(a_i^{l,r} \mid \boldsymbol{a}_{-i}^{l,r})} \\ &= \frac{1}{2R}\sum_{r=1}^{R} \left\{ \frac{[\tilde{\boldsymbol{\Lambda}}^{l,r}(\tilde{\boldsymbol{\mu}}^{l,r} - \boldsymbol{\mu}^{l,r})]_i^2}{\tilde{\Lambda}_{i,i}^{l,r}} + \ln \frac{\Lambda_{i,i}^{l,r}}{\tilde{\Lambda}_{i,i}^{l,r}} \right. \\ &\qquad \left. + \frac{[\tilde{\boldsymbol{\Lambda}}^{l,r}\boldsymbol{\Sigma}^{l,r}\tilde{\boldsymbol{\Lambda}}^{l,r}]_{i,i}}{\tilde{\Lambda}_{i,i}^{l,r}} - 1 \right\} \end{aligned} \tag{10}$$

where the tilde $\tilde{}$ specifies the model learned on $\tilde{\mathrm{D}}$ and $\boldsymbol{\Lambda}^{l,r} \equiv (\boldsymbol{\Sigma}^{l,r})^{-1}$ etc., whose explicit form is given later (see Eq. (22)).

Note that in the above definitions the capability of producing probabilistic output is critical. Also, they can be straightforwardly computed without any expensive computations such as tensor factorization and MC sampling.

Probabilistic model for tensor regression

This section derives the inference algorithm of the proposed probabilistic tensor regression model.

Observation model and priors

Our probabilistic tensor regression model consists of only two primary ingredients: an observation model to describe measurement noise and a prior distribution to represent the uncertainty of regression coefficients.

First, the observation model for the centered data is defined as

$$p(y \mid \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{A}}, \lambda) = \mathcal{N}(y \mid (\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{X}}), \lambda^{-1}), \tag{11}$$

where $\mathcal{N}(y \mid \cdot, \cdot)$ denotes the univariate Gaussian distribution with the mean $(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{X}})$ and the precision λ.

Second, for the prior distribution of the coefficient vectors $\boldsymbol{a}^{l,r}$, we use the Gauss-gamma distribution as

$$p(\{\boldsymbol{a}^{l,r}\}) = \prod_{l=1}^{M}\prod_{r=1}^{R} \mathcal{N}(\boldsymbol{a}^{l,r} \mid \boldsymbol{0}, (b^{l,r})^{-1}\mathbf{I}_{d_l}), \tag{12}$$

$$p(b^{l,r}|\alpha_0, \beta_0) = \mathcal{G}(\alpha_0, \beta_0) \equiv \frac{{\beta_0}^{\alpha_0}}{\Gamma(\alpha_0)}(b^{l,r})^{\alpha_0 - 1}\mathrm{e}^{-\beta_0 b^{l,r}} \tag{13}$$

where $\mathbf{I}_{d_l}$ is the d_l-dimensional identity matrix and $\Gamma(\cdot)$ is the gamma function. The hyper-parameters α_0, β_0 are assumed to be given. Note that the parameter λ is determined as part of learning and there is no need for cross-validation. This is one of the advantages of probabilistic formulation and is in contrast to the existing frequentist tensor regression work (Signoretto et al. 2014; Yu and Liu 2016; Zhou, Li, and Zhu 2013; Zhu, He, and Lawrence 2012).

Variational inference strategy

For model inference, we employ the *variational expectation-maximization (vEM)* framework (Tzikas, Likas, and Galatsanos 2008). The idea of vEM is to integrate point-estimation into the variational Bayes (VB) method. Specifically, the variational E (VE) step finds the posterior distribution of $\{\boldsymbol{a}^{r,l}, b^{r,l}\}$ in the same way as VB, given the latest

point-estimated parameters. Then, given the posterior just estimated, the variational M (VM) step point-estimates the parameters λ by maximizing the posterior expectation of the log complete likelihood. The log complete likelihood of the model plays the central role here:

$$L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda) = \mathrm{c.} + \frac{1}{2}\sum_{n=1}^{N}\left\{\ln\lambda - \lambda\Delta(y^{(n)}, \boldsymbol{\mathcal{X}}^{(n)})^2\right\} + \sum_{l=1}^{M}\sum_{r=1}^{R}\left\{\frac{1}{2}d_l \ln b^{l,r} - \frac{1}{2}b^{l,r}\|\boldsymbol{a}^{l,r}\|_2^2\right\} + \sum_{l=1}^{M}\sum_{r=1}^{R}\left\{(\alpha_0 - 1)\ln b^{l,r} - \beta_0 b^{l,r}\right\}, \quad (14)$$

where c. is a symbolic notation for an unimportant constant, $\|\cdot\|_2$ is the 2-norm, and $\boldsymbol{b}$ is a shorthand notation for $\{b^{r,l}\}$. We also defined

$$\Delta(y, \boldsymbol{\mathcal{X}}) \equiv y - \sum_{r=1}^{R}(\boldsymbol{\mathcal{X}}, \boldsymbol{a}^{1,r} \circ \cdots \circ \boldsymbol{a}^{M,r}), \quad (15)$$

where we have omitted the dependency on $\boldsymbol{\mathcal{A}}$ on the l.h.s. for simplicity.

VE step: Posterior for coefficient vectors

The VB step finds an approximated posterior in a factorized form. In our case, we seek a VB posterior in the form

$$Q(\{\boldsymbol{a}^{l,r}, b^{l,r}\}) = \prod_{l=1}^{M}\prod_{r=1}^{R} q_1^{l,r}(\boldsymbol{a}^{l,r}) q_2^{l,r}(b^{l,r}). \quad (16)$$

The distributions $q_1^{l,r}, q_2^{l,r}$ are determined so that they minimize the KL divergence from the true posterior. The key fact here is that the true posterior is proportional to the complete likelihood by Bayes' rule. Thus, the KL divergence is represented as

$$\mathrm{KL} = \mathrm{c.} + \langle \ln Q\rangle - \langle L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda)\rangle,$$

where $\langle\cdot\rangle$ represents the expectation with respect to Q. Here the unknowns are not variables but functions. However, according to calculus of variations (see *e.g.* Appendix D in (Bishop 2006)), roughly speaking, we can formally take the derivative with respect to $q_1^{l,r}$ or $q_2^{l,r}$ and equate the derivatives to zero. In that way, the condition of optimality is given by

$$\textbf{VE step:}\quad \ln q_1^{l,r}(\boldsymbol{a}^{l,r}) = \mathrm{c.} + \langle L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda)\rangle_{\backslash \boldsymbol{a}^{l,r}}, \quad (17)$$

$$\ln q_2^{l,r}(b^{l,r}) = \mathrm{c.} + \langle L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda)\rangle_{\backslash b^{l,r}}, \quad (18)$$

where $\langle\cdot\rangle_{\backslash \boldsymbol{a}^{l,r}}$ and $\langle\cdot\rangle_{\backslash b^{l,r}}$ denotes the expectation with $Q/q_1^{l,r}$ and $Q/q_2^{l,r}$, respectively.

Now let us solve the first equation. Unlike the case of single-mode vector-based regression, $L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda)$ has a complex nonlinear dependency on $\{\boldsymbol{a}^{l,r}\}$, especially in the term of Δ^2. However, to compute $\langle\cdot\rangle_{\backslash \boldsymbol{a}^{l,r}}$, we can leverage the fact that each of the $\boldsymbol{a}^{l,r}$s can be factored out in the inner product:

$$(\boldsymbol{\mathcal{X}}, \boldsymbol{a}^{1,r} \circ \cdots \circ \boldsymbol{a}^{M,r}) = (\boldsymbol{a}^{l,r})^\top \boldsymbol{\phi}^{l,r}(\boldsymbol{\mathcal{X}}), \quad (19)$$

where $^\top$ denotes the transpose. The j-th element of $\boldsymbol{\phi}^{l,r}(\boldsymbol{\mathcal{X}}) \in \mathbb{R}^{d_l}$ is defined by

$$[\boldsymbol{\phi}^{l,r}(\boldsymbol{\mathcal{X}})]_j \equiv \sum_{i_1,\ldots,i_M} \mathcal{X}_{i_1,\ldots,i_M}\delta(j, i_l)\prod_{m\neq l} a_{i_m}^{m,r}, \quad (20)$$

where $\delta(j, i_l)$ is Kronecker's delta, which takes one only if $j = i_l$ zero otherwise. Using this and $\boldsymbol{\mu}^{l,r} \equiv \langle \boldsymbol{a}^{l,r}\rangle$, we have

$$\langle\Delta(y, \boldsymbol{\mathcal{X}})^2\rangle_{\backslash \boldsymbol{a}^{l,r}} = \mathrm{c.} + \boldsymbol{a}^{l,r\top}\langle\boldsymbol{\phi}^{l,r}\boldsymbol{\phi}^{l,r\top}\rangle\boldsymbol{a}^{l,r} - 2\boldsymbol{a}^{l,r\top}\langle\boldsymbol{\phi}^{l,r}\rangle[y - \sum_{r'\neq r}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mu}^{1,r'} \circ \cdots \circ \boldsymbol{\mu}^{M,r'})], \quad (21)$$

where c. is a constant not including the $\boldsymbol{a}^{l,r}$ and $\langle\cdot\rangle$ without subscript denotes the expectation by Q (Eq. (16)). We dropped the subscript $_{\backslash \boldsymbol{a}^{l,r}}$ on the r.h.s. because $\boldsymbol{\phi}^{l,r}$ does not include the $\boldsymbol{a}^{l,r}$.

The VB equation (17) now looks like:

$$\ln q_1^{l,r}(\boldsymbol{a}^{l,r}) = \mathrm{c.} - \frac{1}{2}\boldsymbol{a}^{l,r\top}(\boldsymbol{\Sigma}^{l,r})^{-1}\boldsymbol{a}^{l,r} + \lambda\boldsymbol{a}^{l,r\top}\boldsymbol{\Phi}^{l,r}\boldsymbol{y}_N^{l,r},$$

where

$$\boldsymbol{\Sigma}^{l,r} \equiv \left\{\lambda\sum_{n=1}^{N}\langle\boldsymbol{\phi}^{l,r,(n)}\boldsymbol{\phi}^{l,r,(n)\top}\rangle + \langle b^{l,r}\rangle\mathbf{I}_{d_l}\right\}^{-1} \quad (22)$$

$$\boldsymbol{\Phi}^{l,r} \equiv [\langle\boldsymbol{\phi}^{l,r,(1)}\rangle, \ldots, \langle\boldsymbol{\phi}^{l,r,(N)}\rangle] \quad (23)$$

$$[\boldsymbol{y}_N^{l,r}]_n \equiv y^{(n)} - \sum_{r'\neq r}(\boldsymbol{\mathcal{X}}^{(n)}, \boldsymbol{\mu}^{1,r'} \circ \cdots \circ \boldsymbol{\mu}^{M,r'}), \quad (24)$$

with $\boldsymbol{\phi}^{l,r,(n)}$ being a shorthand notation for $\boldsymbol{\phi}^{l,r}(\boldsymbol{\mathcal{X}}^{(n)})$. Thus we conclude that $q_1^{l,r}(\boldsymbol{a}^{l,r}) = \mathcal{N}(\boldsymbol{a}^{l,r} \mid \boldsymbol{\mu}^{l,r}, \boldsymbol{\Sigma}^{l,r})$ with $\boldsymbol{\Sigma}^{l,r}$ being Eq. (24) and

$$\boldsymbol{\mu}^{l,r} = \lambda\boldsymbol{\Sigma}^{l,r}\boldsymbol{\Phi}^{l,r}\boldsymbol{y}_N^{l,r}. \quad (25)$$

Using $q_1^{l,r}$, we can explicitly compute $\langle\boldsymbol{\phi}^{l,r}\boldsymbol{\phi}^{l,r\top}\rangle$ as

$$[\langle\boldsymbol{\phi}^{l,r}\boldsymbol{\phi}^{l,r\top}\rangle]_{i,j} = \sum_{\mathbf{i},\mathbf{j}}\mathcal{X}_{\mathbf{i}}\mathcal{X}_{\mathbf{j}}\,\delta(i, i_l)\delta(j_l, j)\prod_{m\neq l} S_{i_m,j_m}^{m,r}, \quad (26)$$

$$\mathbf{S}^{m,r} \equiv \langle\boldsymbol{a}^{m,r}\boldsymbol{a}^{m,r\top}\rangle = \boldsymbol{\Sigma}^{m,r} + \boldsymbol{\mu}^{m,r}(\boldsymbol{\mu}^{m,r})^\top \quad (27)$$

where we used the notation of $\mathbf{i} = (i_1, \ldots, i_M)$ etc. Similarly, $\langle\boldsymbol{\phi}^{l,r}\rangle$ is given just by replacing $a_{i_m}^{m,r}$ with $\mu_{i_m}^{m,r}$ in Eq. (23).

The posterior mean (25) has clear similarities with ordinary least squares. For $\boldsymbol{a}^{l,r}$, the vector $\boldsymbol{\phi}^{l,r}$ acts as the predictor and $\boldsymbol{\Phi}$ can be interpreted as the data matrix. Also, given the other $\boldsymbol{\mu}^{l,r'}$ ($r' \neq r$), the vector $\boldsymbol{y}_N^{l,r}$ represents the residual that has not been explained by the intercept and the other variables. Taking this residual as the target variable, Eq. (25) corresponds to the well-known solution of ordinary least squares.

VE step: Posterior for $\{b^{l,r}\}$

Now let us consider the second VE equation (18). Arranging the last terms of L in Eq. (14), we have

$$\ln q_2^{l,r}(b^{l,r}) = \text{c.} + (\alpha^{l,r}-1)\ln b^{l,r} - \beta^{l,r} b^{l,r},$$

$$\alpha^{l,r} \equiv \alpha_0 + \frac{1}{2} d_l \tag{28}$$

$$\beta^{l,r} \equiv \beta_0 + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}^{l,r}) + \|\boldsymbol{\mu}^{l,r}\|_2^2 \tag{29}$$

which leads to the solution

$$q_2^{l,r}(b^{l,r}) = \mathcal{G}(b^{l,r} \mid \alpha^{l,r}, \beta^{l,r}), \tag{30}$$

where $\mathcal{G}$ denotes the Gamma distribution defined in Eq. (13).

Using this we can compute $\langle b^{l,r}\rangle$ in Eq. (22). By the basic property of the gamma distribution,

$$\langle b^{l,r}\rangle = \frac{\alpha^{l,r}}{\beta^{l,r}} = \frac{d_l + 2\alpha_0}{\mathrm{Tr}(\boldsymbol{\Sigma}^{l,r}) + \|\boldsymbol{\mu}^{l,r}\|_2^2 + 2\beta_0}, \tag{31}$$

where we have used Eq. (27) for Eq. (29).

VM step: Point estimation of λ

The next step of the vEM procedure is to point-estimateλ by maximizing the posterior expectation of the log complete likelihood. Formally, our task is

$$\textbf{VM step: } \lambda = \arg\max_{\lambda} \langle L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda)\rangle. \tag{32}$$

To do this, we need an explicit representation of $\langle L(\boldsymbol{\mathcal{A}}, \boldsymbol{b}, \lambda)\rangle$. Again, the most challenging part is to find the expression of $\langle \Delta^2\rangle$. In this case, we do not need to factor out a specific $\boldsymbol{a}^{l,r}$. By simply expanding the square, we have

$$\begin{aligned}
\langle \Delta(y, \boldsymbol{\mathcal{X}})^2\rangle &= y^2 - 2y\sum_r \langle (\mathcal{X}, \boldsymbol{a}^{1,r}\circ\cdots\circ\boldsymbol{a}^{M,r'})\rangle \\
&+ \sum_{r,r'} \langle (\mathcal{X}, \boldsymbol{a}^{1,r}\circ\cdots\circ\boldsymbol{a}^{M,r})(\boldsymbol{a}^{1,r'}\circ\cdots\circ\boldsymbol{a}^{M,r'}, \mathcal{X})\rangle \\
&= \left\{ y - \sum_r (\mathcal{X}, \boldsymbol{\mu}^{1,r}\circ\cdots\circ\boldsymbol{\mu}^{M,r}) \right\}^2 + \sum_r \Gamma^r(\boldsymbol{\mathcal{X}}),
\end{aligned}$$

where we used $\langle a_{i_m}^{m,r} a_{j_m}^{m,r}\rangle = [\mathbf{S}^{m,r}]_{i_m,j_m}$ to define

$$\begin{aligned}
\Gamma^r(\boldsymbol{\mathcal{X}}) \equiv (\mathcal{X} \times_1 \mathbf{S}^{1,r} \times_2 \cdots \times_M \mathbf{S}^{M,r}, \mathcal{X}) \\
- (\mathcal{X}, \boldsymbol{\mu}^{1,r}\circ\cdots\circ\boldsymbol{\mu}^{M,r})^2.
\end{aligned} \tag{33}$$

The condition of optimality for λ is now given by

$$0 = \frac{\partial\langle L\rangle}{\partial\lambda} = \frac{N}{2\lambda} - \frac{1}{2}\sum_{n=1}^N \langle \Delta(y^{(n)}, \boldsymbol{\mathcal{X}}^{(n)})^2\rangle, \tag{34}$$

resulting in

$$\begin{aligned}
\lambda^{-1} = \frac{1}{N}\sum_{n=1}^N \left\{ y^{(n)} - \sum_{r=1}^R (\boldsymbol{\mathcal{X}}^{(n)}, \boldsymbol{\mu}^{1,r}\circ\cdots\circ\boldsymbol{\mu}^{M,r}) \right\}^2 \\
+ \frac{1}{N}\sum_{n=1}^N\sum_{r=1}^R \Gamma^r(\boldsymbol{\mathcal{X}}^{(n)})
\end{aligned} \tag{35}$$

Algorithm 1 Bayesian ALS (BALS) for Tensor Regression.

Input: $\{(y^{(n)}, \boldsymbol{\mathcal{X}}^{(n)})\}_{n=1}^N$, R.
Output: $\{\boldsymbol{\mu}^{l,r}, \boldsymbol{\Sigma}^{l,r}, \alpha^{l,r}, \beta^{l,r}\}$, λ.
Initialize: $\boldsymbol{\mu}^{l,r}$ as random vector and $\langle b\rangle^{l,r}$ as 1 for $\forall(l,r)$.
repeat
  $\lambda^{-1} \leftarrow$ Eq. (35)
  for $l \leftarrow 1, \ldots, M$ **do**
    for $r \leftarrow 1, \ldots, R$ **do**
      $\boldsymbol{\Sigma}^{l,r} \leftarrow \left\{ \lambda \sum_{n=1}^N \langle \boldsymbol{\phi}^{l,r,(n)} \boldsymbol{\phi}^{l,r,(n)\top}\rangle + \langle b^{l,r}\rangle \mathbf{I}_{d_l} \right\}^{-1}$
      $\boldsymbol{\mu}^{l,r} \leftarrow \lambda \boldsymbol{\Sigma}^{l,r} \boldsymbol{\Phi}^{l,r} \boldsymbol{y}_N^{l,r}$
      $\beta^{l,r} \leftarrow \beta_0 + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}^{l,r}) + \|\boldsymbol{\mu}^{l,r}\|_2^2$
    end for
  end for
until convergence

Although the multi-way nature of tensors makes things significantly complicated, this has a clear interpretation. Since $\langle \boldsymbol{\mathcal{A}}\rangle = \sum_{r=1}^R \boldsymbol{\mu}^{1,r}\circ\cdots\circ\boldsymbol{\mu}^{M,r}$, the first term is the same as the standard definition of the variance as the squared deviation from the mean. The second term comes from interactions between different modes.

Algorithm 1 summarizes the entire vEM inference procedure, which we call the *Bayesian ALS* (BALS) for tensor regression, as this algorithm is most naturally viewed as a Bayesian extension of the ALS algorithm (see Proposition 1). Following (Kohn, Smith, and Chan 2001), we fix $\alpha_0 = 1, \beta_0 = 10^{-6}$ so the prior becomes near non-informative. The rank R is virtually the only input parameter to be tuned. For *e.g.* anomaly detection, R can be determined by evaluating AUC (area under curve) of the ROC (receiver operating characteristic) curve for each of $R = 1, 2, 3, \ldots$.

Despite its seeming simplicity, implementing BALS is not necessarily straightforward. For actual implementation, it is advisable to use a few tensor algebraic tricks. It is also sometime useful to use mean-field-like approximations for numerical stabilities, as shown in the next subsection. The total complexity of the algorithm depends on the formulas to use and there is a subtle trade-off among codability, efficiency, numerical stability, and memory usage. We omit the details here for space limitations.

Relationship with classical ALS

Here we discuss the following proposition:

Proposition 1 *The classical ALS solution is the maximum a posteriori (MAP) approximation of the Bayesian ALS.*

To prove this, let $b^{l,r} = \rho\lambda$ for a constant ρ. The MAP solution is the one to maximize the log likelihood with respect to $\{\boldsymbol{a}^{l,r}\}$. By differentiating Eq. (14), we easily get the condition of optimality

$$\sum_{n=1}^N \left\{ y^{(n)} - \sum_{r'=1}^R (\boldsymbol{a}^{l,r'})^\top \boldsymbol{\phi}^{l,r',(n)} \right\} \boldsymbol{\phi}^{l,r,(n)} + \rho \boldsymbol{a}^{l,r} = \mathbf{0},$$

which readily gives an iterative formula:

$$\tilde{\boldsymbol{\Sigma}}^{l,r} \leftarrow \left\{ \sum_{n=1}^{N} \boldsymbol{\phi}^{l,r}(\boldsymbol{\mathcal{X}}^{(n)}) \boldsymbol{\phi}^{l,r}(\boldsymbol{\mathcal{X}}^{(n)})^{\top} + \rho \mathbf{I}_{d_l} \right\}^{-1} \tag{36}$$

$$\boldsymbol{a}^{l,r} \leftarrow \tilde{\boldsymbol{\Sigma}}^{l,r} \boldsymbol{\Phi}^{l,r} \boldsymbol{y}_N^{l,r}, \tag{37}$$

where we used the same notations as the probabilistic counterpart (Eqs. (23) and (24)). To establish the relationship with the traditional notation of ALS, we note that Eq. (20) can be written as

$$\boldsymbol{\phi}^{l,r} = \mathbf{X}_{(l)} (\boldsymbol{a}^{M,r} \otimes \cdots \otimes \boldsymbol{a}^{l+1,r} \otimes \boldsymbol{a}^{l-1,r} \otimes \cdots \otimes \boldsymbol{a}^{1,r}), \tag{38}$$

where $\mathbf{X}_{(l)}$ is the mode-l matricization of $\boldsymbol{\mathcal{X}}$ (Kolda and Bader 2009) and $\otimes$ denotes the Kronecker product. With this identity, we conclude that the MAP solution is the equivalent to the classical ALS with the ℓ_2 regularizer (see, *e.g.* (Zhou, Li, and Zhu 2013) for an explicit expression).

In comparison to the frequentist ALS solution, there are three major advantages of BALS. First, in the Bayesian ALS, the regularization parameter is automatically learned as part of the model. In ALS, ρ has to be cross-validated. Second, BALS can provide a probabilistic output, while the ALS has to resort to extra model assumptions for that. This is a significant limitation in many real applications, especially when applied to change analysis. Third, in BALS, the alternating scheme is derived as a natural consequence of the factorized assumption of Eq. (16), in which the vEM framework provides a clear theoretical foundation of the whole procedure.

Predictive distribution

Using the learned model parameters and the posterior distribution for $\boldsymbol{a}^{l,r}$, we can build a predictive distribution to predict y for an unseen $\boldsymbol{\mathcal{X}}$ as

$$p(y \mid \boldsymbol{x}, \mathrm{D}) = \int \prod_{l,r} \mathrm{d}\boldsymbol{a}^{l,r} q_1^{l,r}(\boldsymbol{a}^{l,r}) \mathcal{N}(y \mid (\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{A}}), \lambda^{-1}).$$

Due to the intermingled form of different modes, the exact integration is not possible despite the seeming linear Gaussian form. However, we can derive an approximated result in the following way. First, pick an arbitrary (l', r'), and use the factored form Eq. (19). By performing integration with respect to $\boldsymbol{a}^{l',r'}$, we have

$$\begin{aligned} &\int \mathrm{d}\boldsymbol{a}^{l',r'} q_1^{l',r'}(\boldsymbol{a}^{l',r'}) \mathcal{N}(y \mid (\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{A}}), \lambda^{-1}) \\ &= \mathcal{N}(y \mid \boldsymbol{\mu}^{l',r'^{\top}} \boldsymbol{\phi}^{l',r'}, \lambda^{-1} + \boldsymbol{\phi}^{l',r'^{\top}} \boldsymbol{\Sigma}^{l',r'} \boldsymbol{\phi}^{l',r'}). \end{aligned} \tag{39}$$

To proceed to a next (l'', r''), we need to factor out the $\boldsymbol{a}^{l'',r''}$ form $\boldsymbol{\phi}^{l',r'}$. The problem is that the variance is a complex function of $\boldsymbol{a}^{l'',r''}$. Here, as an approximation, we replace the variance with $\lambda^{-1} + \mathrm{Tr}(\boldsymbol{\Sigma}^{l',r'} \langle \boldsymbol{\phi}^{l',r'} \boldsymbol{\phi}^{l',r'^{\top}} \rangle)$ and take account of the dependency of $\boldsymbol{a}^{l'',r''}$ only in the mean.

In this way, we obtain the predictive distribution as

$$p(y \mid \boldsymbol{x}, \mathrm{D}) = \mathcal{N}(y \mid \mu(\boldsymbol{\mathcal{X}}), \sigma^2(\boldsymbol{\mathcal{X}})) \tag{40}$$

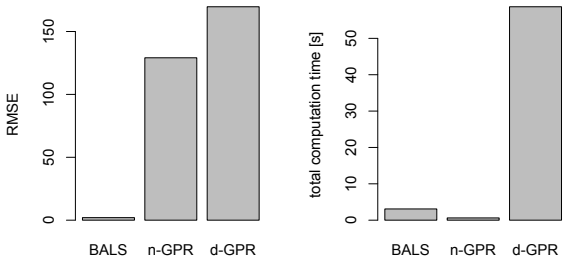

Figure 2: Comparison of the RMSE and the total computation time on average from training through testing.

with

$$\mu(\boldsymbol{\mathcal{X}}) = \eta + \sum_{r=1}^{R} (\boldsymbol{\mathcal{X}}, \boldsymbol{\mu}^{1,r} \circ \cdots \circ \boldsymbol{\mu}^{M,r}), \tag{41}$$

$$\sigma^2(\boldsymbol{\mathcal{X}}) = \lambda^{-1} + \sum_{r=1}^{R} \sum_{l=1}^{M} \mathrm{Tr}\left(\boldsymbol{\Sigma}^{l,r} \langle \boldsymbol{\phi}^{l,r} \boldsymbol{\phi}^{l,r^{\top}} \rangle \right), \tag{42}$$

where η is to offset non-centered testing data. If we denote the sample average of y and $\boldsymbol{\mathcal{X}}$ over *raw* training samples by $\bar{y}$ and $\bar{\boldsymbol{\mathcal{X}}}$, respectively, η is given by

$$\eta \equiv \bar{y} - \sum_{r=1}^{R} (\bar{\boldsymbol{\mathcal{X}}}, \boldsymbol{\mu}^{1,r} \circ \cdots \circ \boldsymbol{\mu}^{M,r}). \tag{43}$$

Here $\langle \boldsymbol{\phi}^{l',r'} \boldsymbol{\phi}^{l',r'^{\top}} \rangle$ is given by Eq. (26). Unlike the GPR-based tensor regression methods (Zhao et al. 2014; Hou, Wang, and Chaib-draa 2015; Suzuki 2015; Kanagawa et al. 2016; Imaizumi and Hayashi 2016), we do not need any heavy computations of CP or Tucker decomposition upon testing.

Experiments

As discussed, the problem of tensorial change analysis is new and existing methods are not directly comparable about the change analysis part. Thus, we focus on 1) demonstrating the practical utility of BALS in computing mode-wise change analysis scores for a real-world application. We also illustrate major features of BALS by 2) comparing with alternatives on metrics such as computational time.

In the present context, whose primary goal is to compute the information-theoretic change analysis score (10), two GPR-based models are relevant to BALS. One is based on the naive Gaussian kernel $\sigma_0 \mathrm{e}^{-\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}'\|_{\mathrm{F}}^2 / \sigma^2}$, which we call n-GPR. The other is based on the state-of-the-art mode-wise kernel decomposition: $\sigma_0 \prod_{l=1}^{M} \mathrm{e}^{-D(\mathbf{X}_{(l)}, \mathbf{X}'_{(l)})/\sigma^2}$, which we call d-GPR. Here $D(\cdot, \cdot)$ a distance function between mode-l matricized tensors (Signoretto et al. 2014). We used R's tensr package for d-GPR. In BALS, we used an approximation $\langle \boldsymbol{\phi}^{l,r} \boldsymbol{\phi}^{l,r^{\top}} \rangle \approx \langle \boldsymbol{\phi}^{l,r} \rangle \langle \boldsymbol{\phi}^{l,r} \rangle^{\top}$ in evaluating λ and $\boldsymbol{\Sigma}^{l,r}$ for numerical stability.

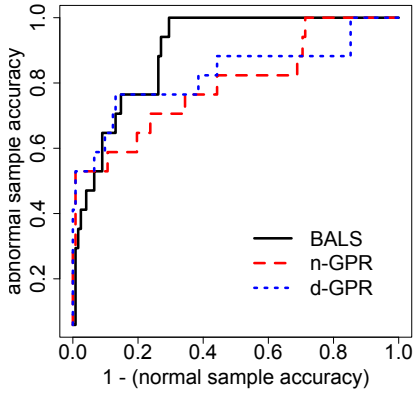

Figure 3: ROC curves for London School data with AUC values 0.96, 0.85, 0.88 for BALS, n-GPR, d-GPR, respectively.

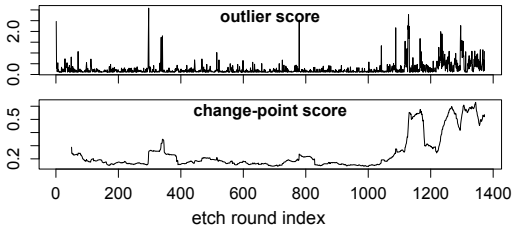

Figure 4: Outlier and change-point scores (with $\tilde{N} = 50$ in Eq.(8)) for the semiconductor etching data.

Synthetic data

To show general features of BALS in comparison to the alternatives, we synthetically generated mode-3 ($M = 3$) tensor data in $(d_1, d_2, d_3) = (10, 8, 5)$ with a given set of the coefficients and randomly generated covariance matrices. For the covariance matrices, we first randomly generated the entries of matrices of the size $\mathbb{R}^{d_1 \times d_1}, \mathbb{R}^{d_2 \times d_2}, \mathbb{R}^{d_3 \times d_3}$ using the standard normal distribution and made them positive definite by replacing the eigenvalues in their eigenvalue decomposition with random positive numbers sampled from $\mathcal{G}(1, 1/2)$. To be fair to n-GPR, which corresponds to the vectorized regression, and to simulate heavy fluctuations in the real-world, we added a t-distributed noise to a vectorized representation and generated 500 samples. The parameters are optimized using 5-fold cross-validation (CV), so the root mean squared error (RMSE) was minimized.

As summarized in Fig. 2, in spite of many outliers due to the t-noise, BALS outperformed the alternatives in RMSE. It is interesting that the GPR-based methods failed to capture the underlying generative model even with the mode-decomposed GPR method. This is mainly because the kernel trick (Bishop 2006) does not guarantee the primal-dual equivalence for tensors with $M \geq 3$. Figure 2 also compares the total computational time on average (for an optimal hyperparameter choice) from training to testing (3.1, 0.6, 58.7 seconds from left to right). n-GPR is the fastest although BALS is comparable to it. Due to extra operations for matricization, d-GPR is much costlier than the others.

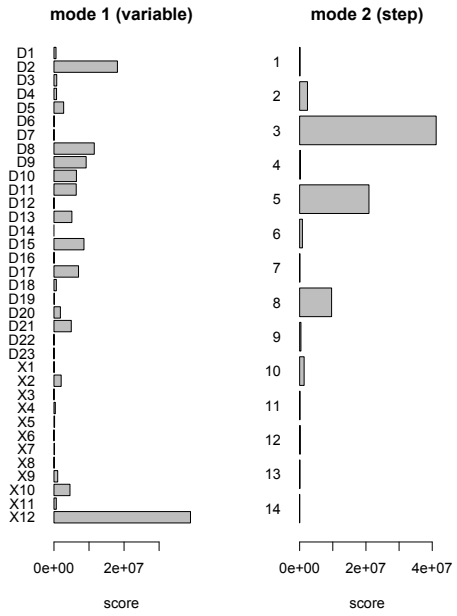

Figure 5: Change analysis score showing the contribution of individual modes and dimensions.

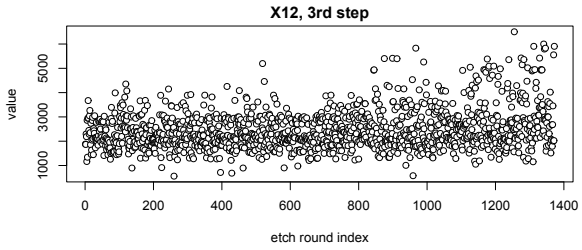

Figure 6: Observed sensor values of $\mathcal{X}_{35,3}$, which corresponds to the most contributing dimension in Fig. 5.

London School data

Next, we tested outlier detection capabilities of BALS using publicly available London School data (Goldstein 1991) to pick unusually well-performing schools in the data cleansing task, in which schools whose median of 'exam.score' is greater than 25 are defined as outliers. We created 139 $\mathbb{R}^{4 \times 5 \times 2}$ tensors by computing '% of FSM,' the number of VR1 students, the number of VR2 students, and the school denomination for each pair of gender and ethnicity. The original eleven ethnicity groups were converted into five groups by merging the smallest groups. In BALS, we picked $R = 7$ that gave the maximum AUC value. Figure 3 compares ROC curves, which shows BALS outperforms the alternative in terms of AUC.

Semiconductor etching diagnosis

Finally, we tested BALS in the semiconductor etching diagnosis task as illustrated in Fig. 1. The training dataset was taken from a "golden period" including about 1 000 pairs

of $(\boldsymbol{\mathcal{X}}, y)$, where $\boldsymbol{\mathcal{X}}$ is a 35 (variables) $\times$ 14 (steps) tensor and y is a scalar performance metric. The testing dataset has 1 372 pairs and includes an excursion event towards the end of the observation period, in which the last 372 samples were assumed to be anomalous. Although human engineers successfully detected the excursion event semi-manually, the true root cause is unknown.

Figure 4 shows the outlier and change-point scores. We picked $R = 4$ that maximized AUC. In the figure, a clear increase of the score is observed towards the end, corresponding to the excursion event, which can be used for early warning. Figures 5 shows the change *analysis* score computed by Eq. (10), which shows a dominant contribution of the variable x_{12} and the third step. Figure 6 shows raw signal of x_{12} in the third step. Very interestingly, this variable has a recognizable increase in the amplitude of fluctuation towards the end, suggesting a potential root cause of the excursion event.

Conclusion

We have proposed a new tensorial change analysis framework based on a newly developed probabilistic tensor regression algorithm, which can be viewed as a probabilistic generalization of the alternating least square algorithm. It can compute change scores for individual tensor modes and dimensions in an information-theoretic fashion, providing useful diagnostic information. To the best of our knowledge, this is the first work of variational Bayesian formulation of probabilistic tensor regression and information-theoretic formulation of tensorial change analysis in the supervised setting. Finally, we successfully applied our method to a change diagnosis task in semiconductor manufacturing.

References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag.

de Araujo, M. R.; Ribeiro, P. M. P.; and Faloutsos, C. 2017. Tensorcast: Forecasting with context using coupled tensors. In *Proc. IEEE International Conference on Data Mining (ICDM)*, 71–80. IEEE.

Dunlavy, D. M.; Kolda, T. G.; and Acar, E. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data* 5(2):10.

Fanaee-T, H., and Gama, J. 2016. Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems* 98:130–147.

Ge, Z.; Song, Z.; Ding, S. X.; and Huang, B. 2017. Data mining and analytics in the process industry: the role of machine learning. *IEEE Access* 5:20590–20616.

Goldsmith, J.; Huang, L.; and Crainiceanu, C. M. 2014. Smooth scalar-on-image regression via spatial Bayesian variable selection. *Journal of Computational and Graphical Statistics* 23(1):46–64.

Goldstein, H. 1991. Multilevel modelling of survey data. *Journal of the Royal Statistical Society. Series D (The Statistician)* 40(2):235–244.

Guhaniyogi, R.; Qamar, S.; and Dunson, D. B. 2017. Bayesian tensor regression. *Journal of Machine Learning Research* 18(79):1–31.

Hou, M.; Wang, Y.; and Chaib-draa, B. 2015. Online local Gaussian process for tensor-variate regression: Application to fast reconstruction of limb movements from brain signal. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5490–5494.

Imaizumi, M., and Hayashi, K. 2016. Doubly decomposing nonparametric tensor regression. In *Proc. International Conference on Machine Learning*, 727–736.

Kanagawa, H.; Suzuki, T.; Kobayashi, H.; Shimizu, N.; and Tagami, Y. 2016. Gaussian process nonparametric tensor estimator and its minimax optimality. In *Proc. International Conference on Machine Learning*, 1632–1641.

Kohn, R.; Smith, M.; and Chan, D. 2001. Nonparametric regression using linear combinations of basis functions. *Statistics and Computing* 11(4):313–322.

Kolda, T. G., and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review* 51(3):455–500.

Rasmussen, C. E., and Williams, C. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Signoretto, M.; Dinh, Q. T.; De Lathauwer, L.; and Suykens, J. A. 2014. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning* 94(3):303–351.

Sun, J.; Tsourakakis, C. E.; Hoke, E.; Faloutsos, C.; and Eliassi-Rad, T. 2008. Two heads better than one: pattern discovery in time-evolving multi-aspect data. *Data Mining and Knowledge Discovery* 17(1):111–128.

Suzuki, T. 2015. Convergence rate of Bayesian tensor estimator and its minimax optimality. In *Proc. International Conference on Machine Learning*, 1273–1282.

Tzikas, D. G.; Likas, A. C.; and Galatsanos, N. P. 2008. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine* 25(6):131–146.

Yamanishi, K.; Takeuchi, J.-I.; Williams, G.; and Milne, P. 2004. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery* 8(3):275–300.

Yu, R., and Liu, Y. 2016. Learning from multiway data: Simple and efficient tensor regression. In *Proc. International Conference on Machine Learning*, 373–381.

Zhao, Q.; Zhou, G.; Zhang, L.; and Cichocki, A. 2014. Tensor-variate Gaussian processes regression and its application to video surveillance. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),*, 1265–1269. IEEE.

Zhou, H.; Li, L.; and Zhu, H. 2013. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association* 108(502):540–552.

Zhu, Y.; He, J.; and Lawrence, R. 2012. Hierarchical modeling with tensor inputs. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 1233–1239.