Exponential Family Tensor Factorization for Missing-Values Prediction and Anomaly Detection

Kohei Hayashi*, Takashi Takenouchi*, Tomohiro Shibata*, Yuki Kamiya†, Daishi Kato†, Kazuo Kunieda†, Keiji Yamada† and Kazushi Ikeda*

*Graduate School of Information Science, Nara Institute of Science and Technology, Japan
†C&C Innovation Research Laboratories, NEC Corporation, Japan

Abstract—In this paper, we study probabilistic modeling of heterogeneously attributed multi-dimensional arrays. The model can manage the heterogeneity by employing an individual exponential-family distribution for each attribute of the tensor array. These entries are connected by latent variables and are shared information across the different attributes. Because a Bayesian inference for our model is intractable, we cast the EM algorithm approximated by using the Laplace method and Gaussian process. This approximation enables us to derive a predictive distribution for missing values in a consistent manner. Simulation experiments show that our method outperforms other methods such as PARAFAC and Tucker decomposition in missing-values prediction for cross-national statistics and is also applicable to discover anomalies in heterogeneous office-logging data.

Keywords—tensor factorization; Bayesian probabilistic model; Gaussian process; data fusion;

I. INTRODUCTION

Multi-dimensional array is a common representation of various data. For example, a multi-dimensional network, having multiple linkages, can be seen as a link types $\times$ nodes $\times$ locations array. Another example is a temporal sequence of measurements from various distributed sensors such as microphones, thermometers, and video cameras, which are represented by an array with dimensions (modes) sensor types $\times$ time. Intuitively, if there exist correlations among such networks or attributes, it is effective to model the entire structure of data rather than to model individual attributes independently. When the data contains missing elements, such an integrative approach will especially be useful for data completion. Moreover, when anomalies are involved in the data, the integrative approach will provide a robust framework for detecting them by incorporating whole information of the data. These ideas, sharing information among multiple data sources, can be seen as a typical data fusion problem [1]; however, the approach is infeasible when the data are heterogeneously attributed, i.e., the statistical properties of the attributes are quite different from each other.

Tensor factorization is a method that transforms the original array into fewer structured parameters, and has various fields of application such as face recognition, social network analysis, and modeling of fluorescence excitation-emission [2]. Tucker decomposition [3] is one of the general forms of standard tensor factorization methods. Similar to other tensor factorizations, Tucker decomposition estimates the parameters by the least square approach, which is equivalent to the maximum likelihood estimate under the isotropic Gaussian noise. From this statistical viewpoint, Tucker’s method is inappropriate to model non-Gaussian observations such as a heterogeneously attributed array with both real and discrete variables.

To overcome this problem, we generalize the likelihood of Tucker decomposition by using exponential-family distributions. The exponential family is a class of distributions that is widely applicable for modeling various types of data. For real-valued observations, we can use the Gaussian distribution. The exponential distribution is especially useful for non-negative values. Discrete variables can be represented by distributions such as Poisson and Bernoulli. To deal with the heterogeneity, we assume an individual exponential-family distribution for each attribute on the array data. In addition, we introduce latent variables that captures noise-corrupted heterogeneous array data into a unified low-dimensional parameter space. Because there is no analytical solution when the exponential family is applied for Bayesian inference in general, we use the expectation-maximization (EM) algorithm for parameter estimation, in which the Gaussian process [4] is employed for approximation. Our approximation scheme provides a computationally efficient algorithm for parameter estimation compared to a naive sampling method, and also allows us to derive a Bayesian predictive distribution for missing elements in a consistent manner. We also show that estimated parameters can be applied for anomaly detection.

The rest of this paper is organized as follows. In section 2, we briefly explain Tucker decomposition and then generalize the model by employing exponential-family distributions. In section 3, we introduce our approximation method with Gaussian Process. In section 4, we show how to apply our model for anomaly detection. In section 5, we review related methods to discuss our study and their relationships, and we show some experimental results with both synthetic and real datasets in section 6. Finally, we discuss some properties of...
the proposed method and conclude our study.

II. PROBABILISTIC MODELING OF HETEROGENEOUS TENSOR DATA

Throughout this paper, we call \( M \)-dimensional arrays \( \text{“M-th order tensor”} \) as special cases, first and second order tensors are equivalent to vectors and matrices, respectively. In the following study, we consider only third order tensors for simplicity. However, the results can easily be generalized to higher order tensors.

A. Tucker Decomposition

Let \( \mathbf{X} \) be a \( D_1 \times D_2 \times D_3 \) observation tensor, which has \( D = D_1 D_2 D_3 \) observation entries. Tucker decomposition provides a way to factorize \( \mathbf{X} \) into a core tensor \( \mathbf{Z} \in \mathbb{R}^{K_1 \times K_2 \times K_3} \) and three factor matrices \( \mathbf{U}^{(m)} \in \mathbb{R}^{D_m \times K_m} \) \((m = 1, 2, 3)\), and represents an \((i, j, k)\)-th element of \( \mathbf{X} \) as

\[
x_{ijk} = \sum_{q=1}^{K_1} \sum_{r=1}^{K_2} \sum_{s=1}^{K_3} z_{qrs} u_{iq}^{(1)} u_{jr}^{(2)} u_{ks}^{(3)} + \epsilon_{ijk} \tag{1}
\]

where \( z_{qrs} \) is a \((q, r, s)\)-th element of \( \mathbf{Z} \), \( u_{iq}^{(m)} \) is an \((i, q)\)-th element of the factor matrix \( \mathbf{U}^{(m)} \), and \( \epsilon_{ijk} \) is an observation noise (Figure 1). The matrix \( \mathbf{U}^{(m)} \) captures the structure of correlation on the \( m \)-th dimension of \( \mathbf{X} \). Tucker decomposition estimates parameters \( \mathbf{Z} \) and \( \{ \mathbf{U}^{(m)} \} \) by minimizing the sum of square errors \( \sum_{ijk} \epsilon_{ijk}^2 \). Higher-order singular value decomposition (HOSVD) [5] is one of the methods to solve Tucker decomposition. In the following, we consider an unfolding (matricizing) of the tensor \( \mathbf{X} \) by reordering its elements into a matrix form, and a resultant matrix keeping the structure with respect to the \( m \)-th mode of \( \mathbf{X} \) is denoted by \( \mathbf{X}^{(m)} \in \mathbb{R}^{D_m \times (D/D_m)} \). For more details, see [2].

HOSVD estimates the factor matrix \( \mathbf{U}^{(m)} \) as the top \( K_m \) left singular vectors of unfolded tensor \( \mathbf{X}^{(m)} \).

For later convenience, we rewrite (1) by a vector and a matrix form. Let \( \tilde{\mathbf{x}} \) be a \( D \)-dimensional vector whose elements are given by these of \( \mathbf{X} \) with arbitrary reordering. This is an analogy of a vectorization of a matrix. Now we can rewrite (1) as

\[
\tilde{\mathbf{x}} = \mathbf{W} \tilde{\mathbf{z}} + \tilde{\mathbf{\epsilon}}, \quad \mathbf{W} = \mathbf{U}^{(3)} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)} \tag{2}
\]

where \( \otimes \) is the Kronecker product of matrices and \( \mathbf{W} \in \mathbb{R}^{D \times K} \) is a linear mapping of the vectorized core tensor \( \tilde{\mathbf{z}} \in \mathbb{R}^{K=K_1 K_2 K_3} \) under a vectorized noise \( \tilde{\mathbf{\epsilon}} \in \mathbb{R}^{D} \). In this form, Tucker decomposition is viewed as a standard linear model in which \( \tilde{\mathbf{x}} \) is a \( K \)-dimensional representation of observation \( \mathbf{x} \) in the linear space spanned by \( D \) bases \( \mathbf{w}_{dijk} = u_{i}^{(3)} \otimes u_{j}^{(2)} \otimes u_{k}^{(1)} \), where \( \mathbf{w}_d \) is the \( d \)-th row vector of \( \mathbf{W} \) and \( u_{i}^{(m)} \) is the \( i \)-th row vector of \( U^{(m)} \). The main difference from the standard linear model is that \( \mathbf{W} \) is constrained by \( \{ U^{(m)} \} \) through the Kronecker products.

By using the unfolding transformation, (1) can also be rewritten as

\[
\mathbf{X}^{(1)} = \mathbf{U}^{(1)} \mathbf{Z}^{(1)} (\mathbf{U}^{(3)} \otimes \mathbf{U}^{(2)})^T + \mathbf{E}^{(1)} \tag{3}
\]

where \( \mathbf{Z}^{(1)} \in \mathbb{R}^{K_1 \times K_2 K_3} \) and \( \mathbf{E}^{(1)} \in \mathbb{R}^{D_1 \times D_2 D_3} \) are unfolded tensors associated with the tensor \( \mathbf{Z} \) and the noise, respectively. Note that we can also rewrite equations about \( \mathbf{X}^{(2)} \) and \( \mathbf{X}^{(3)} \) in the same way as (3). Later we will use this representation to derive the gradients of the expected log-likelihood with respect to \( \{ U^{(m)} \} \) in section III-B.

B. Exponential Family Tensor Factorization

Tucker decomposition estimates the parameters \( \mathbf{U}^{(1)} \), \( \mathbf{U}^{(2)} \), \( \mathbf{U}^{(3)} \), and \( \mathbf{Z} \) by minimizing the sum of square errors \( \epsilon_{ijk} \) in (1). In a probabilistic perspective, we can interpret this estimator as the maximum likelihood solution under the assumption of a spherical Gaussian noise \( \epsilon \); however, this assumption is not appropriate when the data \( \mathbf{X} \) is heterogeneously distributed. To tackle this problem, we generalize the model as

\[
\tilde{\mathbf{x}} | \tilde{\theta} \sim \prod_{d=1}^{D} \text{Expon}_{h_d}(\tilde{x}_d | \tilde{\theta}_d) \tag{4}
\]

where \( \tilde{\theta} \equiv \mathbf{W} \tilde{\mathbf{z}} \) is a \( D \)-dimensional vector called a natural parameter and \( h_d \) is an index specifying an assumed exponential-family distribution of \( \tilde{x}_d \). The distribution function \( \text{Expon}_{h} \) is called the natural exponential family;

\[
\text{Expon}_{h}(x | \theta) \equiv \exp[x \theta - \psi(h)(\theta) + F_{h}(x)] \tag{5}
\]

where the function \( F_{h}(x) \) is called base measure and the smooth function \( \psi(h)(\theta) \) is the log-partition function, i.e.,

\[
\psi(h)(\theta) = \ln \int \exp[x \theta + F_{h}(x)] dx. \tag{6}
\]

Note that when we set all \( \text{Expon}_{h} \) as the isotropic Gaussian, the log-likelihood is equivalent to the loss function of the conventional Tucker decomposition. By differentiating both sides of (6), we observe that the derivative \( \psi''(h) : \theta \mapsto \frac{d^2 \psi(h)}{d\theta^2} |_{\theta > 0} \) is a mapping from the natural parameter \( \theta \) to the conditional expectation of \( x \) on \( \text{Expon}_{h}(x | \theta) \). We can also compute the variance by using the second derivative \( \psi''(h) : \theta \mapsto \frac{d^2 \psi(h)}{d\theta^2} |_{\theta > 0} \).
which implies that $\psi_1$ is convex and then Expon$_h$ is log-concave. We summarize these functions $\psi$ and its derivatives of various exponential family distributions in Table I.

Nevertheless, our model has a strong connection to generalized linear models [6]. For example, if we choose a Bernoulli distribution as Expon$_h(x|\theta)$ for binary data $x \in \{0, 1\}$, $\psi^\prime$ becomes a sigmoid function. In this case our model is equivalent to a logistic regression, where $\mathbf{z}$, $\mathbf{x}$, and $\mathbf{W}$ correspond the input, output, and regression coefficients, respectively.

The key assumption of our model is heterogeneity of the attributes controlled by the index $h$ in (4). Unlike existing factorization models such as PCA, our model can independently choose distributions from exponential family (5) by changing the index $h$ (d = 1, . . . , D). This allows us more flexible modeling of data. Let us consider two $D_2 \times D_3$ matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ for example. We assume the statistical natures of these matrices are quite different, e.g., the domain of $\mathbf{X}_1$ is positive integer and $\mathbf{X}_2$ is binary. A naive approach would be to model $\mathbf{X}_1$ and $\mathbf{X}_2$ independently. In our framework, we combine $\mathbf{X}_1$ and $\mathbf{X}_2$ as a $2 \times D_2 \times D_3$ tensor $\mathbf{X}$ and set the index $h$ to one of Poisson for $x_{1jk}$ and one of Bernoulli for $x_{2jk}$ ($j = 1, . . . , D_2$, $k = 1, . . . , D_3$). When our model is applied to $\mathbf{X}$, $\mathbf{U}^{(1)}$ would extract the correlation between $\mathbf{X}_1$ and $\mathbf{X}_2$ in the low-dimensional parameter space. In this sense, our approach has richer information than the independent modeling of $\mathbf{X}_1$ and $\mathbf{X}_2$.

Note that we need to specify the distributions of attributes before learning the model. Choosing wrong distributions usually declines the performance of the model. Therefore, when statistical properties of the data are completely unknown or the distributions of all attributes are not members of the exponential family, standard methods assuming Gaussian distribution (e.g., PARAFAC) should be more plausible.

In the following, we treat the vector $\mathbf{z}$ as a latent variable associated with $\mathbf{x}$ and assume the standard Gaussian prior $N(0, I)$ for $\mathbf{z}$ by following the idea of generalized linear latent variable models [7]. We also add a spherical Gaussian prior with the precision $\alpha_{\text{m}}$ for each of the factor matrices $\{\mathbf{U}^{(m)}\}$, which is equivalent to the quadratic regularization. Finally, the joint log-likelihood $\mathcal{L}$ is written as

$$
\mathbf{x}^T \theta - \sum_{d=1}^{D} \psi_{h_d}(\theta_d) - \frac{||\mathbf{Z}||^2}{2} - \sum_{m=1}^{M} \frac{\alpha_{\text{m}}}{2} \left| \mathbf{U}^{(m)} \right|^2 + \text{const}.
$$

The first and the second terms correspond to the likelihood (4) with the natural parameter $\theta = \mathbf{Wz}$. The third and the fourth terms are the priors of $\mathbf{z}$ and $\{\mathbf{U}^{(m)}\}$, respectively. We call the model “Exponential family Tensor Factorization” (ETF).

III. Inference Methods

We employ the EM algorithm for parameter estimation; however, there are two difficulties for applying the EM algorithm to our model:

- E-step: The normalization term $\int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ of the posterior is generally intractable because of the non-conjugacy of the prior and the likelihood, except in the case that the given likelihood is fully Gaussian.
- M-step: When we maximize the expected log-likelihood $\mathcal{L} \equiv \mathbb{E}_\theta[\mathcal{L}|\mathbf{x}]$ with respect to the parameters $\{\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(M)}\}$, the main difficulty for evaluating $\mathcal{L}$ resides in obtaining the expectation of $\psi$. Because the function $\psi$ is nonlinear, its expectation by the posterior is generally intractable.

To tackle these problems, we propose a new framework for approximation of the EM algorithm with combining the two techniques: the Laplace approximation and the Gaussian process (GP). The Laplace approximation gives a posterior by the Gaussian, and we can analytically compute the gradient of the expected log-likelihood with approximation of $\psi$ and $\psi'$ by GP that allows a gradient-based optimization in M-step. Furthermore, we can also derive the mean of the Bayesian predictive distribution in a consistent manner.

A. Posterior Inference by Laplace’s Method

First we approximate the posterior $p(\mathbf{z}|\mathbf{x})$ by Gaussian $q(\mathbf{z}) \equiv N(\mathbf{z}|\mathbf{z}_0, \Sigma_0)$ with the Laplace approximation where $\mathbf{z}_0$ is maximum a posteriori (MAP), i.e., the mode of the posterior distribution of $\mathbf{z}$, and $\Sigma_0$ is the negative inverse of the Hessian at $\mathbf{z}_0$. We can use gradient-based method to find $\mathbf{z}_0$ with the following gradient and Hessian,

$$
\frac{\partial \mathcal{L}}{\partial \mathbf{z}} = \mathbf{W}^T(\mathbf{z} - \tilde{\psi}^\prime) - \mathbf{z}, \quad \frac{\partial^2 \mathcal{L}}{\partial \mathbf{z} \partial \mathbf{z}^T} = -\mathbf{W}^T \tilde{\psi}'' \mathbf{W} - \mathbf{I},
$$

where we define $\tilde{\psi}^\prime \equiv (\psi_{h_1}(\theta_1), \ldots, \psi_{h_D}(\theta_D))$ and $\tilde{\psi}'' \equiv \text{diag}(\tilde{\psi}'')$. Note that the negative Hessian is positive definite, thus, given $\mathbf{W}$, we can find the global maximum.

B. Approximation of Expectation with GP

Next, we obtain the marginal-MAP estimate of $\mathbf{U}^{(m)}$. We consider the approximated expected log-likelihood

$$
\tilde{\mathcal{L}} \equiv \mathbb{E}_q[\mathcal{L}(\mathbf{z})] = \int \mathcal{L}(\mathbf{z}) N(\mathbf{z}|\mathbf{z}_0, \Sigma_0) d\mathbf{z},
$$

and calculate the gradients of $\tilde{\mathcal{L}}$ using the notation (3). Here we introduce the unfolded core tensor $\mathbf{Z}^{(1)} \in \mathbb{R}^{K_1 \times K_2 K_3}$ and natural parameter

$$
\Theta^{(1)} = \mathbf{U}^{(1)} \mathbf{Z}^{(1)} (\mathbf{U}^{(3)} \otimes \mathbf{U}^{(2)})^T \in \mathbb{R}^{D_1 \times D_2 D_3}.
$$

<table>
<thead>
<tr>
<th>DISTRIBUTION</th>
<th>$\psi(\theta)$</th>
<th>$\psi'(\theta)$</th>
<th>$\psi''(\theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAUSSIAN</td>
<td>$\frac{1}{2} \theta^2$</td>
<td>$\theta$</td>
<td>1</td>
</tr>
<tr>
<td>BERNOUlli</td>
<td>$\ln(1 + e^\theta)$</td>
<td>$\sigma(\theta)$</td>
<td>$\sigma(\theta)(1 - \sigma(\theta))$</td>
</tr>
<tr>
<td>POISSON</td>
<td>$e^\theta$</td>
<td>$e^\theta$</td>
<td>$e^\theta$</td>
</tr>
<tr>
<td>EXPONENTIAL</td>
<td>$-\ln(-\theta)$</td>
<td>$-\frac{1}{\theta}$</td>
<td>$\frac{1}{\theta^2}$</td>
</tr>
</tbody>
</table>
By denoting $U_{(1)}^{−1}$ as the pseudo inverse of $U^{(1)}$, we obtain the gradient of expected log-likelihood (7) as

$$\frac{\partial \mathcal{L}}{\partial U^{(1)}} = E_q[(X^{(1)} - \Psi^{(1)})(U^{(3)} \otimes U^{(2)})(Z^{(1)})^T] - \alpha_1 U^{(1)}$$

$$= X^{(1)}(A^{(1)})^T - E_q[\Psi^{(1)}(\Theta^{(1)})^T](U^{(1)})^T - \alpha_1 U^{(1)}$$

where $\Psi^{(1)}$ is the unfolded tensor of $\psi'$ and $A^{(1)} \equiv E_q[Z^{(1)}](U^{(3)} \otimes U^{(2)})^T$. Here we assume that differential and integral operators are commutative. As previously pointed out, the calculation of the expectation of $E_q[\Psi^{(1)}(\Theta^{(1)})^T]$ in the gradient is generally intractable because of the non-linearity of the function $\psi$.

To approximate the expectation, we apply useful properties about the integral and the derivative of GP. We denote $\dot{\psi}(\theta)$ as an approximation of $\psi(\theta)$ by GP whose covariance function is a Gaussian kernel. First, if $\theta$ is a Gaussian random variable, the expectation $E_\theta[\psi(\theta)\theta^n]$ for arbitrary $n \in \mathbb{N}$ can be solved analytically by using (11) in Appendix A. Second, $\partial^n\psi/\partial \theta^n$, the $n$-th order derivative of $\psi$, can directly be approximated by using $\dot{\psi}$ as another GP.

For further details about these properties, see Appendix.

Since we approximate the posterior of $\dot{z}$ as Gaussian in previous section, the posterior of $\dot{\theta} = \dot{W}\dot{z}$ is, if we consider that $\dot{W}$ is not a random variable, also a Gaussian. Then we can compute the expectation $E_q[\psi^{(1)}(\Theta^{(1)})^T]$ by combining the result of the expectation of the kernel (11a) and the derivative of GP (13). Note that we can also calculate the gradients with respect to $U^{(2)}$ or $U^{(3)}$ in the same manner. We use an alternating optimization approach for updates of $\{U^{(m)}\}_{m = 1, 2, 3}$, i.e., maximizing $\mathcal{L}$ with respect to $U^{(m)}$ with fixed $\{U^{(n)}\}_{n \neq m}$ iteratively by changing the index $m$. We use a quasi-Newton method to find the (local) optimums with respect to $\{U^{(m)}\}$.

C. Mean of Predictive Distribution

After the convergence of the EM iterations, we have an estimated natural parameter $\hat{\theta}_0$ under the observed entries $\mathcal{D}$, and we can predict missing entries $\tilde{x}_d$ by the corresponding means of the predictive distribution $E[\tilde{x}_d \mid \mathcal{D}]$. The marginalization by the posterior of the model parameter $\tilde{z}$, which requires in the computation of Bayesian predictive distribution, is intractable. In our framework, however, $\psi'(\hat{\theta}_{0d})$ is a conditional mean of $\tilde{x}_d$ given $\hat{\theta}_{0d}$, thus we can approximate the mean of the predictive distribution by marginalizing $\psi'(\hat{\theta}_{0d})$ by the posterior of $\hat{z}_0$.

$$E[\tilde{x}_d \mid \mathcal{D}] = \int \tilde{x}_d \int p(\tilde{x}_d \mid \tilde{z}, w_d)p(\tilde{z} \mid \mathcal{D})d\tilde{z}d\tilde{x}_d = \int \psi'(\hat{\theta}_{0d})p(\tilde{z} \mid \mathcal{D})d\tilde{z} \simeq E_q[\psi'(\hat{\theta}_{0d})]. \quad (8)$$

Again, the expectation $E_q[\psi'(\hat{\theta}_{0d})]$ can be solved analytically by using the properties of GP. We can also derive the variance or any other higher-order moments of the predictive distribution, but we omit them here for lack of space.

D. Detail of Implementation and Computational Complexity

Since we assume that the variance of Gaussian is 1 (section II-B), we standardize the part of the data distributed by Gaussian, i.e., scaling each part by the standard deviation of the data samples before parameters estimation. We initialize $\{U^{(m)}\}$ by using HOSVD before starting the EM algorithm. We show a pseudo code of our inference algorithm for an $M$-th order data tensor $X$ as below. Note that C, s, y, and $\gamma^2$ appeared in the pseudo code are a kernel matrix, inputs, outputs, and a hyper parameter of GP described in Appendix.

Input: $X$, $\gamma^2$, $\alpha$
Standardize Gaussian attributes in $X$
$[U^{(1)}, \ldots, U^{(M)}] = \text{HOSVD}(X)$

repeat
W = $U^{(M)} \otimes \ldots \otimes U^{(2)} \otimes U^{(1)}$

// E-step
$\tilde{z}_0 = \text{argmax} \mathcal{L}(\tilde{z})$
$\Psi'_0 = \text{diag}(\psi'(w_1^T\tilde{z}_0), \ldots, \psi'(w_D^T\tilde{z}_0))$
$\Sigma_0 = (W^T \Psi'_0 (W + I))^{-1}$

// M-step
Sample $\{s, s'\}$ and $\{y_h, y'_h\}$
Compute $C^{-1}$ and $\beta_h = C^{-1}(y_h^T, y'_h)^T$
for $m = 1, \ldots, M$
$U^{(m)} = \text{argmax}_{U^{(m)}} \mathcal{L}(U^{(1)}, \ldots, U^{(M)})$
end for
until Convergence
Return $E_q[\psi']$

For GP approximation in the M-step, we need to determine the training inputs $s$. Note that the expectation can factorize to the product of independent expectation, i.e., $\int \psi(\theta_0)dp(\theta_0) = \prod_{d=1}^D \int \psi_{s,d}(\hat{\theta}_{0d})dp(\hat{\theta}_{0d})$ since we assume that each element of $X$ is independently distributed in (4). Thus, we can share the inputs for each dimension of $\psi(\theta_0)$. Also, it is important to cover an area by the training inputs in which the posterior is dense and/or functions $\psi$ and $\psi'$ take a large value. Thus, we randomly choose $N$ one-dimensional inputs $\theta_n (n = 1, \ldots, N)$ from each dimension of the posterior mean $\theta_0 \equiv W\tilde{z}_0$ with the weights $\psi(\tilde{\theta}_0)$.

In an area in which the inputs are sparse, on the other hand, the mean of GP is close to zero because of the property of the GP. This property would be problematic when we apply the gradient-based optimization, since the cost function $\mathcal{L}(\theta)$ diverges when $\theta \to \infty$. To avoid this problem, we use a barrier function instead of the mean of GP as the cost function $\mathcal{L}(\theta)$.

The dominating complexity in E-step is calculation of the covariance (inverse of the Hessian) in the Laplace approximation, which needs $O(K^3)$ where $K = \text{dim}(\tilde{z})$. In the GP approximation of M-step, the inverse of kernel matrix $C$ defined in (12) needs $O((N+N')^3)$, where $N$ and $N'$ are the
number of inputs of normal and derivative observation in GP. We only need this most expensive computation once during the EM iteration. We also need $\mathcal{O}(D(N + N')^2) + \mathcal{O}(K^3)$ times computation for matrix multiplication in the gradient of expected log-likelihood in the maximization procedure.

IV. ANOMALY DETECTION

As described in section II-A, the factor matrix $U^{(m)}$ can be seen as low-dimensional features of the $m$-th dimension (mode) of the observation tensor. If we estimate parameters from the tensor which contains anomalous values, the corresponding parts of the factor matrices will be captured as outliers compared to regular parts. By using the factor matrices as inputs of outlier detection, we will find high-impact anomalies affecting to the whole dataset and can achieve more robust detection than the independent modeling approaches. In addition, the anomalous values in the heterogeneous tensor would be represented as distinctive outliers on the feature space of ETF rather than pTucker, since ETF can naturally extract the regular parts of the data under the appropriate assumption of exponential-family distributions.

One simplest way to detect such outliers is to use a distance between inputs. We employ the definition by Knorr et al. [8]: “an object $O$ in a dataset $T$ is a DB$(p, D)$ outlier if at least fraction $p$ of the objects in $T$ lies greater than distance $D$ from $O$.”

V. RELATED WORKS

A. Tensor factorization and Bayesian extension

PARAFAC [9] is a tensor factorization method that can be seen as a special case of the Tucker method whose core tensor shares the number of dimensions $(K_1 = K_2 = \cdots = K_M)$ and is restricted to diagonal. PARAFAC is applied to the problems such as chemoinformatics [2]. Pairwise interaction tensor factorization (PITF) [10] is another tensor factorization method for tag recommendation problem implemented by the stochastic gradient. Like PARAFAC, the core tensor of PITF is restricted to diagonal that improves its runtime to $O(K_m)$.

There also exist Bayesian extensions of tensor factorization methods. Shashua and Hazan [11] studied the PARAFAC model under the non-negativity constraint with latent variables. To our knowledge, it is the first tensor factorization method that employed the EM algorithm for parameter inference. Chu and Ghahramani [12] proposed a probabilistic extension of the Tucker method, known as pTucker. pTucker can be seen as a special case of ETF whose attributes are specified solely by Gaussian distributions, and the posterior distribution of pTucker is analytically obtained. In fact, the parameter of pTucker can be estimated by direct maximization of the marginal likelihood, which is faster than the EM algorithm.

B. Non-Gaussian matrix and tensor factorization

In several matrix and tensor factorization studies, non-Gaussian observations have been dealt with. Collins et al. [13] proposed exponential family PCA (EPCA), which generalizes the likelihood of probabilistic PCA to the exponential family with no prior for the latent variable. Its parameters are estimated by the minimization of Bregman divergence, which corresponds to the negative log-likelihood of exponential family distribution. A fully Bayesian extension of EPCA with Markov chain Monte Carlo was also proposed [14]. Hayashi et al. [15] extended EPCA for modeling time-evolving matrices. The study employed the Laplace approximation to estimate MAP values of factor matrices on a linear dynamical system.

Yu and Tresp [16] approached a heterogeneous data fusion problem by a probabilistic model for data visualization, which can be viewed as an EPCA with its likelihood restricted to Gaussian or Bernoulli distributions, and introduced a Gaussian latent variable. Wedel and Kamakura [17] proposed a similar model that also generalized the prior distribution of latent variable to exponential family. Several studies on tensor factorization have approached the data fusion problem, using Tucker decomposition [18], for example, or PARAFAC [19] for homogeneous data. To our knowledge, ETF is the first tensor factorization model that allows integration of heterogeneous data.

C. Approximation framework for expectation of log-partition function

Although we use GP for the approximation of expected log-likelihood, there exists other approximation methods. Both piecewise linear and quadratic splines approximation of $\psi$ give the expected log-likelihood $\hat{L}$ as closed form with a Gaussian posterior. The splines would be useful for large-scale data because it saves the computational cost. However, GP approximates $\hat{L}$ more accurately than the simple splines; GP can jointly treat the information of both $\psi$ and its derivative $\psi'$ together, while the splines need to approximate them separately.

Blei and McAuliffe [20] used the delta method to obtain $\hat{L}$. In our model, zeroth-order and first-order delta methods are equivalent to the first-order and the second-order Taylor expansions of $\hat{L}$ with respect to $\bar{z}$ around $\bar{z}_0$. Although the zeroth-order delta method drastically reduces the computational cost for parameter estimation, the accuracy of the approximation is not as good as GP in some cases (see section VI-A). For the first-order delta method, the convergence of the EM algorithm becomes highly unstable.

Seeger and Nickisch [21] proposed a variational lower bound of log-partition function, $\psi$ in our setting, of the exponential family. Unfortunately, this lower bound cannot apply to the Poisson distributions of ETF due to the natural parametrization (4).
VI. EXPERIMENTS

We examined performance of the proposed method by comparing it with existing methods using both synthetic and real datasets. We prepared PARAFAC, Tucker, and pTucker for comparison. We used the N-way Toolbox for Matlab [22] for implementation of PARAFAC and Tucker decomposition. We performed the experiments on an Intel® XEON® CPU 2.40GHz. As measurements of the accuracy of missing-value prediction, we used the root mean square error (RMSE), the mean absolute error (MAE), and the area under the ROC curve (AUC) for real-, count-, and binary-valued datasets, respectively. Lower RMSE or MAE means the estimate is more accurate, and the higher AUC value has the same meaning. We fixed the hyper parameters of our method as $\gamma^2 = 1$ and $\alpha_m = 1$ ($m = 1, 2, 3$) throughout the experiments. We also fixed the regularization coefficients of other methods as 1. In our approximation framework with the GP, we generated training inputs $s$ as described in section III-D and set inputs $s'$ for derivative predictions at the same values.

A. Synthetic Data – Comparison of Approximation Methods

We considered $9 \times 9 \times 9$ observation tensor $X$ to verify the validity of our approximation scheme. The true values of parameters $Z$, $U^{(1)}$, $U^{(2)}$, and $U^{(3)}$ were randomly drawn from their prior distributions in which the ranks of bases $(K_1, K_2, K_3)$ were set to $(2, 3, 4)$. We assumed that elements $\{x_{ijk} \mid j, k = 1, \ldots, 9\}$ of $X$ were distributed by Gaussian for $i = 1, 2, 3$, Poisson for $i = 4, 5, 6$, and Bernoulli for $i = 7, 8, 9$, respectively. Then we generated $X$ and randomly chose 50% of its elements for training of each method. The rest were used to investigate the performance for the missing-values prediction task. We generated 50 datasets with different random seeds and examined the average performance of each method over the 50 trials. We compared the proposed approximation by the GP with the zeroth delta method (Delta) described in section V-C and a simple Monte Carlo (MC) approach for the approximation of the expectation, e.g., $E[\psi(\theta)] \approx \frac{1}{N} \sum_{n=1}^{N} \theta_n \psi(\theta_n)$ where the training inputs $\{\theta_n\}$ were drawn from the Gaussian-approximated posterior distribution $p(\theta)$. We set the sample size $N$ to 100, 1000, 10000 for MC and 100, 1000 for our method, respectively. Note that the approximation errors of both MC and GP converge to 0 when $N \to \infty$. We set the dimension of the core tensor $\{K_m\}$ at the true value.

The RMSEs between the estimated and true natural parameters are shown in Figure 2. We observe that the GP approximation method with 100 samples is more accurate than MC with 10000 samples, and also the computational cost (indicated by the bar chart) of GP was roughly 8 times faster than MC with 10000 samples. Next we investigated the error for the missing-values prediction. The result is summarized in Table II, which shows that ETF with GP outperformed other methods.

B. Cross-National Statistics – Missing-Values Prediction

We prepared a 35 (countries) $\times$ 122 (countries) $\times$ 6 (statistics) tensor data $X$ from six 35 $\times$ 122 matrices $X_1, \ldots, X_6$, whose component represents a statistic between the 35 European Union countries and the other 122 countries. Each statistic is summarized in Table III. Note that more than half of the entries of $X$ were originally missing. We evenly divided observed entries into the training and test data for evaluation of the accuracy for missing-values prediction, and repeated this procedure 20 times with different random seeds. To evaluate the effect of the integrative modeling, the Generalized2 Linear2 Model (G2L2M) [23], a generalization of the singular value decomposition with an exponential family, was individually applied for each matrix $X_1, \ldots, X_6$ and was compared with the proposed method. Because the data took quite large values, we standardized the data for the Gaussian-based methods (PARAFAC, Tucker, and the Gaussian model of G2L2M). Note that the test errors of missing values were measured in the original metrics. For the proposed method and G2L2M, we assumed that the datasets $\{X_i\}$ were distributed by Poisson for $i = 1, 2, 3, 4,$
Table IV
DESCRIPTION OF EACH CROSS-NATIONAL STATISTIC. DATA ABOUT X1, . . . , X5 ARE THE ANNUAL STATISTICS FROM 2007 PUBLISHED BY EUROSTAT (HTTP://EE. EUROP. EU/EUROSTAT).

<table>
<thead>
<tr>
<th>STATISTIC</th>
<th>TYPE</th>
<th>MISSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>AIR PASSENGER</td>
<td>COUNT 65.5%</td>
</tr>
<tr>
<td>X2</td>
<td>AIR FREIGHT &amp; MAIL</td>
<td>COUNT 69.6%</td>
</tr>
<tr>
<td>X3</td>
<td>RAILWAY PASSENGER</td>
<td>COUNT 51.6%</td>
</tr>
<tr>
<td>X4</td>
<td>IMMIGRATION</td>
<td>COUNT 91.2%</td>
</tr>
<tr>
<td>X5</td>
<td>INVESTMENT FLOW</td>
<td>REAL 27.3%</td>
</tr>
<tr>
<td>X6</td>
<td>COLONIAL RELATIONSHIP</td>
<td>BINARY 0%</td>
</tr>
</tbody>
</table>

Table V
DESCRIPTION OF EACH MEASUREMENT OF THE OFFICE-LOGGING DATA.

<table>
<thead>
<tr>
<th>MEASUREMENT</th>
<th>TYPE</th>
<th>MISSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td># OF SENT EMAILS</td>
<td>COUNT 0.0%</td>
</tr>
<tr>
<td>X2</td>
<td># OF RECEIVED EMAILS</td>
<td>COUNT 0.0%</td>
</tr>
<tr>
<td>X3</td>
<td># OF TYPED KEYS</td>
<td>COUNT 0.0%</td>
</tr>
<tr>
<td>X4</td>
<td>X COORDINATE</td>
<td>REAL 4.8%</td>
</tr>
<tr>
<td>X5</td>
<td>Y COORDINATE</td>
<td>REAL 4.8%</td>
</tr>
<tr>
<td>X6</td>
<td>MOVEMENT DISTANCE</td>
<td>NON-NEGATIVE 4.8%</td>
</tr>
</tbody>
</table>

Gaussian for i = 5, and Bernoulli for i = 6, respectively. We used 200 samples for the approximation with GP. Although we could arbitrarily set values K1, K2 and K3, which are the dimensions of the core tensor, we make the restriction K1 = K2 = K3 to compare with the PARAFAC. In G2L2M, however, we chose the best result in test errors by tuning the rank of the parameters for fair comparison. Note that G2L2M, Tucker, and PARAFAC took less than 1 minute in CPU run time; pTucker and ETF were approximately 6 and 24 hours, respectively.

Results are shown in Table IV. For the fixed dimensionality K_m of the core tensor, our method consistently outperformed the other tensor factorization methods except for a case of the binary dataset at K_m = 4. Furthermore, for both count and binary datasets, our method achieved the best estimate, which implies validity of employing the exponential family for the heterogeneous array. In particular for the count datasets X1, . . . , X4, the integrative approaches drastically improved the accuracy of the prediction compared with the G2L2M. This is because the missing rates of the count dataset are high (see Table 3). While the G2L2M was substantially influenced by the missing data, the integrative approaches successfully completed the missing data by utilizing information of correlation between attributes.

C. Office-logging data – Anomaly Detection

In this experiment, we used six temporal sequences of different types of sensor measurements which have been recorded human behaviors in C&C Innovation Research Laboratories (CCIL), NEC Corporation. Each sequence contains the measurement for 20 members, recorded in daily working hours (9 a.m. – 5 p.m.) from September 30 to December 28, 2009. We aggregated the sequence per hour and we got 20 × 255 matrices X1, . . . , X6 for each measurement. We summarized the data matrices in Table V. Finally we integrated them into a 20 (members) × 255 (hours) × 6 (measurements) tensor in the same way of the previous experiment. We used the same setting of the previous experiment except the assumption of distributions; we assumed that the datasets {Xi} were distributed by Poisson for i = 1, 2, 3, 6, and Gaussian for i = 4, 5, respectively. Results of missing-values prediction are shown in Table VI. Similarly to the previous experiment, ETF
outliers by the two methods are distinctive. The detected
in Table VII. The results show that the tendencies of detected
example in [8], we set
comparison. To obtain top six outliers for each method with
unexpected events in CCIL from the sensor measurements,
from the estimated factor matrix. The aim here is to find
outperformed for the test error for the count-valued data but
not for the real-valued one.
We also investigated an efficiency for anomaly detection
from the estimated factor matrix. The aim here is to find
unexpected events in CCIL from the sensor measurements,
and we examined $DB(p, D)$ of ETF and pTucker for a
comparison. To obtain top six outliers for each method with
fixing the parameters $p = 0.995$, which we followed the
example in [8], we set $D$ as 0.075 and 0.18 for ETF and
pTucker, respectively. We summarized the detected outliers
in Table VII. The results show that the tendencies of detected
outliers by the two methods are distinctive. The detected
outliers of ETF, Dec 21 and 22 in Table VII, correspond
to really irregular events which happened only once in this
month, while the ones of pTucker do not contain such
irregular events.

VII. CONCLUSION AND DISCUSSION

In this paper, we proposed an integration method to model
heterogeneously attributed array data by exponential family
tensor factorization. We solved an intractable calculation
in the EM algorithm by approximating the log-partition
function $\psi$ by the GP and achieved efficient parameter
inference. With this approximation, we can also obtain the
mean of the predictive distribution in the closed form (8).
The experimental results clearly showed the feasibility of
our method except for the real-valued datasets.

In both experiments of cross-national statistics and office-
logging data, the test error of ETF for the real-valued
attributes is larger when compared to G$^2$L$^2$M. One plausible
reason is the assumption that the variance parameter of
Gaussian distributions is 1. Although we standardize the
part of the data which obeys Gaussian distributions, this
is not equivalent to modeling a variance parameter in the
likelihood (4) and estimate it.

Although we mainly focus on capturing the heterogeneity
of data tensors, and we also choose the approximation
 technique for parameter inference based on accuracy, scal-
ability of algorithms is also important. In order to apply
our method to large-scale datasets such as EEG and fMRI,
Drastic change is needed (in the model and/or the algorithm).

As described in section III, we approximate the posterior
by a Gaussian density. Due to the log-concavity of the
likelihood and the prior, the posterior is also log-concave
(unimodal), and the Gaussian approximation is reasonable.
The Laplace approximation allows us to implement an
algorithm which is faster than the numerical approach based
on accuracy, scalability of algorithms is also important. In order to apply
our method to large-scale datasets such as EEG and fMRI,
drastic change is needed (in the model and/or the algorithm).

For example, tensor-CUR decomposition [24] estimates the
factor matrices by sampling from a corresponding mode of
data tensor. Stochastic gradient optimization [25] is also a
good solution for the implementation of a scalable algorithm,
whereas its extension for Bayesian inference is not trivial.

As described in section III, we approximate the posterior
by a Gaussian density. Due to the log-concavity of the
likelihood and the prior, the posterior is also log-concave
(unimodal), and the Gaussian approximation is reasonable.

The Laplace approximation allows us to implement an
algorithm which is faster than the numerical approach based
on the MC sampling. On the other hand, we lose the
information about the higher order moments by using the
Gaussian approximation. This drawback can be solved by
approximating the posterior with a skew Gaussian distribu-
tion rather than the regular Gaussian. The skew Gaussian
can handle the skewness, i.e., the third order central moment,
and the integral (11) with the skew Gaussian can still be
analytically solved [26].
ACKNOWLEDGMENT
The authors would like to thank Masayoshi Nakamura for formatting and preprocessing the cross-national statistics dataset for the experiment in section VI-B.

REFERENCES

APPENDIX
Let us consider approximating \( \psi_h \) by the GP. Because \( \psi_h \) is a univariate function, we only consider the case in which the input is unidimensional for simplicity. First we generate \( N \) samples \( s = \{ \theta_1, \ldots, \theta_N \} \) from arbitrary distribution, which is suitable for approximating \( E[\psi_h] \). Then GP prior gives the joint distribution over outputs \( y_h = \{ \psi_h(\theta_1), \ldots, \psi_h(\theta_N) \} \), which is the Gaussian distribution with zero-mean and a covariance matrix \( K(s, s) \). Note that, unlike the normal regression settings, we know the functional form of \( \psi_h(\theta) \), e.g., \( e^\theta \) for Poisson distribution,
and the observation $y_h$ is noise-free. The $(qr)$-th element of the covariance $K$, $k_{qr}$, takes its value by a covariance function $k(\theta_q, \theta_r)$. For convenience we employ a Gaussian kernel as the covariance function:

$$k(\theta_q, \theta_r; \gamma^2) = \exp\left(-\frac{1}{2\gamma^2}||\theta_q - \theta_r||^2\right).$$ (9)

The hyperparameter $\gamma^2$ controls the smoothness of outputs. Using the Bayes theorem, the predictive distribution of $\psi_{h*} \equiv \psi_h(\theta_q)$ at a new input $\theta_q$ with training data $D_h \equiv \{s, y_h\}$ is given by a univariate Gaussian with the mean $m_h(\theta_q)$ and the variance $\nu^2_h(\theta_q)$:

$$m_h(\theta_q) = b_h^T k_s, \quad \nu^2_h(\theta_q) = k_{ss} - k_s^T K^{-1} k_s,$$

where $k_{ss} = k(\theta_s, \theta_s)$, $k_s = (k_{s1}, \ldots, k_{sN})$ and $b_h \equiv K^{-1} y_h$.

**A. Marginalization of GP with Gaussian Density**

Now we consider the case where $\theta_q$ is not a constant but a Gaussian-distributed random variable with a mean $\mu_s$ and a variance $\sigma^2_s$. Thanks to employing the Gaussian kernel as covariance function, we can analytically obtain the predictive mean $\bar{m}_{hs}$ and variance $\bar{\nu}^2_{hs}$ at the uncertain input $\theta_q$ [27], [28], where

$$\bar{m}_{hs} = \int \psi_h p(\psi_h | D_h, \theta_s) p(\theta_s | \mu_s, \sigma^2_s) d\psi_h d\theta_s = E_{\theta_s}[m_h(\theta_s)] = b_h^T E_{\theta_s}[k_s],$$ (10a)

$$\bar{\nu}^2_{hs} = \int (\psi_h - \bar{m}_{hs})^2 p(\psi_h | D_h, \theta_s) p(\theta_s | \mu_s, \sigma^2_s) d\psi_h d\theta_s = b_h^T E_{\theta_s}[k_s k_s^T] b_h + 1 - \text{tr}(K^{-1} E_{\theta_s}[k_s k_s^T]) - \bar{\nu}^2_{hs}.$$ (10b)

Since both functions $p(\theta_s)$ and $k_s$ take squared exponential forms with respect to $\theta_s$, we can obtain the exact first and second moment of $k_s$. Additionally, we can introduce the (finite) polynomial form $\pi(\theta_s) \equiv \sum_i a_i \theta_i^i$ into the inside of the expectation. The solutions are given by

$$E_{\theta_s}[\pi(\theta_s) \kappa_s] = \sum_i \frac{\partial^2}{\partial \sigma^2_i} \bar{\pi}_s k_{\mu, \theta_i; \gamma^2 + \sigma^2_s},$$ (11a)

$$E_{\theta_s}[\pi(\theta_s) \kappa_{sr}] = \sum_i \frac{\partial^2}{\partial \sigma^2_i} \bar{\pi}_s k_{\mu, \theta_i; \gamma^2 + \sigma^2_s}$$

where $\bar{\pi}_s = a_s E_{\theta_s}[\theta]^s$ and $\bar{\pi}_{sr} = a_s E_{\theta_s}[\theta^s]$. The parameters of these Gaussian distributions are given by

$$\bar{\mu}_{qs} = \frac{\sigma^2_s \theta_q + \gamma^2 \mu_s}{\sigma^2_s + \gamma^2},$$

$$\bar{\sigma}^2_{qs} = \frac{\sigma^2_s \gamma^2}{\sigma^2_s + \gamma^2},$$

$$\bar{\mu}_{qrs} = \frac{\sigma^2_s \theta_q + \gamma^2 \mu_{qs}}{2\sigma^2_s + \gamma^2},$$

$$\bar{\sigma}^2_{qrs} = \frac{\sigma^2_s \gamma^2}{2\sigma^2_s + \gamma^2}.$$ 

Since any higher order moments of a Gaussian distribution only depend on the mean and the variance, we can compute the polynomials $\bar{\pi}_s$ and $\bar{\pi}_{qs}$ of moments by the combination of $\{\bar{\mu}_{sqs}, \bar{\sigma}^2_{sqs}\}$ and $\{\bar{\mu}_{qs}, \bar{\sigma}^2_{qs}\}$, respectively. For instance, if we take $\pi(\theta_q)$ as $\theta_s + \theta_s + \theta_s^3$, $\bar{\pi}_s$ is $\bar{\mu}_{qs} + (\bar{\mu}_{qs} + \bar{\sigma}^2_{qs}) + (3\bar{\mu}_{qs} + 3\bar{\sigma}^2_{qs} + 3\bar{\sigma}^2_{qs})$.

**B. Derivative of GP**

Differentiation is a linear operation, so the derivative of $\psi$ is also a GP. Therefore we can use the derivative of the function $\psi$ as additional training data and can also predict an unknown derivative value at a new input. Now we consider the joint distribution of function outputs $\psi$ and its derivative $\psi' \equiv \frac{d\psi}{d\theta}$. Solak et al. [29] show that the covariance of the joint distribution is written as

$$\text{cov} [\psi'(\theta_q), \psi(\theta_q)] = k'(\theta_q, \theta_q),$$

$$\text{cov} [\psi'(\theta_q), \psi'(\theta_q)] = k''(\theta_q, \theta_q),$$

where

$$k'(\theta_q, \theta_r) = \frac{\partial^2 k(\theta_q, \theta_r)}{\partial \theta_q \partial \theta_r} = -\frac{1}{\gamma^2} (\theta_q - \theta_r) k(\theta_q, \theta_r),$$

$$k''(\theta_q, \theta_r) = \frac{\partial^2 k(\theta_q, \theta_r)}{\partial \theta_q \partial \theta_r} = \frac{1}{\gamma^2} (1 - \frac{1}{\gamma^2} (\theta_q - \theta_r)^2) k(\theta_q, \theta_r).$$

We generate $N'$ inputs $s' \equiv \{\theta_{N+1}, \ldots, \theta_{N+N'}\}$ and the corresponding derivative observations $y_h' \equiv \{\psi'_h(\theta_{N+1}), \ldots, \psi'_h(\theta_{N+N'})\}$ as new training data $D'_h \equiv \{s', y_h'\}$. Now the predictive mean and variance of $\psi_h(\theta_q)$ are given by

$$m_h(\theta_q) = \beta_h^T K_{ss}^T S, \quad \nu^2_h(\theta_q) = k_{ss} - \beta_s^T C^{-1} \kappa_s,$$

where $\kappa_s \equiv (K_{ss}^T, -K_{ss}^T)^T$, $\beta_s \equiv C^{-1}(y_h^T, y_h^T)^T$, and

$$C \equiv \begin{pmatrix} K(s, s) & K(s, s') \\ K'(s', s) & K'(s', s') \end{pmatrix}.$$ (12)

Predictive mean and variance of the derivative are also given by

$$m_h'(\theta_q) = \beta_h^T \kappa_s', \quad \nu^2_h'(\theta_q) = k_{ss} - \kappa_s'^T C^{-1} \kappa_s'$$ (13)

where $\kappa_s' \equiv (K_{ss}^T, K_{ss}^T)^T$. 