Discovering Latent Covariance Structures for Multiple Time Series

Anh Tong [1] **Jaesik Choi** [1]

Abstract

Analyzing multivariate time series data is important to predict future events and changes of complex systems in finance, manufacturing, and administrative decisions. The expressiveness power of Gaussian Process (GP) regression methods has been significantly improved by compositional covariance structures. In this paper, we present a new GP model which naturally handles multiple time series by placing an Indian Buffet Process (IBP) prior on the presence of *shared* kernels. Our selective covariance structure decomposition allows exploiting shared parameters over a set of multiple, selected time series. We also investigate the well-definedness of the models when infinite latent components are introduced. We present a pragmatic search algorithm which explores a larger structure space efficiently. Experiments conducted on five real-world data sets demonstrate that our new model outperforms existing methods in term of structure discoveries and predictive performances.

1. Introduction

Time series data analysis is important for numerous real-world applications: signal processing of audio and video data; the study of financial variables such as stocks, currencies, and crude oil prices. When several data sources are correlated, a model that exploits a group structure often demonstrates competitive predictive performance (Yuan & Lin, 2006). It is critical to learn how multiple time series are correlated. Many practical applications i.e. visualizing, filtering or generating reports from multiple time series, depend on their inherent encoded relations. However, it is non-trivial to extract such important relations among them.

A recent work contributed a highly general framework called the Automatic Bayesian Covariance Discovery (ABCD) which solves regression tasks using Gaussian Process (GP) models (Duvenaud et al., 2013; Lloyd et al., 2014; Ghahramani, 2015; Hwang et al., 2016; Malkomes et al., 2016; Kim & Teh, 2018). Previously, selecting GP kernels was heavily based on expert knowledge or trial-and-error. The ABCD automatically extracts an appropriate compositional covariance structure to fit data based on grammar rules; then it generates human-friendly reports explaining data. The compositional covariance structure makes the GP models more expressive and interpretable so that GP kernels are explained in a form of natural language. There are cognitive studies (Schulz et al., 2016; 2017) showing that compositional functions are intuitively preferred by humans. Exploiting these key properties of compositional kernel, we develop a kernel composition framework for multiple time series which produces explainable outputs with improved predictive accuracy.

A solid foundation for multi-task GP regression methods has been established in (Bonilla et al., 2007; Titsias & Lázaro-Gredilla, 2011; Álvarez et al., 2012; Wilson et al., 2012; Guarnizo et al., 2015). However, assigning compositional kernel structures has not yet been investigated in the existing multi-task GP regression methods. Notably, the multi-output GP regression network (GPRN) (Wilson et al., 2012) is highly general, and models data by the combinations of latent GP functions and weights which are also GPs. Applying structure search is challenging due to the huge search space to cover the whole network. In order to select appropriate covariance structures for multiple correlated sequences, we model time series by additive structures which are, instead of staying fixed, searched over a set of kernels. We place Indian Buffet Process (IBP) (Griffiths & Ghahramani, 2005; 2011) prior over an indicator matrix that represents whether the time series share one or many of these additive kernels. Furthermore, we introduce a search algorithm which enables us to explore a large kernel space.

Here, we present a new model to handle heterogeneous, correlated multiple time series by stochastic GP kernels. The combination of latent features and interpretable covariance structures brings a new tool to understand multiple time series better. Our model outputs human-readable reports with high-level abstraction as well as the relation among time series. We believe such results potentially facilitate

[1]Department of Computer Science and Engineering, Ulsan National Institute of Science and Technology, Ulsan, 44919, South Korea. Correspondence to: Jaesik Choi <jaesik@unist.ac.kr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

the process of decision making in many fields i.e. scientific discovery, financial management.

This paper offers the following contributions: (1) we introduce the Latent Kernel Model (LKM), justify its well-definedness and develop its approximate inference algorithm; (2) we introduce a search procedure applicable to multiple time series and our working model; (3) an application making comparison reports among multiple time series.

This paper is structured as follows. Section 4 presents our LKM. Section 5 introduces a search procedure working with this model. Section 6 shows our experiments on several real-world data sets and gives comparison reports produced from our models. We conclude in Section 7.

2. Related work

In the compositional kernel, there have been efforts on improving the efficiency of model selection i.e. using Bayesian optimization, or sparse GP (Malkomes et al., 2016; Kim & Teh, 2018; Lu et al., 2018) and relating human cognitive procedures (Schulz et al., 2016; 2017). Recently, (Sun et al., 2018) proposed a neural network construction of compositional kernels with a guarantee in approximation capacity. Yet, the framework is less interpretable. For multiple time series, (Hwang et al., 2016) introduced a global *shared* information among multiple sequences and individual kernels for each kernel. Our model is more general because no strong correlation assumption is required, the relation among time series is automatically discovered by IBP matrix instead.

Stochastic grammar for ABCD (Schaechtle et al., 2015) is introduced where interpretable kernels are selected via Bayesian learning over a binomial distribution imposed on the presence of kernels. It provides a sampling approach based on Venture probabilistic programming language (Mansinghka et al., 2014). Another work (Tong & Choi, 2016) represents kernel compositions in Stan language (Carpenter et al., 2017). A recent work (Saad et al., 2019) built on the top of Venture as well presents a program synthesis approach to extract compositional kernels. However, these works only can apply to a single time series. While in our case, we work on multiple time series using IBP prior with an in-depth investigation of the model construction.

In the multi-task learning perspective, multi-task learning for GP regression has been studied extensively (Teh et al., 2005; Bonilla et al., 2007; Álvarez et al., 2012; Wilson et al., 2012; Titsias & Lázaro-Gredilla, 2011; Guarnizo et al., 2015; Guarnizo & Álvarez, 2015). These methods commonly share limitations that GP kernel structures are fixed or given, not having the flexibility in selecting GP kernels. The additive kernel construction of our model is common with the Linear Model of Coregionalization (LMC) (Álvarez et al., 2012) and extensions (Álvarez & Lawrence, 2008; Ulrich et al., 2015; Parra & Tobar, 2017) where kernels are constructed by a linear combination of kernels. While LMC optimizes these weights together with GP hyperparameters, our model is based on a Bayesian approach to infer $\mathbf{Z}$. More importantly, the binary latent matrix $\mathbf{Z}$ enhances the interpretability transparency over real-valued weights.

In terms of stochastic kernel generation, (Jang et al., 2017b) proposed a Lévy kernel process where the mixture of kernels is obtained by placing a Lévy prior over the corresponding spectral density. The LKM is one of the attempts to put uncertainty on kernel constructions using IBP prior to select a set of interpretable kernels.

It is worth mentioning methods which learn complex functions including convolutional networks (LeCun et al., 1989) and sum-product networks (Poon & Domingos, 2011). AND-like and OR-like operation have the intuitively similar mechanisms of multiplication and summation in compositional kernels. Beyond this similarity between these operations and composing kernel operations, our work targets to study multiple complex functions where sharing kernels can be understood as AND-like operation among sequences.

3. Background

In this section, we provide a brief review of the Automatic Bayesian Covariance Discovery (ABCD) framework (Grosse et al., 2012; Duvenaud et al., 2013; Lloyd et al., 2014; Ghahramani, 2015) and Indian Buffet Process (IBP) (Griffiths & Ghahramani, 2005).

Gaussian Process (GP) Gaussian Process (GP) (Rasmussen & Williams, 2005) is defined as a multivariate Gaussian distribution over a (possibly infinite) collection of random variables. Whenever we select a subset from this collection, the distribution over the subset also is Gaussian. Commonly, GP is used as a prior over function values, denoted as $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$ with $m(x)$ is the mean function, $k(x, x')$ is the covariance (kernel) function. In practice, the mean function is usually chosen as a zero mean function. Like many other kernel methods, kernel tricks are applicable to construct new kernels for GP, be one of the key properties in the framework that we will describe next.

The ABCD framework The ABCD framework follows a typical Bayesian modeling process (see MacKay (2002)), being composed of several parts e.g. a language of models, a search procedure among models, and a model evaluation. The framework makes use of Gaussian Processes (GPs) to perform various regression tasks.

Selecting kernel functions plays a crucial role in learning

GP. ABCD searches a model out of an open-ended language of models which is constituted from a context-free grammar and base kernels. The base kernels model different characteristics of data such as white noise (WN), constant (C), smoothness (SE), periodicity (PER), and trending (LIN) (see Appendix A). The grammar makes it possible to explore and generate new kernels from base ones via composition rules such as the product rule and the sum rule. A greedy search is applied in ABCD like in Grosse et al. (2012), picking the most appropriate model based on a criterion e.g. Bayesian Infomation Criteria (BIC). Once the search procedure is finished, a human-readable report is generated from the interpretability of GP base kernels and their compositions.

Indian Buffet Process The IBP (Griffiths & Ghahramani, 2005) defines a distribution over a binary matrix $\mathbf{Z}$ with a finite number of rows and an infinite number of columns: $\mathbf{Z} \sim \text{IBP}(\alpha)$, with α is the concentration parameter. The matrix indicates feature assignments where the element at the i-th row and the j-th column expresses the presence or absence of the j-th feature in the i-th object. A natural application of IBP is the linear-Gaussian latent feature model (LFM) (Griffiths & Ghahramani, 2005). Data represented by $\mathbf{X}$ is factorized into an IBP latent matrix $\mathbf{Z}$ multiplying with a feature matrix $\mathbf{A}$ with a Gaussian noise matrix $\mathcal{E}$: $\mathbf{X} = \mathbf{Z}\mathbf{A} + \mathcal{E}$.

4. Latent Kernel Model (LKM)

In this section, we define the Latent Kernel Model (LKM) and discuss its theoretical properties and unique characteristics. Then we will introduce inference algorithms for LKM.

4.1. Definition

Notation Let us denote $\mathbf{x}_n = (x_{n1}, ..., x_{nD})^\top$ be a vector representing the n-th time series where x_{nd} is the data point of the n-th time series at the d-th time step t_d. Here, N is the number of time series and D is the number of data points in each time series. To clarify further notations, we denote a data matrix $\mathbf{X}$ taking $\mathbf{x}_n, n = 1 \dots N$ as rows. We introduce a latent matrix $\mathbf{Z}$ taking $\mathbf{z}_n, n = 1 \dots N$ as rows.

Given a set of GP kernels $\{\mathbf{C}_k\}_{k=1}^K$, we wish to model each time series $\mathbf{x}_n$ with

$$\begin{aligned} \mathbf{Z} &\sim \text{IBP}(\alpha), \\ \boldsymbol{f}_n &\sim \mathcal{GP}(\mathbf{0}, \sum_{k=1}^{K} z_{nk}\mathbf{C}_k), \\ \mathbf{x}_n &\sim \mathcal{N}(\boldsymbol{f}_n, \sigma_n^2\mathbf{I}), \end{aligned} \tag{4.1}$$

where α is the IBP concentration parameter. By the above model construction, an observation x_{nd} corresponds to a GP latent function variable $f_n(t_d)$. The $p(\mathbf{X}|\mathbf{Z})$ is the product of all $p(\mathbf{x}_n|\mathbf{z}_n)$ where

$$p(\mathbf{x}_n|\mathbf{z}_n) = |2\pi\mathbf{D}(\mathbf{z}_n)|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}_n^\top \mathbf{D}(\mathbf{z}_n)^{-1}\mathbf{x}_n\right), \tag{4.2}$$

with $\mathbf{D}(\mathbf{z}_n) = \sum_{k=1}^K z_{nk}\mathbf{C}_k + \sigma_n^2\mathbf{I}$, and $z_{nk} \in \{0, 1\}$ is the element of $N \times K$ matrix $\mathbf{Z}$ indicating whether the n-th time series has additive kernel $\mathbf{C}_k$. Since we place IBP on $\mathbf{Z}$, it can have infinitely many columns as $K \to \infty$. This model focuses on the process of creating the stochastic kernel $\mathbf{D}(\mathbf{z}_n)$ for each $\mathbf{x}_n$. The kernel selection procedure relies on learning IBP matrix via Bayesian inference.

4.2. Properties

Well-definedness of LKM Since an IBP prior is imposed on the matrix $\mathbf{Z}$, the number of its columns can go to infinity. Thus we may have an infinite number of kernels. It is important to verify whether $p(\mathbf{X}|\mathbf{Z})$ forms a well-defined probability distribution even with an infinite number of kernels. Griffiths & Ghahramani (2011) gave a detailed analysis in the case of LFM. In fact, $p(\mathbf{X}|\mathbf{Z})$ in LFM is independent to feature matrix because of marginalization over feature matrix. However, $p(\mathbf{X}|\mathbf{Z})$ in LKM is still associated with kernels in its representation. We will justify the well-definedness in the case of LKM as follow.

Proposition 1. *The likelihood of LKM is well-defined.*

Proof. The likelihood can be easily obtained by

$$p(\mathbf{X}|\mathbf{Z}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n).$$

We will use *lof* operator on $\mathbf{Z}$. The *lof* transforms a binary matrix by reordering its columns by the binary number associated to that column (Griffiths & Ghahramani, 2011). Since all kernels $\mathbf{C}_k$ are commutative, *lof* performs on $\mathbf{Z}$ without affecting $p(\mathbf{X}|\mathbf{Z})$ as kernels are exchanged accordingly. We apply *lof* on $\mathbf{Z}$ to obtain $[\mathbf{Z}^+\mathbf{Z}^0]$ where $\mathbf{Z}^+$ contains K^+ nonzero columns and $\mathbf{Z}_0$ contains K^0 zero columns. Each row in $\mathbf{Z}_+$ contributes to generate kernel $\mathbf{D}(\mathbf{z}_n) = \sum_{k=1}^{K^+} z_{nk}^+\mathbf{C}_k + \sigma_n^2\mathbf{I}$. When $K \to \infty$, K^+ still stays finite as the property of IBP. Thus, $\mathbf{D}(\mathbf{z}_n)$ is now the sum of a finite number of covariances kernels $\mathbf{C}_k$. This means that each multivariate Gaussian likelihood $p(\mathbf{x}_n|\mathbf{z}_n)$ has a well-defined covariance. Finally, we can conclude that $p(\mathbf{X}|\mathbf{Z})$ is well-defined. □

With the above proposition, IBP prior becomes a regularizer preventing the degradation of kernel construction (an explosion of the kernel variances) when increasing the number of kernels K.

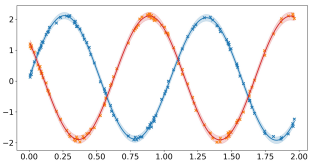


Figure 1: Fitting two functions using LKM. The toy data set contains two realizations generated from a GP prior with a periodic kernel.

Comparisons with existing models Feature sharing models (Titsias & Lázaro-Gredilla, 2011; Wilson et al., 2012; Guarnizo et al., 2015) commonly represent data as

$$\mathbf{x}_n = \sum_{k=1}^{K} w_k \boldsymbol{f}_k + \boldsymbol{\epsilon}_n,$$

with $\boldsymbol{f}_k, k = 1 \dots K$ are shared features, $\boldsymbol{\epsilon}_n, n = 1 \dots N$ are Gaussian noise vectors. Each $\boldsymbol{f}_k$ is a drawn GP realization from $\mathbf{C}_k$. The w_k can be placed spike and slab prior (Titsias & Lázaro-Gredilla, 2011) or are samples from GPs (Wilson et al., 2012).

Our LKM is more expressive than the feature sharing family in terms of function realizations. Suppose the posterior decomposition of additive Gaussian distributions presents as: If $\boldsymbol{f} = \boldsymbol{f}_1 + \boldsymbol{f}_2$, where $\boldsymbol{f}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_1)$, $\boldsymbol{f}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_2)$, the conditional distribution of $\boldsymbol{f}_1$ given the sum $\boldsymbol{f}$ is

$$\boldsymbol{f}_1 | \boldsymbol{f} \sim \mathcal{N}(\mathbf{K}_1^\top (\mathbf{K}_1+\mathbf{K}_2)^{-1} \boldsymbol{f}, \mathbf{K}_1 - \mathbf{K}_1^\top (\mathbf{K}_1+\mathbf{K}_2)^{-1} \mathbf{K}_1).$$

In the multiple time series setting, each decomposed component under the same GP prior could be realized differently in different time series. In other words, for a specific k, the posterior $\boldsymbol{f}_k | \mathbf{x}_n$ varies whenever $\mathbf{x}_n$ changes even with the fixed covariance $\mathbf{C}_k$. A simple setup in Figure 1 can verify this observation. We generate two sequences from a single periodic GP and then run LKM on this data with two different periodic kernels $\mathbf{C}_1$ and $\mathbf{C}_2$. When we learn LKM, $\mathbf{Z} = [0, 1; 0, 1]$ is obtained. That is, LKM is able to recognize these two realizations from one GP.

We also emphasize that the Bayesian approach that is considered in our kernel construction, can be viewed as a stochastic kernel generative process (Jang et al., 2017b).

Figure 2 illustrates the plate notations of LKM and R-ABCD (Hwang et al., 2016). R-ABCD shares a global kernel for all time series and allocates a distinctive kernel $\mathbf{C}_n$ for each time series. Note that spectral mixture kernel (Wilson & Adams, 2013) is used for $\mathbf{C}_n$ in R-ABCD prevents ones from deriving interpretable models.

4.3. Inference algorithm

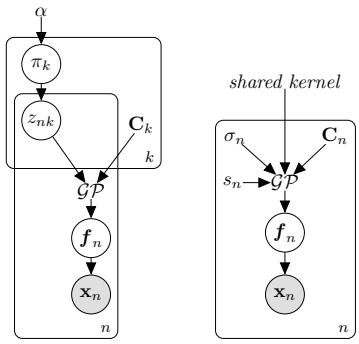
Variational inference Variational inference methods approximate the true posterior $p(\mathbf{Z}|\mathbf{X})$ by a variational distribution $q(\mathbf{Z})$. The method converts the optimization problem of KL divergence between p and q into an equivalent problem by maximizing the evidence lower bound (ELBO) $\mathcal{L}$,

$$\begin{aligned} \log p(\mathbf{X}) &\geq \mathbb{E}[\log p(\mathbf{X}, \mathbf{Z})] + H[q] \\ &= \mathbb{E}[\log p(\mathbf{Z})] + \mathbb{E}[\log p(\mathbf{X}|\mathbf{Z})] + H[q] \triangleq \mathcal{L}. \end{aligned}$$

where $\mathbb{E}$ indicates the expectation over the approximate posterior distribution $q(\mathbf{Z})$, and $H[q]$ is the entropy of q. The last equation in the above derivation comes from the model definition in Equation 4.1 where the joint distribution $p(\mathbf{X}, \mathbf{Z})$ is in the form of $p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})$. Here, we choose the variational distribution $q(\mathbf{Z})$ in the mean-field family. It is factorized into $q(z_{nk}) = \text{Bernoulli}(z_{nk}; \nu_{nk})$.

The first term $\mathbb{E}[\log p(\mathbf{Z})]$ in $\mathcal{L}$ is explained in Appendix B (Doshi et al., 2009).

Now our main focus is to estimate $\mathbb{E}[\log p(\mathbf{X}|\mathbf{Z})]$. Recall that $p(\mathbf{X}|\mathbf{Z}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)$, we can break $\mathbb{E}[\log p(\mathbf{X}|\mathbf{Z})]$ into the sum of $\mathbb{E}[\log p(\mathbf{x}_n|\mathbf{z}_n)]$. The evaluation of each $\mathbb{E}[\log p(\mathbf{x}_n|\mathbf{z}_n)]$ is expensive since it needs to compute the expectation of GP likelihood functions associated with discrete random variables $\mathbf{Z}$. Specifically, $\mathbb{E}[\log p(\mathbf{x}_n|\mathbf{z}_n)]$ is written as the sum of $-\frac{1}{2}\mathbf{x}_n^\top \mathbb{E}\left[\mathbf{D}(\mathbf{z}_n)^{-1}\right] \mathbf{x}_n$ (or the expectation of data-fit term in GP likelihood), $-\frac{1}{2}\mathbb{E}\left[\log |2\pi \mathbf{D}(\mathbf{z}_n)|\right]$ (or the expectation of GP model complexity) and a constant term. Each expectation is the sum of following 2^K terms: (1) $p(\mathbf{z}_n = \mathbf{t})\mathbf{D}(\mathbf{t})^{-1}$ for all $\mathbf{t} \in \{0, 1\}^K$ in the case of the expectation of inverse matrix; (2) $p(\mathbf{z}_n = \mathbf{t}) \log |2\pi \mathbf{D}(\mathbf{t})|$ for all $\mathbf{t} \in \{0, 1\}^K$ in the case of the expectation of log-determinant. Hence, it is not practical to estimate an exponential number of inverse and determinant operations.



(a) LKM (b) R-ABCD

Figure 2: Graphical model of (a) LKM and (b) R-ABCD.

Relaxation To mitigate the difficulty in estimating $\mathbb{E}[\log p(\mathbf{x}_n|\mathbf{z}_n)]$, we first relax the discrete random variables z_{nk} to a continuous ones, then estimate the expectation using Monte Carlo method. The relaxation turns the Bernoulli random variables $z_{nk} \sim \text{Bernoulli}(\nu_{nk})$ into 2-dimensional

continuous random variable $[\tilde{z}_{nk}, \tilde{z}_{nk}] \sim \text{Concrete}(\nu_{nk}, \lambda)$, where λ is the temperature parameter (Maddison et al., 2017). Here, the categorical random variable $[z_{nk}, 1 - z_{nk}]$ corresponds to the relaxed one $[\tilde{z}_{nk}, \tilde{z}_{nk}]$. We are interested in $\tilde{z}_{nk}$ which corresponds to z_{nk}. A sample of $\tilde{z}_{nk}$ is drawn by sampling g_1 and g_2 from $\text{Gumbel}(0, 1)$ and computing as

$$\tilde{z}_{nk} = \frac{\exp(\frac{\log(\nu_{nk})+g_1}{\lambda})}{\exp(\frac{\log(\nu_{nk})+g_1}{\lambda}) + \exp(\frac{\log(1-\nu_{nk})+g_2}{\lambda})}.$$

This is known as the Gumbel-Softmax trick (Maddison et al., 2017; Jang et al., 2017a). The unbiased estimation of $\mathbb{E}[\log p(\mathbf{x}_n|\mathbf{z}_n)]$ after relaxation is

$$\mathbb{E}[\log(p(\mathbf{x}_n|\mathbf{z}_n)] \approx \frac{1}{m}\sum_{i=1}^{m} \log p(\mathbf{x}_n|\tilde{\mathbf{z}}_n^{(i)})),$$

where m is the number of samples, $\{\mathbf{z}_n^{(i)}\}_{i=1}^m$ is the set of samples. The kernel $\mathbf{D}(\tilde{\mathbf{z}}_n)$ now takes all $\mathbf{C}_k$ into account since $\tilde{\mathbf{z}}_n$ is in $(0, 1)^K$ instead of $\{0, 1\}^K$. Now the number of evaluations on matrix inversions and determinants is the number of sample M, instead of the number of all (exponential) configurations generated from K binary random variables $\mathbf{z}_n$. Moreover, the estimation benefits from this reparameterization trick to estimate gradients in stochastic computation graph (Schulman et al., 2015).

5. Structure discovery in multiple time series

In this section, we present a search algorithm to discover GP compositional kernels for multiple time series.

Search scheme To cope with the broad structure space, our algorithms follows the principle of greedy algorithms (Grosse et al., 2012; Duvenaud et al., 2013; Lloyd et al., 2014). That is, we maintain a set of additive kernel structures $\{\mathcal{S}_d^{(k)} | \mathcal{S}_d^{(k)} = \prod_l \mathcal{B}_d^{(k_l)}$ with $\mathcal{B}_d^{(k_l)}$s are base kernels, $k = 1 \dots K\}$ at a search depth d. We map correspondingly $\mathcal{S}_d^{(k)}$ to the required kernels $\mathbf{C}_k$ in LKM. At the next depth, the set will recruit new additive kernels by expanding some of the elements of the set at the current depth d. The context-free grammar rules of the expansion are the same with Compositional Kernel Learning (CKL) (Duvenaud et al., 2013). However, for the case when $\mathcal{S}_d^{(k)}$ is expanded into a new kernel which is written in an additive form as $\sum_{m=1}^{M} \mathcal{S}_{d+1}^{(k_m)}$, we will consider this expansion as M separated expansions $\mathcal{S}_d^{(k)} \to \mathcal{S}_{d+1}^{(k_m)}$. The generated structures $\mathcal{S}_{d+1}^{(k_m)}$ are added to the set rather than the sum $\sum_{m=1}^{M} \mathcal{S}_{d+1}^{(k_m)}$. This procedure always makes new candidate structures satisfy the definition of $\{\mathcal{S}_d^{(k)}\}$ without assuming an arbitrary sum.

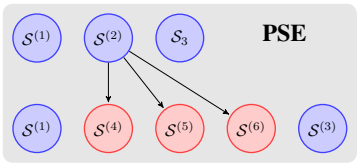


Figure 3: PSE with $\mathcal{S}^{(2)}$ expanded into 3 others to create a new set.

Algorithm 1 Partial set expansion of LKM learning

Require: Input data and search depth D, initial $\{\mathcal{S}_d^{(k)}\}$
for $d = 1 \dots D$ **do**
  for $\mathcal{S}$ in $\{\mathcal{S}_d^{(k)}\}$ of depth d **do**
    Update $\{\mathcal{S}_d^{(k)}\} \leftarrow \{\mathcal{S}_d^{(k)}\} \backslash \mathcal{S} \cup expand(\mathcal{S})$
    Run LKM learning
    if improvement in BIC **then**
      Use this updated set $\{\mathcal{S}_d^{(k)}\}$
    else
      Rollback to previous set $\{\mathcal{S}_d^{(k)}\}$
    end if
  end for
end for

Partial set expansion (PSE) Our search algorithm iteratively expands $\mathcal{S}_d^{(k)}$ and obtain a set of candidates $\{\mathcal{S}_d^{(k_1)}, \dots, \mathcal{S}_d^{(k_m)}\}$. We make a new set which is the union of the previous one excluded the selected structure $\{\mathcal{S}_d^{(k)}\}_{k=1}^K \backslash \{\mathcal{S}_d^{(i)}\}$ and the new candidate structures $\{\mathcal{S}_d^{(i_1)}, \dots, \mathcal{S}_d^{(i_m)}\}$ (Figure 3). Our variational inference algorithm (described in Section 4.3) learns $\mathbf{Z}$ and GP kernels. If there is an improvement in BIC (Schwarz, 1978.), we keep the updated kernel set. Otherwise, it rolls back to the previous one. We proceed to the next expansion using this updated one (Algorithm 1).

Advantages of our PSE algorithm are (1) it does not make drastic increases in structure space in each expansion, (2) it carefully assesses models by a selection criterion (BIC) and flexibly falls back to the previous model if the criterion does not select the new one, (3) the fewer number of kernels in PSE makes us easier to initialize GP hyperparameters as well as reduce the number of restarts learning $\mathbf{Z}$.

Our kernel search procedure is a meta search algorithm inspired from oracle machines in computational theory (Papadimitriou, 1994). The LKM plays a role as an *oracle*. Given a set of kernel structures, one tries to ask the oracle to decide the appropriate structures. The oracle will response an answer as $\mathbf{Z}$ in our case. Exploiting the returned $\mathbf{Z}$, the kernel structures will be elaborated more by performing PSE. The procedure is repeated by making new inquiry based on the expanded structures.

We emphasize that PSE with LKM considers a larger number of kernel structures than those in CKL. Suppose that CKL and our search algorithm have the same found structure at a depth d. Whereas the CKL's structure is $\mathcal{S}_d = \mathcal{S}_d^{(1)} + \cdots + \mathcal{S}_d^{(K)}$, PSE represents it as a set $\{\mathcal{S}_d^{(1)}, \ldots, \mathcal{S}_d^{(K)}\}$. Let L be the largest number of base kernels in $\mathcal{S}_d^{(k)}$, and R be the maximum number of grammar rules per substructure. All possible search candidates in CKL is $O(RK2^L + R2^K)$ kernels, while PSE incorporating with LKM considers $O(K2^{R2^L+K})$ number of kernels. Detailed analysis is provided in Appendix C.

Although our search algorithm explores a much larger search space than CKL in theory, the prior over $\mathbf{Z}$ still limits the expressiveness power of our model. Moreover, learning $\mathbf{Z}$ relies on a gradient-based method where the global optimal is not guaranteed. Thus, our kernel search algorithm may not find the optimal kernel over all the possible candidates.

6. Experimental evaluations

In this section, we describe data sets and demonstrate both qualitative and quantitative results.

6.1. Real-world time series data

Strongly correlated data sets We tested our algorithm on three different data sets: US stock prices, US housing markets and currency exchanges. These data sets are well-described and publicly accessible (Hwang et al., 2016). The US stock price data set consists of 9 stocks (GE, MSFT, XOM, PFE, C, WMT, INTC, BP, and AIG) containing 129 adjusted closes taken from the second half of 2001. The US housing market data set includes the 120-month housing prices of 6 cities (New York, Los Angeles, Chicago, Phoenix, San Diego, San Francisco) from 2004 to 2013. The currency data set includes 4 currency exchange rates from US dollar to 4 emerging markets: South African Rand (ZAR), Indonesian Rupiah (IDR), Malaysian Ringgit (MYR), and Russian Rouble (RUB). Each currency exchange time series has 132 data points.

Heterogeneous data set We collected time series from various domains into a data set. It consists of gold prices, crude oil prices, NASDAQ composite index, and USD index[1] from 2015 July 1st to 2018 July 1st. We call this data set as GONU (Gold, Oil, NASDAQ, USD index). Each time series has 157 weekly prices or indexes taken from Quandl (2018). The interactions between this sets of time series are known to be complex. For instance, the gold and oil prices might have a negative correlation where one may increase but the other decreases. There are many studies in the financial research focusing on these target time series (Filis et al., 2011; Reboredo et al., 2014).

Epileptic seizure data set We retrieved the epileptic seizure data set (Andrzejak et al., 2002) from UCI repository (Dheeru & Karra Taniskidou, 2017). This data set contains EEG recordings of brain activities for 23.6s. Each record corresponds to one out of 5 activities including eyes open, eyes closed, identifying the tumor, located the tumor and seizure activity. Each time series contains 178 data points.

[1]Quandl codes respectively are `WGC/GOLD_DAILY_USD`, `FRED/DCOILBRENTEU`, `NASDAQOMX/COMP`,`FRED/DTWEXM`

6.2. Qualitative results

With the motivation that interpretable machine learning models can help understand data better, thereby fostering scientific discovery and decision making, we carried experiments on the mentioned data sets to demonstrate the potential applicability of our search algorithm on LKM.

6.2.1. Exploiting information from $\mathbf{Z}$

Learning Z We visualize the variational parameters $\boldsymbol{\nu}$ in Figure 4. The value of ν_{nk} is the probability of $z_{nk} = 1$. The bigger ν_{nk} is, the more probable the kernel $\mathbf{C}_k$ is selected for time series $\mathbf{x}_n$.

Interpreting Z We randomly take 50 time series from the epileptic seizure data where each activity has 10 time series. Because finding a covariance kernel decomposition for a large number of time series is time-consuming, and therefore prohibits kernel structure search, we looked for latent kernels from the set of kernels $\{\mathrm{SE}_1, \mathrm{SE}_2, \mathrm{PER}_1, \mathrm{PER}_2, \mathrm{SE}_3 \times \mathrm{PER}_3, \mathrm{SE}_4 \times \mathrm{PER}_4\}$. Figure 5 illustrates a summary of the model outputs. Readers may refer Appendix E for the full output.

We observe several interesting properties. Located tumor and identifying tumor are quite similar because the corresponding block matrix from $\mathbf{Z}$ has the same sparsity. Also, having fewer active SE kernels indicates that they do not vary much. The activities of opening eyes and closed eyes commonly have rapidly varying signals with small length-scales. The seizure, on the other hand, has a similar level of sparsity comparing to those of opening eyes or closed eyes. However, there is no sign of low-frequency periodic pattern.

The latent matrix $\mathbf{Z}$ encodes certain relations between time series in the light of kernel interpretability. Next, we fully employ the description of kernels to generate comparison reports.

6.2.2. Comparison report

Overview comparison By taking the advantage of the learned latent matrix $\mathbf{Z}$ and the descriptive properties of found GP covariance structures, we generate a human-

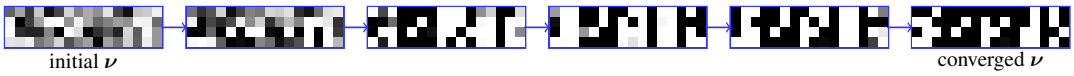


Figure 4: The visualization of ν as the training of LKM goes on. The columns indicates time series. The row indicates kernels $\mathbf{C}_k$.

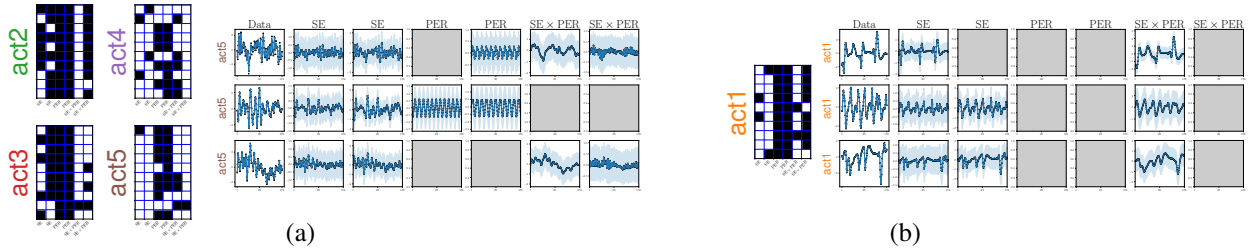


Figure 5: Epileptic seizure data set. There are 5 activities of EEG recording: seizure (act1), located tumor (act2), identifying tumor (act3), eyes closed (act4), eyes open (act5). (a) Non-seizure. *Left*: part of learned $\mathbf{Z}$ corresponding to each activity, black means $z_{nk} = 0$, otherwise white; *Right*: posterior of 3 last time series from act5 with their decomposition. (b) Seizure. *Left*: part of learned $\mathbf{Z}$ from act1; *Right*: posterior plot of 3 first time series from act1 with their decomposition. The missing subplots or gray background plots indicate $z_{nk} = 0$.

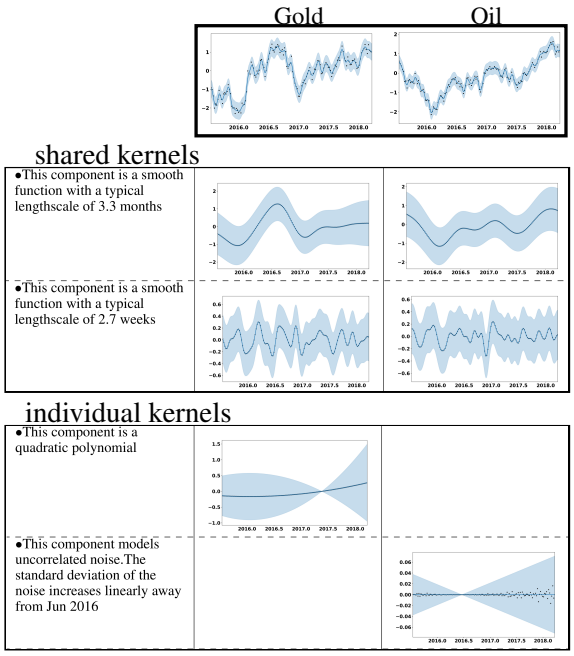


Figure 6: A part of pairwise comparison between Gold and Oil in GONU data set. The uppermost plots are the posterior means and variances of two time series. The remaining plots contain shared components and individual components with descriptions and posteriors $\boldsymbol{f}_k|\mathbf{x}_n$ for each time series. The blank in the individual components means "not available".

readable report containing the comparison among time series. For example, the generated text can have formats like

"$[T_1, \ldots, T_m]$ share [*description*]"

where the replacement of $[T_1, \ldots, T_m]$ is a set of time series, [*description*] is generated by the found GP

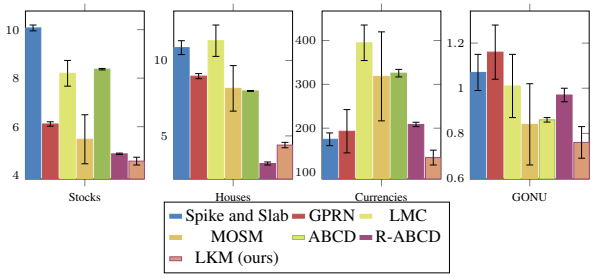


Figure 7: RMSEs for each data set (9 stocks, 6 houses, 4 currencies, GONU) with corresponding methods.

structure. Below is extracted from GONU data set.

- Gold, Oil, NASDAQ, USD index share the following property:

This component is periodic with a period of 1.4 years but with varying amplitude. The amplitude of the function increases linearly away from Apr 2017. The shape of this function within each period has a typical lengthscale of 4.9 days.

- Gold, Oil, USD index share the following property:

This component is a smooth function with a typical lengthscale of 2.7 weeks.

- NASDAQ has the following property:

This component is a linear function.

Pairwise comparison We provide another type of descriptive comparisons. Given a set of N time series, the output of our model can generate $\binom{N}{2}$ reports which compare each pair of time series. These reports give us a more detailed insight than the overview comparison. A report consists of shared components and individual ones between time series. Alongside with the description of the kernel structure of

	9 stocks		6 houses		4 currencies		GONU	
	RMSE	MNLP	RMSE	MNLP	RMSE	MNLP	RMSE	MNLP
Spike and Slab	$10.07_{\pm 0.12}$	$2.87_{\pm 0.05}$	$10.85_{\pm 0.46}$	$6.92_{\pm 0.09}$	$174.71_{\pm 14.52}$	$4.09_{\pm 0.10}$	$1.07_{\pm 0.08}$	$2.36_{\pm 0.11}$
GPRN	$6.11_{\pm 0.09}$	$2.78_{\pm 0.14}$	$8.96_{\pm 0.17}$	$6.64_{\pm 0.46}$	$193.13_{\pm 49.40}$	$4.24_{\pm 0.20}$	$1.16_{\pm 0.12}$	$2.46_{\pm 0.28}$
LMC	$8.20_{\pm 0.53}$	$2.24_{\pm 0.23}$	$11.31_{\pm 1.04}$	$5.90_{\pm 0.46}$	$394.83_{\pm 40.54}$	$4.90_{\pm 0.15}$	$1.01_{\pm 0.14}$	$\mathbf{1.43_{\pm 0.11}}$
MOSM	$5.48_{\pm 1.01}$	$2.97_{\pm 0.01}$	$8.15_{\pm 1.51}$	$5.90_{\pm 0.20}$	$318.26_{\pm 101.52}$	$3.93_{\pm 0.15}$	$0.84_{\pm 0.18}$	$3.13_{\pm 1.06}$
ABCD	$8.37_{\pm 0.03}$	$2.58_{\pm 0.05}$	$7.98_{\pm 0.03}$	$5.61_{\pm 0.05}$	$325.58_{\pm 8.64}$	$4.47_{\pm 0.04}$	$0.86_{\pm 0.01}$	$2.21_{\pm 0.03}$
R-ABCD	$4.88_{\pm 0.03}$	$1.95_{\pm 0.05}$	$\mathbf{3.17_{\pm 0.10}}$	$6.07_{\pm 0.09}$	$208.32_{\pm 5.02}$	$\mathbf{3.62_{\pm 0.03}}$	$0.97_{\pm 0.03}$	$2.01_{\pm 0.10}$
LKM	$\mathbf{4.58_{\pm 0.16}}$	$\mathbf{1.87_{\pm 0.10}}$	$4.37_{\pm 0.16}$	$\mathbf{5.54_{\pm 0.40}}$	$\mathbf{133.00_{\pm 16.92}}$	$\mathbf{3.61_{\pm 0.16}}$	$\mathbf{0.76_{\pm 0.07}}$	$1.90_{\pm 0.25}$

Table 1: RMSEs and NMLPs for each data set with corresponding methods (5 independent runs per method). In most cases, LKM has lower RMSEs and NMLPs compared to those of existing methods.

$\mathbf{C}_k$, this type of report presents the corresponding posterior $\boldsymbol{f}_k|\mathbf{x}_n$ which will illustrate the variations of GP realizations on different time series (see Figure 6).

We bring a brief analysis of GONU data set as an example after taking a quick look over the generated report. For instance, the gold and oil prices share many common characteristics (long and short lengthscale varying), showing a marginally small difference. On the other hand, NASDAQ and USD indices differ each other with many distinctive individual kernels $\mathbf{C}_k$s. Interestingly, the negative correlation behavior between the oil and USD indices (i.e. two time series often go in opposite directions) can be observed by shared kernels using LKM (see Appendix D). These reports give an easy understanding for ones who do not have knowledge in finance.

6.3. Quantitative results

Experiment setup All experiments are conducted to predict future events (extrapolation) by splitting all data sets and trained with the first 90%, then tested with the remaining 10% as in the standard setting for extrapolation tasks. Root mean square error (RMSE) and Mean Negative Log Likelihood (MNLP) (Lázaro-Gredilla et al., 2010) are the main evaluation metrics in all data sets.

Compare to multi-task GPs We compare multi-task GP models including 'Spike and Slab' model (Titsias & Lázaro-Gredilla, 2011) , GP regression network (GPRN) (Wilson et al., 2012; Nguyen & Bonilla, 2013), Linear Model of Coregionalization (LMC) (Álvarez et al., 2012; GPy, since 2012) and Multi-Output Spectral Mixture (MOSM) (Parra & Tobar, 2017). The result in Table 1 and Figure 7 indicates that our methods significantly outperform these models. This result could be attributed to that LKM leveraged by PSE selects compositional kernels which are flexible enough to fit complex data.

Compare to existing kernel composition approaches We ran ABCD on individual time series then aggregated the results to compare with our models. Our model outperforms ABCD which is known as one of the state-of-the-art GP-based regression methods on univariate time series. It proves that our belief about the correlations among multiple time series is plausible.

We then compare with R-ABCD (Hwang et al., 2016). Rather than making the assumption that all time series share a single global kernel, our model recognizes which structures are shared globally or partially. Quantitatively, LKM shows promising results in prediction tasks. It outruns R-ABCD in most of the data sets (Table 1 and Figure 7). In a relationally complex data set like GONU, LKM is significantly better while R-ABCD failed as the restriction due to its feature (function) sharing assumption.

Spike and Slab and GPRN models perform better than ABCD and R-ABCD in the currency data set where it contains highly volatile data. Although our model shares some computational procedures with ABCD and R-ABCD, our model is more robust to handle different types of time series data.

7. Conclusion

In this paper, we study a new perspective of multi-task GP learning where kernel structures are appropriately selected. We introduce the LKM which learns kernel decompositions from a stochastic kernel process. We further present a pragmatic search algorithm leveraging our models to explore a larger structure space efficiently. Experimental results demonstrate promising performance in prediction tasks. Our proposed model also outputs a high-quality set of interpretable kernels which produces a comparison reports among multiple time series.

Acknowledgment

This work is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: the Ministry of Science and ICT) (NRF-2017R1A1A1A05001456) and Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the MSIT (No.2017-0-01779, a machine learning and statistical inference framework for explainable artificial intelligence).

References

Álvarez, M. A. and Lawrence, N. D. Sparse convolved gaussian processes for multi-output regression. In *NeurIPS*, pp. 57–64, 2008.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3), 2012.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 64, 2002. doi: 10.1103/PhysRevE.64.061907.

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I. Multi-task gaussian process prediction. In *NeurIPS*, pp. 153–160, 2007.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 2017.

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Doshi, F., Miller, K., Gael, J. V., and Teh, Y. W. Variational inference for the indian buffet process. In *AISTATS*, 2009.

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. In *ICML*, pp. 1166–1174, 2013.

Filis, G., Degiannakis, S., and Floros, C. Dynamic correlation between stock market and oil prices: The case of oil-importing and oil-exporting countries. *International Review of Financial Analysis*, 20(3):152 – 164, 2011.

Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, since 2012.

Griffiths, T. L. and Ghahramani, Z. Infinite latent feature models and the indian buffet process. In *NeurIPS*, pp. 475–482, 2005.

Griffiths, T. L. and Ghahramani, Z. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

Grosse, R., Salakhutdinov, R., Freeman, W., and Tenenbaum, J. Exploiting compositionality to explore a large space of model structures. In *UAI*, pp. 306–315, 2012.

Guarnizo, C. and Álvarez, M. A. Indian Buffet process for model selection in convolved multiple-output Gaussian processes. *ArXiv e-prints*, 1503.06432, 2015.

Guarnizo, C., Álvarez, M. A., and Orozco, Á. Á. Indian buffet process for model selection in latent force models. In *CIARP*, pp. 635–642, 2015.

Hwang, Y., Tong, A., and Choi, J. Automatic construction of nonparametric relational regression models for multiple time series. In *ICML*, pp. 3030–3039, 2016.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017a.

Jang, P. A., Loeb, A., Davidow, M., and Wilson, A. G. Scalable levy process priors for spectral kernel learning. In *NeurIPS*, pp. 3943–3952. 2017b.

Kim, H. and Teh, Y. W. Scaling up the Automatic Statistician: Scalable structure discovery using Gaussian processes. In *AISTATS*, pp. 575–584, 2018.

Lázaro-Gredilla, M., Quiñonero Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. Sparse spectrum gaussian process regression. *J. Mach. Learn. Res.*, 11:1865–1881, 2010.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Lloyd, J. R., Duvenaud, D., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. Automatic construction and natural-language description of nonparametric regression models. In *AAAI*, pp. 1242–1250, 2014.

Lu, X., Gonzalez, J., Dai, Z., and Lawrence, N. Structured variationally auto-encoded optimization. In *ICML*, volume 80, pp. 3267–3275, 2018.

MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

Maddison, C. J., Mnih, A., and Teh, Y. W. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *ICLR*, 2017.

Malkomes, G., Schaff, C., and Garnett, R. Bayesian optimization for automated model selection. In *NeurIPS*, pp. 2892–2900, 2016.

Mansinghka, V. K., Selsam, D., and Perov, Y. Venture: A higher-order probabilistic programming platform with programmable inference. *arXiv preprint*, arXiv:1404.0099, 2014.

Nguyen, T. and Bonilla, E. Efficient variational inference for gaussian process regression networks. In *AISTATS*, pp. 472–480, 2013.

Papadimitriou, C. H. *Computational complexity.* Addison-Wesley, 1994.

Parra, G. and Tobar, F. Spectral mixture kernels for multi-output gaussian processes. In *NeurIPS*, pp. 6684–6693, 2017.

Poon, H. and Domingos, P. M. Sum-product networks: A new deep architecture. In *UAI*, pp. 337–346, 2011.

Quandl. A marketplace for financial data, 2018.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

Reboredo, J. C., Rivera-Castro, M. A., and Zebende, G. F. Oil and us dollar exchange rate dependence: A detrended cross-correlation approach. *Energy Economics*, 42:132 – 139, 2014.

Saad, F. A., Cusumano-Towner, M. F., Schaechtle, U., Rinard, M. C., and Mansinghka, V. K. Bayesian synthesis of probabilistic programs for automatic data modeling. *Proc. ACM Program. Lang.*, 3(POPL):37:1–37:32, 2019. doi: 10.1145/3290350.

Schaechtle, U., Zinberg, B., Radul, A., Stathis, K., and Mansinghka, V. K. Probabilistic programming with gaussian process memoization. *ArXiv e-prints*, 1512.05665, 2015.

Schulman, J., Heess, N., Weber, T., and Abbeel, P. Gradient estimation using stochastic computation graphs. In *NeurIPS*, pp. 3528–3536. 2015.

Schulz, E., Tenenbaum, J., Duvenaud, D. K., Speekenbrink, M., and Gershman, S. J. Probing the compositionality of intuitive functions. In *NeurIPS*, pp. 3729–3737. 2016.

Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M., and Gershman, S. J. Compositional inductive biases in function learning. *Cognitive Psychology*, 99 (Supplement C):44 – 79, 2017.

Schwarz, G. Estimating the dimension of a mode. *The Annals of Statistics*, 6(2), 1978.

Sun, S., Zhang, G., Wang, C., Zeng, W., Li, J., and Grosse, R. Differentiable compositional kernel learning for Gaussian processes. In *ICML*, volume 80, pp. 4828–4837, 2018.

Teh, Y. W., Seeger, M. W., and Jordan, M. I. Semiparametric latent factor models. In *AISTATS*, 2005.

Teh, Y. W., Grr, D., and Ghahramani, Z. Stick-breaking construction for the indian buffet process. In *AISTATS*, pp. 556–563, 2007.

Titsias, M. K. and Lázaro-Gredilla, M. Spike and slab variational inference for multi-task and multiple kernel learning. In *NeurIPS*, pp. 2339–2347. 2011.

Tong, A. and Choi, J. Automatic Generation of Probabilistic Programming from Time Series Data. *arXiv e-prints*, art. arXiv:1607.00710, 2016.

Ulrich, K. R., Carlson, D. E., Dzirasa, K., and Carin, L. GP kernels for cross-spectrum analysis. In *NeurIPS*, pp. 1999–2007, 2015.

Wilson, A. G. and Adams, R. P. Gaussian process kernels for pattern discovery and extrapolation. In *ICML*, pp. 1067–1075, 2013.

Wilson, A. G., Knowles, D. A., and Ghahramani, Z. Gaussian process regression networks. In *ICML*, 2012.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006. doi: 10.1111/j.1467-9868.2005.00532.x.