Article

Kernel Analysis Based on Dirichlet Processes Mixture Models

Jinkai Tian ^{1,2}, Peifeng Yan ^{1,2} and Da Huang ^{1,2,*}

¹ State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China

² College of Computer, National University of Defense Technology, Changsha 410073, China

* Correspondence: huangda1109@163.com

Received: 22 July 2019; Accepted: 29 August 2019; Published: 2 September 2019

Abstract: Kernels play a crucial role in Gaussian process regression. Analyzing kernels from their spectral domain has attracted extensive attention in recent years. Gaussian mixture models (GMM) are used to model the spectrum of kernels. However, the number of components in a GMM is fixed. Thus, this model suffers from overfitting or underfitting. In this paper, we try to combine the spectral domain of kernels with nonparametric Bayesian models. Dirichlet processes mixture models are used to resolve this problem by changing the number of components according to the data size. Multiple experiments have been conducted on this model and it shows competitive performance.

Keywords: Dirichlet processes; Dirichlet processes mixture models; Gaussian mixture models; kernel; spectral analysis

1. Introduction

Probabilistic models are essential for machine learning to have a correct understanding of problems. While frequentist models are considered a traditional method for statisticians, Bayesian models are more widely used in machine learning fields. A Bayesian model is a statistical model where probability is used to represent all uncertainties in the model. Uncertainty of both input and output are under consideration in a Bayesian model. A parametric Bayesian model assumes a finite set of parameters which is not flexible because the complexity of the model is bounded while the amount of data is unbounded. A Bayesian nonparametric model is a Bayesian model on infinite-dimensional parameter space. The amount of information captured by the model from the data can grow as the amount of the data grows. These models are always connected with some kinds of stochastic processes to describe an infinite number of parameters. For example, the Gaussian process (GP) and Dirichlet process mixture models (DPMM) are two cornerstones of Bayesian nonparametric models.

Dirichlet processes were first introduced in Reference [1]. The stick-breaking methods [2], Pólya urn scheme [3] and the Chinese restaurant process (CRP) [4,5] are several different definitions and constructions of Dirichlet processes. Dirichlet processes are the basis for many Bayesian nonparametric models and are used to process spatial data [6,7], tree-structured data [8,9], relational data [10,11] and sequential data [12–15].

The recent development of deep learning has led to the extensive use of the neural network in a variety of domains and a higher performance than conventional methods is achieved. However, this is less than satisfactory for statisticians because of the uninterpretability of black-box function approximation. Neal [16] shows that as the number of hidden unions approaches infinity, Bayesian neural networks converge to Gaussian processes. Thus Gaussian processes are regarded as powerful and interpretable alternatives to neural networks. It provides a principled, practical and probabilistic approach to learning in kernel machines [17].

The property of a Gaussian process is closely related to its kernel (or covariance function). Kernels provide a depiction of smoothness and periodicity for models and parameter selection of kernel is essential for model training. There are many widely used kernels, for example, the squared exponential (SE) kernels, rational quadratic kernels, polynomial kernels and the Matérn kernels family. The expressiveness of a kernel is defined as the ability to discover the inherent patterns of the data.

Recent research [18–20] has proposed some expressive kernels to discover patterns without human intervention. Deep architectures are used in designing kernels. Other previous work [21,22] sought to combine the Gaussian process into a Bayesian neural network framework. However, these approaches usually do not have closed forms and are thus less interpretable. Meanwhile, sophisticated approximate inference techniques, for example, variational inference and expectation propagation, are required in the training step. Combining and modifying existing kernels to make new kernels is straightforward because the sum, the product or the convolution of two kernels is also a kernel. There are many challenges [23,24] to designing kernels according to this principle for specialized applications. Specific restrictions are enforced on these kernels to avoid overfitting and complicated inference. However, it is hard to generalize these kernels to other applications for their specialized kernel structures.

We use Gaussian mixture models (GMM) to analyze the spectral density of popular kernels and find that traditional GMM suffer from overfitting and underfitting by a fixed number of components. Therefore, we seek to combine an infinite mixture model in designing kernels. This innovation introduces more flexibility and expressiveness into designed kernels.

Our work is distinct in that we combine Dirichlet process mixture models into analyzing and designing kernels from the spectral domain. Using traditional Gaussian mixture models in kernel analysis [25] imposes restraint on the expressiveness of kernels. Dirichlet process mixture models are flexible models to fit data because the complexity of models grows as more data are observed. By using it to our advantage, we propose the infinite mixture (IM) kernel. The combination of these two traditional Bayesian nonparametric models is one of the most innovative contributions of this work. Theoretically, we prove that with possibly infinite components in mixture models, designed kernels can fit any linear or nonlinear function by arbitrary accuracy. Besides, a Markov Chain Monte Carlo (MCMC) implementation of a hierarchical Bayesian model for spectral analysis is presented and experiments are conducted on real-world temporal data which show the IM kernel's competitiveness in comparison with other kernels and its superiority in extrapolation.

To the best of our knowledge, this is the first trial to combine the Dirichlet process mixture model in designing kernels. The result is promising on account of the computational convenience and expressiveness of Dirichlet process mixture models. Our infinite version has several advantages. (1) We use MCMC methods (collapsed Gibbs sampling specifically) to avoid the local minimum problem which frequently occurs in derivative-based methods like EM used in conventional GMM. (2) The number of Gaussian components K is automatically determined by the provided data, which alleviates the requirement for expertise. On the other hand, it is beneficial for the customization of a kernel for a specific scene. (3) Setting K in advance limits the expressiveness of kernels, while an infinite version makes more details available.

The main contribution of the paper can be summarized in two parts:

- We try to combine a hierarchical Bayesian model with the spectral mixture kernel. Our method shows better and more robust performance.
- The original spectral mixture kernel initializes the mean of each component uniformly between 0 and the Nyquist frequency. We analyze the robustness of the spectral mixture kernel and find this strategy to have fluctuant performance. One of the strongest advantages of our method is that the infinite version has a better solution for initialization and has more stable performance.

In Section 2, we have a brief review of the Dirichlet process and the Gaussian process. We introduce a hierarchical Dirichlet mixture model in Section 3 and use this model to analyze kernels in the spectral

domain and put forward the concept of *the infinite mixture kernel* in Section 4. In Section 5, we conduct multiple experiments to show the robustness and good performance of the IM kernel.

2. Background

2.1. Dirichlet Distribution

Dirichlet distribution is an extension of Beta distribution from bivariate to multivariate and can be expressed as:

$$f(x_1, \dots, x_K; \alpha_1, \dots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i-1}, \quad (1)$$

where $x_1, \dots, x_K > 0$ and $\sum_{i=1}^K x_i = 1$. The Beta function $B(\alpha)$ is the normalizer of the distribution and can be expressed in the form of Gamma function:

$$B(\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}, \quad \alpha = (\alpha_1, \dots, \alpha_K). \quad (2)$$

Thus we can regard a sample from the Dirichlet distribution $(x_1, \dots, x_K)$ as a distribution itself.

The probability density function (PDF) of the gamma distribution is:

$$p(x|\alpha, \theta) = \mathcal{G}(\alpha, \theta) = \frac{x^{\alpha-1} e^{-x/\theta}}{\Gamma(\alpha) \theta^\alpha}. \quad (3)$$

There is another version of the PDF of the gamma distribution which is also widely used:

$$p(x|\alpha, \theta) = \mathcal{G}_R(\alpha, \theta) = \frac{x^{\alpha/2-1} e^{-\alpha x/2\theta}}{\Gamma(\alpha/2) (2\theta/\alpha)^{\alpha/2}}. \quad (4)$$

The transformation between these two versions is:

$$\mathcal{G}(\alpha, \theta) = \mathcal{G}_R(2\alpha, \alpha\theta) \quad (5)$$

$$\mathcal{G}_R(\alpha, \theta) = \mathcal{G}\left(\frac{\alpha}{2}, \frac{2\theta}{\alpha}\right). \quad (6)$$

2.2. Dirichlet Process

Dirichlet processes (DP) are a family of stochastic processes whose observations are probability distributions. A Dirichlet process has two parameters, a base distribution H and a positive real-valued scalar α called scaling or concentration parameter. Given these two parameters and a measurable set S , the Dirichlet process $DP(H, \alpha)$ is a stochastic process whose sample path is a probability distribution over S , such that the following holds:

For any finite partition of S , denoted $B_{i=1}^n$, if

$$G \sim DP(H, \alpha), \quad (7)$$

then

$$((G(B_1), \dots, G(B_n))) \sim \text{Dir}(\alpha H(B_1), \dots, \alpha H(B_n)). \quad (8)$$

The concentration parameter α is considered the pseudo count number and the base distribution H can be considered the prior knowledge of the distribution [26].

2.3. Gaussian Process

The Gaussian process is widely used as a statistical model for modeling functions. In the field of machine learning, we are not especially interested in drawing random function from the prior. There always exist raw training data or evaluation results of a model from a series of experiments iterations. These data provide extra knowledge of the model and incorporating this knowledge into the model is of primary interest. In the Gaussian process regression, we seek to predict $f(x)$ based on limited observations.

We express Gaussian process as:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \quad (9)$$

Suppose $f(x)$ is a real process, we can define the mean function $m(x)$ and the covariance function $k(x, x')$ as:

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)], \\ k(x, x') &= \mathbb{E}[(f(x) - m(x))(f(x') - m(x')))]. \end{aligned} \quad (10)$$

3. A Hierarchical Dirichlet Process Mixture Model

A Gaussian mixture model can be written as:

$$p(y|\mu_1, \dots, \mu_K, \Lambda_1, \dots, \Lambda_K, \pi_1, \dots, \pi_K) = \sum_{i=1}^K \pi_i \mathcal{N}(\mu_i, \Lambda_i^{-1}), \quad (11)$$

where μ_i denotes the mean of the i -th component, Λ_i denotes the precision of the i -th component and π_i denotes the weight or mixture coefficient of the i -th component which satisfies $\sum_{i=1}^K \pi_i = 1$ and $\pi_i \geq 0$ for $i = 1, \dots, K$. A GMM model can be learned using expectation–maximization (EM) algorithm. However, the number of components K is set manually here which is inconvenient in practical applications because K should vary with the complexity of data.

Given hyperparameters for means, precisions and weights respectively, a hierarchical model in high dimension with a potentially infinite number of Gaussian components is implemented. We first consider inference in a finite version with K components and then extend it into infinite hierarchical models as the limiting case of the finite one.

We outline our model in a probabilistic graphical model for clarity. Figure 1 illustrates the hierarchical structure embedded in the model. Nine nodes represent different collections of random variables and one node represents observations. We classify these random variables into hyperparameters, global variables and local variables. The first column contains hyperparameters with priors set by sufficient statistics from the data. The second column contains means, precisions and weights of K components which we call global variables. The node tagged by c_i represents the local variable that indicates which component generates this data point.

In Gaussian mixture models, the data $y_{j=1}^N$ is assumed to come from a generative model:

$$y_j|c_j \sim \mathcal{N}(\mu_{c_j}, \Lambda_{c_j}^{-1}). \quad (12)$$

To establish a hierarchical model, we put priors on μ_i and Λ_i :

$$p(\mu_i|\lambda, r) \sim \mathcal{N}(\lambda, r^{-1}), \quad p(\Lambda_i|\beta, w) \sim \mathcal{G}(\beta, w^{-1}), \quad (13)$$

where $\mathcal{G}$ denotes the gamma distribution. First we have a deeper insight into μ . We give hyperparameters λ and r vague priors as:

$$p(\lambda) \sim \mathcal{N}(\mu_y, \Sigma_y), \quad (14)$$

$$p(r) \sim \mathcal{G}\left(\frac{1}{2}, 2\Sigma_y^{-1}\right) \propto r^{-1/2} \exp\left(\frac{-r\Sigma_y}{2}\right), \quad (15)$$

where $\mu_y = \frac{1}{N} \sum_{j=1}^N y_j$ and $\Sigma_y = \frac{1}{N} \sum_{j=1}^N (y_j - \mu_y)^2$ denote the mean and variance of all observations.

Figure 1. Graphical model representation of an infinite hierarchical mixture model. Nodes denote random variables. Under the condition of high dimensionality, each node can be a collection of multiple random variables. Edges denote dependence. Two blue plates denote replication of global hidden variables and the red plate denotes replication of local hidden variables and observations. Notice that blue plates have potentially infinite number of replications, while the red plate has only N replications corresponding to each data point.

When considering prior in Bayesian inference, the usage of conjugate priors allows the results to be derived in closed form. It is conventional to use Gaussian, gamma and inverse-gamma distribution as conjugate priors for scalar means, precisions and variances, respectively [27]. Considering the Wishart distribution is a generalization of the gamma distribution to multiple dimensions, we use Gaussian, Wishart and inverse-Wishart distribution as conjugate priors for high-dimensional hyperparameters. The posterior distribution of μ given other variables is:

$$p(\mu_i | \lambda, r, \Lambda_i, \mathbf{c}, \mathbf{y}) \sim \mathcal{N}\left(\lambda + \frac{(\bar{y}_i - \lambda)}{n_i \Lambda_i + r} n_i \Lambda_i, (n_i \Lambda_i + r)^{-1}\right), \quad (16)$$

where $n_i = \sum_{j=1}^N \delta(c_j, i)$ denotes the number of data points generated by the i -th component and $\bar{y}_i = \frac{1}{n_i} \sum_{j: c_j=i} y_j$ denotes the mean of observations in this component. As to hyperparameters, taking Equation (14) and Equation (15) as priors and Equation (13) as likelihood, we get the posteriors:

$$\begin{aligned} p(\lambda | \mu_1, \dots, \mu_K, r) &\sim \mathcal{N}\left(\mu_y + \frac{\sum_{i=1}^K \mu_i - K\mu_y}{\Sigma_y^{-1} + Kr} r, (\Sigma_y^{-1} + Kr)^{-1}\right), \\ p(r | \mu_1, \dots, \mu_K, \lambda) &\sim \mathcal{W}\left(\frac{1}{D} \left(\Sigma_y + \sum_{i=1}^K (\mu_i - \lambda)^\top (\mu_i - \lambda)\right)^{-1}, \frac{K+1}{D}\right), \end{aligned} \quad (17)$$

where $\mathcal{W}$ denotes the Wishart distribution and D denotes the number of dimensions. The posterior of precision and its hyperparameters can be derived in the same way.

The weight of components $\pi_1, \dots, \pi_K$ have a hyperparameter α . We can simply set the prior of α as $p(\alpha^{-1}) \sim \mathcal{G}(\frac{1}{2}, 2)$, which means $p(\alpha) \propto \alpha^{-3/2} \exp(-\frac{1}{2\alpha})$. The joint distribution of N category variables c_j is:

$$p(c_1, \dots, c_N | \pi_1, \dots, \pi_K) = \prod_{i=1}^K \pi_i^{n_i}. \quad (18)$$

Notice that this distribution is a multinomial distribution with permutation rather than combination and the Dirichlet distribution is the conjugate prior of the multinomial distribution in Bayesian statistics. In order to bring as less prior information as possible, we give a symmetric Dirichlet distribution prior for π_i :

$$p(\pi_1, \dots, \pi_K | \alpha) \sim \text{Dirichlet}(\alpha/K, \dots, \alpha/K) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{i=1}^K \pi_i^{\alpha/K-1}. \quad (19)$$

By combining Equations (18) and (19) we can integrate out the weight:

$$p(c_1, \dots, c_N | \alpha) = \frac{\Gamma(\alpha)}{\Gamma(n + \alpha)} \prod_{i=1}^K \frac{\Gamma(\alpha/K + n_i)}{\Gamma(\alpha/K)}. \quad (20)$$

Collapsed Gibbs sampling requires updating one variable each time, so c_j for all $j = 1, \dots, N$ should be considered separately. We can derive from Equation (20) directly:

$$p(c_j = i | \mathbf{c}_{-j}, \alpha) = \frac{n_{-j,i} + \alpha/K}{n - 1 + \alpha}, \quad (21)$$

where $\mathbf{c}_{-j}$ denotes c_j for $i = 1, \dots, j-1, j+1, \dots, N$ and $n_{-j,i}$ denotes the number of data points generated by the i -th component without the j -th data point.

When K rounds towards infinity, Equation (21) become $p(c_j = i | \mathbf{c}_{-j}, \alpha) = n_{-j,i} / (n - 1 + \alpha)$. Notice that the probability of the j -th data point in all existing components is $\sum_{i=1}^K p(c_j = i | \mathbf{c}_{-j}, \alpha) = (n - 1) / (n - 1 + \alpha)$, which is not equal to 1. Thus the probability of this data point belonging to a new component is $p(c_j \neq c_{j'} \text{ for all } j' \neq j | \mathbf{c}_{-j}, \alpha) = \alpha / (n - 1 + \alpha)$. The mean and precision of the new component are sampled from the prior. In the meantime, components are removed when they get empty. After enough iterations, this model tends to be stabilized. The detailed derivation of all formulas in this section can be found in Appendix A.

4. Kernel Analysis in the Spectral Domain

In this section, we analyze kernels in their spectral domain and propose a new kernel called the infinite mixture (IM) kernel. The IM kernel is defined as a kernel with infinite Gaussian mixture components in its spectral domain. Considering the symmetric property of spectral density for stationary kernels, the spectral density of the infinite mixture kernel is designed as:

$$\begin{aligned} \theta(s_i) &\sim \mathcal{N}(\mu_i, \Lambda_i^{-1}), \\ S(s) &= \sum_{i=1}^{\infty} \frac{1}{2} w_i (\theta(s_i) + \theta(-s_i)), \end{aligned} \quad (22)$$

where μ_i , Λ_i and w_i are the mean, precision and weight for the i -th component respectively.

The design of a kernel is of essential importance in Gaussian process. When considering interpolation, we need the covariance function to be smooth enough, which means if two input points are close to each other, we expect the value of the function at those points to be similar. This requires the covariance matrix to have the attribute that if two points are closer, the corresponding element in the matrix will be larger. The squared exponential (SE) kernel, also called the radial basis function kernel, is the most widely used kernel and has the form:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right). \quad (23)$$

It is infinitely differentiable, which means the Gaussian process fitted with this kernel has mean square derivatives in any order and is thus very smooth. A stationary kernel is a function of $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$.

According to the definition, the SE kernel is a typical stationary kernel and can be reformulated as $k(\boldsymbol{\tau}) = \exp(-\|\boldsymbol{\tau}\|^2 / 2\ell^2)$. The application of the Bochner's theorem (Theorem 1) [28,29] enables us to analyze a stationary kernel from its spectral domain.

Theorem 1. A complex-valued function k on $\mathbb{R}^D$ is the function of a weakly stationary mean square continuous complex-valued random process on $\mathbb{R}^D$ if and only if it can be represented as:

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s} \cdot \boldsymbol{\tau}} d\mu(\mathbf{s}), \quad (24)$$

where μ is a positive finite measure.

If μ has a density $S(\mathbf{s})$, that is, $d\mu(\mathbf{s}) = S(\mathbf{s})d\mathbf{s}$, then S is called the spectral density or power spectral of kernel k . Furthermore, when the spectral density exists, the Wiener-Khinchine theorem shows that the covariance function and the spectral density are Fourier duals of each other:

$$k(\boldsymbol{\tau}) = \int S(\mathbf{s}) e^{2\pi i \mathbf{s} \cdot \boldsymbol{\tau}} d\mathbf{s}, \quad S(\mathbf{s}) = \int k(\boldsymbol{\tau}) e^{-2\pi i \mathbf{s} \cdot \boldsymbol{\tau}} d\boldsymbol{\tau}. \quad (25)$$

By substituting the SE kernel in Equation (23) into (25), we can get the corresponding spectral density of the SE kernel as a Gaussian function $S(\mathbf{s}) = (2\pi\ell^2)^{D/2} \exp(-2\pi^2\ell^2\mathbf{s}^\top\mathbf{s})$, where D denotes the dimensionality of $\mathbf{s}$. Note that the variance of process is $k(\mathbf{0}) = \int S(\mathbf{s})d\mathbf{s}$, which is exactly the integral in spectral domain.

Analyzing kernel from a spectral (or frequency) perspective, the characteristic of zero-centered Gaussian in the spectral density of the SE kernel restricts its expressiveness. Wilson and Adams [25] proposed the spectral mixture (SM) kernel as an extension of traditional kernels by using several Gaussian components instead of one zero-centered Gaussian.

The number of components of the SM kernel is set to a fixed value. However, without additional expertise, setting K in advance is not suitable for practical application which gives rise to underfitting or overfitting. When the inherent pattern of data is simple, setting K with a large value will take noise in frequency domain into consideration and result in overfitting. Meanwhile, when the inherent pattern is sophisticated, setting K with a small value will lead to the ignorance of details and result in underfitting.

In order to address this problem, we combine an infinite mixture model into kernel design and propose the infinite mixture kernel under a Bayesian framework. By combining Equations (25) and (22), we get the analytic expression of the IM kernel:

$$k(\boldsymbol{\tau}) = \sum_{i=1}^{\infty} w_i \exp(-2\pi^2\boldsymbol{\tau}^\top\boldsymbol{\Lambda}_i^{-1}\boldsymbol{\tau}) \cos(2\pi\boldsymbol{\tau}^\top\boldsymbol{\mu}_i), \quad (26)$$

where $\sum_{i=1}^{\infty} w_i = 1$ and $w_i \geq 0$.

The IM kernel can assign K automatically as the distribution of data changes. The Wiener's Tauberian theorem (Theorem 2) gives the theory ground to the assertion that an infinite mixture of Gaussians is dense in the set of all possible distributions.

Theorem 2. Let $f \in L_1(\mathbb{R})$ be an integrable function. The span of translations $f_a(x) = f(x + a)$ is dense in $L_1(\mathbb{R})$ if and only if the Fourier transform of f has no real zeros.

This theorem is stronger because it only uses translation transformation to make the span dense in $L_1(\mathbb{R})$ function space. Various popular kernels can be reconstructed by the IM kernel via analyzing their spectral density which shows the expressiveness of the IM kernels from another point of view.

Before applying infinite mixture models, we need to get the empirical spectral density of the data first. From Theorem 1, the spectral density is proportional to the discrete Fourier transform

(DFT) of the squared data. The time complexity of DFT is $O(n^2)$, which can be reduced to $O(n \log n)$ using fast Fourier transform (FFT). FFT is used here to calculate the spectral density. Notice that if the observations are not equidistant, data should be preprocessed. A simple way to preprocess the data is using interpolation methods like linear interpolation, polynomial interpolation and spline interpolation. A more precise and complex approach is using a compressed sensing algorithm [30]. Further experiments show that when the input has a grid structure except for a lack of input in a few ranges, it works well without preprocessing. We refer to it as a quasi-grid structure. The empirical spectral density is taken as an unnormalized distribution and we get observations by slice sampling [31].

5. Experiments

In this section, we conduct a series of experiments to verify the robustness and good performance of the IM kernel on multiple datasets. We first introduce details of the implementation of the hierarchical mixture model and analyze the independence of samples. Then we perform robustness test on the SE kernel and the IM kernel, finding that performance of the SE kernel fluctuates wildly with different random seeds while IM kernel maintains good performance. Finally, we compare IM kernel with other widely used kernels in several datasets. The IM kernel shows smaller negative log-likelihood and mean square error (MSE) than others. Matlab (version 9.1) code to perform all experiments is available on GitHub (<https://github.com/deeperKernelInsight/infinitemixturekernel>). All experiments are conducted on a PC with a 3.20 GHz Hexa-core CPU and 16 GB RAM.

5.1. An MCMC Implementation of the Hierarchical Model

A collapsed Gibbs sampler marginalizes over a part of dependent variables while sampling from other variables. This strategy is widely used in hierarchical Bayesian models like latent Dirichlet allocation (LDA). Collapsing out the Dirichlet distribution in this model helps to accelerate sample steps. We use collapsed Gibbs sampling as the specific MCMC method to sample hyperparameters, local variables and global variables. Meanwhile, the exact order of sampling parameters influences the performance. In the implementation, we sample means and precisions for all components first, with hyperparameters next and indicator variables at last.

We use the airline passenger dataset which records monthly passenger numbers from 1949 to 1961 here to look into this MCMC process. Before the Markov chain becomes stationary, there are a few iterations for “burn-in”. In all experiments below, we set 1000 iterations for “burn-in”. Since collapsed Gibbs sampling draws only one parameter each time, observations close to each other tend to be correlated. Figure 2 plots the autocorrelation function of hyperparameters taking lag as x -axis. From this figure, the parameters are deemed independent of itself after 200 iterations. After the “burn-in” iterations, we record observations every 100 iterations and choose the most suitable one to refine.

Figure 2. Autocorrelation analysis for hyperparameters.

Figure 3a shows the number of components in the process of training with a different number of sampling points. For universality, we set only one component in the beginning. It increases rapidly in the first few iterations and stays stable with slight fluctuations. With the number of sampling points increases, the number of components used to depict the model grows. The primary cause is that models become complex with the growth of sampling points and this is where the flexibility emerges. Figure 3b plots histogram of 100 observations for α in the condition of 10,000 sampling points. The concentrated parameter α takes values around 2.5, which means only $\alpha/(n + \alpha) \approx 0.025\%$ of data points belonging to a new component.

Figure 3. Parameters observed in the process of Markov Chain Monte Carlo (MCMC).

5.2. Robustness Test

Since the learning surface for a spectral mixture kernel is often high multimodal, the original model of the SE kernel is vulnerable for its stern demand for initialization. The SE kernel takes a rather simple and greedy strategy by setting initial weights as $k(0)/K$, initial variances sampled from a truncated Gaussian distribution with a mean equal to the range of the data and initial frequencies sampled from a uniform distribution from 0 to its Nyquist frequency. Figure 4 shows the performance of the SM kernel with different random seeds. It performs well when the random seed takes the value 3,141,879. However, taking 579 and 1457 as random seeds results in bad performance. Notice that there are obvious performance gaps between random seeds which indicates that the simple strategy used by the SM kernel is not stable and is thus not a good option. Performances of the IM kernel with different random seeds are not plotted because results remain stable and overlap in the chart. By switching the number of sampling points, the performances are slightly different. With a small number of sampling points, kernel tends to show secular trends at the expense of details. Meanwhile, with a large number of sampling points, kernel tends to show more details while secular trend flattens out.

Figure 4. Performance of the spectral mixture (SM) kernel with different random seeds.

5.3. Performance on Multiple Datasets

The datasets we employ are publicly available and has been used in previous research [24]: (a) monthly gasoline demand in Ontario; (b) monthly death tolls from bronchitis in the UK; (c) mean monthly air temperature; (d) weekday bus ridership in Iowa city. We divide these datasets each into two parts with 2/3 data as the training set and others as the testing set. We attempt to give fair treatment to these kernels by initializing hyperparameters with high marginal log-likelihoods and fully training kernels to convergence. The marginal log-likelihood with noise observations can be expressed as:

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^\top \left(K_\theta + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2} \log |K_\theta + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi. \quad (27)$$

We use GPML toolbox (<http://www.gaussianprocess.org>) to refine parameters and evaluate other popular kernels.

Figure 5 shows the predictive distribution of the IM kernel and its spectral density. The top row displays predicting results on the testing set with the training and testing set in blue and black respectively. The mean of the predictive distribution is in red. The IM kernel shows good predictive ability with the testing set located in 95% of the predictive mass shown in shadow. The bottom row shows the log spectral density of the IM kernel in red and the log empirical spectral density in blue. The learned kernel fitted well with the empirical spectral. The log spectral density of kernel in the Temperature dataset shows the ability to filter noise.

Tables 1 and 2 show negative log-likelihood and MSE of the SE kernel, the Matérn kernel, the squared exponential (SE) kernel and the IM kernel with different numbers of sampling points. In experiments, we set numbers of sampling points as 5000, 10,000 and 50,000. Only the standard deviations of the IM and SM kernels are recorded in the brackets due to the space limitation. The IM kernel shows better performance than others both in negative log-likelihood and in MSE. Meanwhile, when the number of sampling performance increases, the result turns out better. While a larger number of sampling points needs a longer time to sample and inference, the trade-off between accuracy and time cost is concerned in practical application.

Table 1. Negative Log-Likelihood. (The results with the best performance are shown in bold).

Dataset	Unit	IM			SM	Matérn	SE
		5×10^3	1×10^4	5×10^4			
Gasoline	1×10^3	1.31(0.09)	1.31(0.07)	1.27(0.06)	14.3(0.27)	1.49	1.44
Death toll	1×10^2	8.55(0.14)	8.49(0.20)	8.48(0.10)	9.66(0.43)	28,500,000	11.2
Temperature	1×10^2	3.80(0.12)	3.70(0.15)	3.71(0.15)	3.68(0.29)	5.71	4.74
Ridership	1×10^2	6.57(0.23)	6.96(0.26)	6.48(0.14)	7.01(0.33)	44,100,000	7.66

Table 2. Mean Square Error (MSE). (The results with the best performance are shown in bold).

Dataset	Unit	IM			SM	Matérn	SE
		5×10^3	1×10^4	5×10^4			
Gasoline	1×10^8	3.50(0.21)	3.27(0.19)	2.94(0.13)	6.18(0.76)	335,000,000	394
Death toll	1×10^4	1.74(0.06)	1.67(0.03)	1.83(0.03)	6.35(0.16)	1,040,000	80.7
Temperature	1	7.05(0.10)	6.75(0.32)	5.89(0.10)	6.85(0.69)	81.1	1850
Ridership	1×10^6	2.32(0.08)	3.12(0.11)	1.14(0.13)	2.64(0.18)	750,000	52.0

Figure 5. Performance of the infinite mixture (IM) kernel on a bunch of datasets.

6. Conclusions

We introduced the infinite mixture kernel by combining a hierarchical Bayesian model into the analyze of the spectral domain. The extensive experiments have shown the robustness and better performance of our method than the finite version. The results are competitive with popularly used kernels and the spectral density of the IM kernel provides an extra interpretation of the inherent pattern of the data.

In future work, one can combine the IM kernel with approximate inference algorithms. What's more, in this paper we only analyze the kernel of a specific stochastic process. Although the Gaussian distribution and the Gaussian processes are widely used in many different fields, there are many other distributions and their corresponding stochastic processes, for example, the q-Gaussian distribution and the distributions which emerge in the anomalous diffusion contexts and their corresponding stochastic processes. Analyzing the performance of these unconventional distributions and their corresponding stochastic processes can result in interesting discoveries. We believe that kernel analysis from a spectral perspective helps to design more powerful kernels while a deeper investigation is still needed for future research.

Author Contributions: conceptualization, J.T.; methodology, J.T.; software, J.T.; validation, J.T.; formal analysis, P.Y.; investigation, P.Y.; writing—original draft preparation, J.T.; writing—review and editing, J.T. and P.Y.; supervision, D.H.

Funding: This work was partially supported by the National Natural Science Foundation of China (grants 61803375 and 91648204), by the National Key Research and Development Program of China (grants 2017YFB1001900 and 2017YFB1301104) and by the National Science and Technology Major Project.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

GP	Gaussian process
GMM	Gaussian mixture models
DPMM	Dirichlet processes mixture model
PDF	Probability density function
CDF	Cumulative distribution function
SM	Spectral mixture kernel
SE	Squared exponential kernel
IM	Infinite mixture kernel

Appendix A. Hierarchical Bayesian Models

A traditional Gaussian mixture model is expressed as:

$$p(y|\mu_1, \dots, \mu_k, s_1, \dots, s_k, \pi_1, \dots, \pi_k) = \sum_{j=1}^k \pi_j \mathcal{N}(\mu_j, s_j^{-1}), \quad (\text{A1})$$

where μ_j , s_j , and π_j are the mean, the precision, and the proportional hyperparameters for the j -th components.

The mean μ_j is given Gaussian prior:

$$p(\mu_j|\lambda, r) \sim \mathcal{N}(\lambda, r^{-1}), \quad (\text{A2})$$

where λ and r are hyperparameters.

The hyperparameters λ and r are also given Gaussian and Gamma priors:

$$p(\lambda) \sim \mathcal{N}(\mu_y, \sigma_y^2), \quad (\text{A3})$$

$$p(r) \sim \mathcal{G}(1, \sigma_y^{-2}) \propto r^{-1/2} \exp(-r\sigma_y^2/2), \quad (\text{A4})$$

where μ_y and σ_y^2 are the mean and variance of all observations.

Take Equation (A2) as the prior and Equation (A1) as the likelihood function, we can get the posterior distribution for the means μ_j :

$$\begin{aligned}
 p(\mu_j | \mathbf{y}_j, s_j, \lambda, r) &\propto p(\mu_j | s_j, \lambda, r) p(\mathbf{y}_j | \mu_j, s_j, \lambda, r) \\
 &\propto p(\mu_j | \lambda, r) p(\mathbf{y}_j | \mu_j, s_j) \\
 &\propto \mathcal{N}(\mu_j | \lambda, r^{-1}) \prod_{i=1}^{n_j} \mathcal{N}(y_{j,i} | \mu_j, s_j^{-1}) \\
 &\propto \exp \left(-\frac{r}{2} (\mu_j - \lambda)^2 - \frac{s_j}{2} \sum_{i=1}^{n_j} (y_{j,i} - \mu_j)^2 \right) \\
 &\propto \exp \left(-\frac{s_j n_j + r}{2} \left(\mu_j - \frac{r\lambda + s_j n_j \bar{y}_j}{s_j n_j + r} \right)^2 \right) \\
 &\sim \mathcal{N} \left(\frac{r\lambda + s_j n_j \bar{y}_j}{s_j n_j + r}, (s_j n_j + r)^{-1} \right),
 \end{aligned} \tag{A5}$$

where $\mathbf{y}_j$ is all observations belonging to class j , n_j is the number of these observations, and $\bar{y}_j = \frac{1}{n_j} \sum y_{j,i}$.

Take Equation (A3) as the prior and Equation (A2) as the likelihood function, we get the posterior distribution for the hyperparameter λ :

$$\begin{aligned}
 p(\lambda | \mu_1, \dots, \mu_k, r) &\propto p(\lambda | r) p(\mu_1, \dots, \mu_k | \lambda, r) \\
 &\propto p(\lambda) \prod_{j=1}^k p(\mu_j | \lambda, r) \\
 &\propto \mathcal{N}(\lambda | \mu_y, \sigma_y^2) \prod_{j=1}^k \mathcal{N}(\mu_j | \lambda, r^{-1}) \\
 &\propto \exp \left(-\frac{1}{2\sigma_y^2} (\lambda - \mu_y)^2 - \frac{r}{2} \sum_{j=1}^k (\mu_j - \lambda)^2 \right) \\
 &\propto \exp \left(-\frac{kr + \sigma_y^{-2}}{2} \left(\lambda - \frac{\mu_y \sigma_y^{-2} + r \sum_{j=1}^k \mu_j}{kr + \sigma_y^{-2}} \right)^2 \right) \\
 &\sim \mathcal{N} \left(\frac{\mu_y \sigma_y^{-2} + r \sum_{j=1}^k \mu_j}{kr + \sigma_y^{-2}}, (kr + \sigma_y^{-2})^{-1} \right).
 \end{aligned} \tag{A6}$$

Take Equation (A4) as the prior and Equation (A2) as the likelihood function, we get the posterior distribution for the hyperparameter r :

$$\begin{aligned}
p(r|\mu_1, \dots, \mu_k, \lambda) &\propto p(r) \prod_j p(\mu_{j=1}^k | \lambda, r) \\
&\propto \mathcal{G}_R \left(r | 1, \sigma_y^{-2} \right) \prod_{j=1}^k \mathcal{N}(\mu_j | \lambda, r^{-1}) \\
&\propto r^{-\frac{1}{2}} \exp \left(-\frac{\sigma_y^2}{2} r \right) \prod_{j=1}^k r^{\frac{1}{2}} \exp \left(-\frac{r}{2} (\mu_j - \lambda)^2 \right) \\
&\propto r^{\frac{k-1}{2}} \exp \left(-\frac{1}{2} \left(\sigma_y^2 + \sum_{j=1}^k (\mu_j - \lambda)^2 \right) r \right) \\
&\sim \mathcal{G}_R \left(k+1, (k+1) \left(\sigma_y^2 + \sum_{j=1}^k (\mu_j - \lambda)^2 \right)^{-1} \right) \\
&\sim \mathcal{G} \left(\frac{k+1}{2}, 2 \left(\sigma_y^2 + \sum_{j=1}^k (\mu_j - \lambda)^2 \right)^{-1} \right).
\end{aligned} \tag{A7}$$

The precision of each component s_j is given Gamma prior:

$$p(s_j | \beta, w) \sim \mathcal{G}(\beta, w^{-1}), \tag{A8}$$

where β and w are hyperparameters with inverse Gamma and Gamma priors:

$$p(\beta^{-1}) \sim \mathcal{G}(1, 1) \implies p(\beta) \propto \beta^{-3/2} \exp(-1/(2\beta)), \tag{A9}$$

$$p(w) \sim \mathcal{G}(1, \sigma_y^2). \tag{A10}$$

Take Equation (A8) as the prior and Equation (A1) as the likelihood function, we can get the posterior distribution for the means s_j :

$$\begin{aligned}
p(s_j | \mathbf{y}_j, \mu_j, \beta, w) &\propto p(s_j | \mu_j, \beta, w) p(\mathbf{y}_j | \mu_j, s_j, \beta, w) \\
&\propto p(s_j | \beta, w) p(\mathbf{y}_j | \mu_j, s_j) \\
&\propto \mathcal{G}_R(s_j | \beta, w^{-1}) \prod_{i=1}^{n_j} \mathcal{N}(y_{j,i} | \mu_j, s_j^{-1}) \\
&\propto s_j^{\frac{\beta}{2}-1} \exp \left(-\frac{\beta w}{2} w_j \right) \prod_{i=1}^{n_j} s_j^{\frac{1}{2}} \exp \left(-\frac{s_j}{2} (y_{j,i} - \mu_j)^2 \right) \\
&\propto s_j^{\frac{\beta}{2}-1+\frac{n_j}{2}} \exp \left(-\frac{w\beta}{2} s_j - \frac{1}{2} \sum_{i=1}^{n_j} (y_{j,i} - \mu_j)^2 s_j \right) \\
&\sim \mathcal{G}_R \left(\beta + n_j, (\beta + n_j) \left(w\beta + \sum_{i=1}^{n_j} (y_{j,i} - \mu_j)^2 \right)^{-1} \right).
\end{aligned} \tag{A11}$$

Take Equation (A10) as the prior and Equation (A8) as the likelihood function, we get the posterior distribution for the hyperparameter w :

$$\begin{aligned}
p(w|s_1, \dots, s_k, \beta) &\propto p(w|\beta)p(s_1, \dots, s_k|w, \beta) \\
&\propto p(w) \prod_{j=1}^k p(s_j|w, \beta) \\
&\propto \mathcal{G}_R(w|1, \sigma_y^2) \prod_{j=1}^k \mathcal{G}_R(s_j|\beta, w^{-1}) \\
&\propto w^{-\frac{1}{2}} \exp\left(-\frac{w}{2\sigma_y^2}\right) \prod_{j=1}^k w^{\frac{\beta}{2}} \exp\left(-\frac{\beta s_j}{2} w\right) \\
&\propto w^{\frac{k\beta-1}{2}} \exp\left(-\frac{1}{2}(\sigma_y^{-2} + \beta \sum_{j=1}^k s_j)w\right) \\
&\sim \mathcal{G}_R\left(k\beta + 1, (k\beta + 1)(\sigma_y^{-2} + \beta \sum_{j=1}^k s_j)^{-1}\right).
\end{aligned} \tag{A12}$$

Take Equation (A9) as the prior and Equation (A8) as the likelihood function, we get the posterior distribution for the hyperparameter β :

$$\begin{aligned}
p(\beta|s_1, \dots, s_k, w) &\propto p(\beta|w)p(s_1, \dots, s_k|\beta, w) \\
&\propto p(\beta) \prod_{j=1}^k p(s_j|\beta, w) \\
&\propto \beta^{-\frac{3}{2}} \exp\left(-\frac{1}{2\beta}\right) \prod_{j=1}^k \mathcal{G}_R(s_j|\beta, w^{-1}) \\
&\propto \beta^{-\frac{3}{2}} \exp\left(-\frac{1}{2\beta}\right) \prod_{j=1}^k \frac{s_j^{\frac{\beta}{2}-1} e^{-\frac{\beta w s_j}{2}}}{\Gamma\left(\frac{\beta}{2}\right) \left(\frac{2}{w\beta}\right)^{\frac{\beta}{2}}} \\
&\propto \Gamma\left(\frac{\beta}{2}\right)^{-k} \beta^{-\frac{3}{2}} \left(\frac{w\beta}{2}\right)^{\frac{k\beta}{2}} \exp\left(-\frac{1}{2\beta} - \frac{\beta w}{2} \sum_j s_j\right) \prod_{j=1}^k s_j^{\frac{\beta}{2}-1}.
\end{aligned} \tag{A13}$$

The posterior distribution of β is in analytical form but not a standard form. It can be proved that $p(\log(\beta)|s_1, \dots, s_k, w)$ is log-concave. Thus we can use adaptive rejection sampling method [32] to sample this distribution.

The proportional hyperparameters $\pi_1, \dots, \pi_k$ are given a symmetric Dirichlet prior:

$$p(\pi_1, \dots, \pi_k|\alpha) \sim \text{Dirichlet}(\alpha/k, \dots, \alpha/k) = \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{k})^k} \prod_{j=1}^k \pi_j^{\frac{\alpha}{k}-1}. \tag{A14}$$

The probability of all indicators c_j under the condition where $\pi_1, \dots, \pi_k$ are fixed is:

$$p(c_1, \dots, c_n|\pi_1, \dots, \pi_k) = \prod_{j=1}^k \pi_j^{n_j}. \tag{A15}$$

We can integrate out the proportional hyperparameters and get the distribution of each indicator conditioning on α :

$$\begin{aligned}
 p(c_1, \dots, c_n | \alpha) &= \int p(c_1, \dots, c_n | \pi_1, \dots, \pi_k) p(\pi_1, \dots, \pi_k) d\pi \\
 &= \int \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{k})^k} \prod_{j=1}^k \pi_j^{\frac{\alpha}{k}-1} \prod_{j=1}^k \pi_j^{n_j} d\pi \\
 &= \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{k})^k} \int \prod_{j=1}^k \pi_j^{n_j + \frac{\alpha}{k} - 1} d\pi \\
 &= \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{k})^k} \frac{\prod_{j=1}^k \Gamma(n_j + \frac{\alpha}{k})}{\Gamma(n + \alpha)} \\
 &= \frac{\Gamma(\alpha)}{\Gamma(n + \alpha)} \prod_{j=1}^k \frac{\Gamma(n_j + \frac{\alpha}{k})}{\Gamma(\frac{\alpha}{k})}.
 \end{aligned} \tag{A16}$$

When using Gibbs sampling methods to sample $\mathbf{c}$, we have to sample each c_i conditioning on others c_i . So we need to get the conditional distribution of c_i :

$$\begin{aligned}
 p(c_i = j | \mathbf{c}_{-i}, \alpha) &= \frac{p(\mathbf{c} | \alpha)}{p(\mathbf{c}_{-i} | \alpha)} \\
 &= \frac{\frac{\Gamma(\alpha)}{\Gamma(n + \alpha)} \prod_{j=1}^k \frac{\Gamma(n_j + \frac{\alpha}{k})}{\Gamma(\frac{\alpha}{k})}}{\frac{\Gamma(\alpha)}{\Gamma(n - 1 + \alpha)} \frac{\Gamma(n_{-i,j} + \frac{\alpha}{k})}{\Gamma(n_j + \frac{\alpha}{k})} \prod_{j=1}^k \frac{\Gamma(n_j + \frac{\alpha}{k})}{\Gamma(\frac{\alpha}{k})}} \\
 &= \frac{n_{-i,j} + \alpha/k}{n - 1 + \alpha},
 \end{aligned} \tag{A17}$$

where $n_{-i,j}$ is the number of the j -th components except the i -th point.

α is given an inverse gamma distribution:

$$p(\alpha^{-1}) \sim \mathcal{G}_R(1, 1) \implies p(\alpha) \propto \alpha^{-\frac{3}{2}} \exp(-\frac{1}{2\alpha}). \tag{A18}$$

Given α , the probability of the number of points for all components, $n_1, \dots, n_k$ is:

$$p(n_1, \dots, n_k | \alpha) = \frac{\alpha^k \Gamma(\alpha)}{\Gamma(n + \alpha)}. \tag{A19}$$

Taking Equation (A18) as the prior and Equation (A19) as the likelihood function, we can get the posterior distribution of α :

$$\begin{aligned}
 p(\alpha | n_1, \dots, n_k) &\propto p(\alpha) p(n_1, \dots, n_k | \alpha) \\
 &\propto \alpha^{-\frac{3}{2}} \exp(-\frac{1}{2\alpha}) \frac{\alpha^k \Gamma(\alpha)}{\Gamma(n + \alpha)}.
 \end{aligned} \tag{A20}$$

References

1. Ferguson, T.S. A Bayesian analysis of some nonparametric problems. *Ann. Stat.* **1973**, *1*, 209–230. [CrossRef]
2. Ishwaran, H.; James, L.F. Gibbs sampling methods for stick-breaking priors. *J. Am. Stat. Assoc.* **2001**, *96*, 161–173. [CrossRef]
3. Blackwell, D.; MacQueen, J.B. Ferguson distributions via Pólya urn schemes. *Ann. Stat.* **1973**, *1*, 353–355. [CrossRef]

4. Pitman, J. *Combinatorial Stochastic Processes*; Technical Report 621; Department Statistics, University of California: Berkeley, CA, USA, July 2002.
5. Aldous, D.J. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII—1983* **1985**, 1117, 1–198.
6. Sudderth, E.B.; Jordan, M.I. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *Advances in Neural Information Processing Systems 21*; Curran Associates, Inc.: New York, NY, USA, 2009; pp. 1585–1592.
7. Gelfand, A.E.; Kottas, A.; MacEachern, S.N. Bayesian nonparametric spatial modeling with Dirichlet process mixing. *J. Am. Stat. Assoc.* **2005**, *100*, 1021–1035. [[CrossRef](#)]
8. Liang, P.; Petrov, S.; Jordan, M.; Klein, D. The infinite PCFG using hierarchical Dirichlet processes. In Proceedings of the 2007 Joint Conference on Empirical Methods In Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 688–697.
9. Johnson, M.; Griffiths, T.L.; Goldwater, S. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2007; pp. 641–648.
10. Navarro, D.J.; Griffiths, T.L. Latent features in similarity judgments: A nonparametric Bayesian approach. *Neural Comput.* **2008**, *20*, 2597–2628. [[CrossRef](#)] [[PubMed](#)]
11. Kemp, C.; Tenenbaum, J.B.; Griffiths, T.L.; Yamada, T.; Ueda, N. Learning systems of concepts with an infinite relational model. In Proceedings of the AAAI'06 21st National Conference on Artificial Intelligence, Boston, MA, USA, 16–20 July 2006; pp. 381–388.
12. Fox, E.; Sudderth, E.B.; Jordan, M.I.; Willsky, A.S. Nonparametric Bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2009; pp. 457–464.
13. Fox, E.; Jordan, M.I.; Sudderth, E.B.; Willsky, A.S. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2009; pp. 549–557.
14. Paisley, J.; Carin, L. Hidden Markov models with stick-breaking priors. *IEEE Trans. Signal Process.* **2009**, *57*, 3905–3917. [[CrossRef](#)]
15. Beal, M.J.; Ghahramani, Z.; Rasmussen, C.E. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2002; pp. 577–584.
16. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer Science & Business Media: Berlin, Germany, 2012; Volume 118.
17. Rasmussen, C.E. Gaussian processes in machine learning. *Adv. Lect. Mach. Learn.* **2003**, 3176; 63–71.
18. Salimbeni, H.; Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 4588–4599.
19. Wilson, A.G.; Hu, Z.; Salakhutdinov, R.R.; Xing, E.P. Stochastic Variational Deep Kernel Learning. In Proceedings of the NIPS'16 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2594–2602.
20. Yang, Z.; Wilson, A.; Smola, A.; Song, L. A La Carte—Learning Fast Kernels. In Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 9–12 May 2015; pp. 1098–1106.
21. Damianou, A.; Lawrence, N. Deep gaussian processes. In Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, Scottsdale, AZ, USA, 29 April–1 May 2013; pp. 207–215.
22. Hinton, G.E.; Salakhutdinov, R.R. Using deep belief nets to learn covariance kernels for Gaussian processes. In NIPS'07 Proceedings of the 20th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1249–1256.
23. Gönen, M.; Alpaydm, E. Multiple kernel learning algorithms. *J. Mach. Learn. Res.* **2011**, *12*, 2211–2268.
24. Duvenaud, D.; Lloyd, J.R.; Grosse, R.; Tenenbaum, J.B.; Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. *arXiv*, **2013**, arXiv:1302.4922.
25. Wilson, A.; Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1067–1075.

26. Antoniak, C.E. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Stat.* **1974**, *2*, 1152–1174. [[CrossRef](#)]
27. Murphy, K.P. Conjugate Bayesian analysis of the Gaussian distribution. *def* **2007**, *1*, 16. Available online: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.126.4603> (accessed on 31 August 2019).
28. Stein, M.L. *Interpolation of Spatial Data: Some Theory for Kriging*; Springer Science & Business Media: Berlin, Germany, 2012.
29. Cox, D.R. *The Theory of Stochastic Processes*; Routledge: Abingdon-on-Thames, UK, 2017.
30. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
31. Neal, R.M. Slice sampling. *Ann. Stat.* **2003**, *31*, 705–767. [[CrossRef](#)]
32. Gilks, W.R.; Wild, P. Adaptive rejection sampling for Gibbs sampling. *J. R. Stat. Soc. C* **1992**, *41*, 337–348. [[CrossRef](#)]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).