Dirichlet Mixture of Gaussian Processes with Split-kernel: An Application to Predicting Disease Course in Multiple Sclerosis Patients

Yijun Zhao
Computer and Information Sciences Department
Fordham University
New York, NY, USA
yzhao11@fordham.edu

Tanuja Chitnis
Department of Neurology
Harvard Medical School
Boston, MA, USA
tchitnis@rics.bwh.harvard.edu

***Abstract*—In many machine learning applications, data are collected from multiple sources and thus may suffer from idiosyncratic biases. A typical approach to modeling such a dataset is via multi-task learning in which a group of tasks are trained simultaneously by exploiting their similarities. However, because different types of bias are often reflected in disjoint subsets of features and may not conform to the same parametric form, traditional multi-task learning algorithms are not sufficient to capture the individualized impact of these subsets. In this paper, we develop a Dirichlet Mixture of Gaussian Processes with Split-kernel (S-DPM) to address this challenge. We establish a mixture model in which each component consists of instances with similar bias characteristics. The prediction task for each component is modeled by a Gaussian process whose kernel is derived from two sub-kernels operating on separate feature spaces. The number of mixing components is inferred automatically by invoking Dirichlet process based clustering on the data. We apply our model to a clinical dataset to predict disease course in Multiple Sclerosis patients and demonstrate its efficacy by comparing it to standard Dirichlet mixture model (DPM) and the non-mixture single Gaussian process model.**

***Index Terms*—Gaussian process, Dirichlet process, nonparametric Bayesian methods, multiple sclerosis, disease course prediction**

I. Introduction

Multi-task learning (MTL) methods learn a classification or regression model for a set of related tasks jointly using a shared representation [1]. They are particularly effective when these tasks share some commonality and are potentially under-sampled if treated individually. MTL methods seek to uncover the distribution for each individual task as a function of the entire feature space. However, for some applications, the underlying distributions may be induced from independent subsets of features which breaks a fundamental assumption of MTL methods. For example, in the medical field, some features in clinical datasets may involve physicians' subjective interpretation of test results whereas others, such as the demographic features, reflect patient preferences in choice of physician (e.g., patients may have a tendency to choose doctors of the same age or gender [2]). The contributions to the underlying distribution from each subset of features in the above example can be independent and thus may not yield to the same parametric form of modeling.

In this paper, we develop an MTL model to capture the idiosyncratic contributions from partitioned feature subspaces. Our motivating domain, predicting disease progression in multiple sclerosis (MS) patients, suffers from both physician subjectivity and patient bias and thus current MTL methods, which focus on exploring the similarities among correlated tasks, are not sufficient to distinguish the individualized impact of different features. We introduce a new approach that consists of a non-parametric mixture of Gaussian processes (GPs) [3] in which the mixing components consist of data with similar bias characteristics. Specifically, each mixing component is fit with a set of "split" kernels, each of which acts on a disjoint feature subspace to model each type of bias separately (in our motivating domain there are two types of bias). Because we have no a priori knowledge to estimate the number of clusters in our mixture model, we apply Dirichlet process (DP) based clustering to infer the number of mixing components in the data.

A typical mixture model such as Gaussian Mixture Model (GMM) [4] partitions the input space into different regions and models each local region separately. Prediction for a new instance is obtained by first deciding which region the new input belongs to, and then applying the parameters of that particular region to obtain the prediction. In our problem, however, a new instance may manifest one or multiple types of bias. Hence each of our mixing components operates on the entire input space. Prediction for a new patient is a weighted average of predictions from all components.

Before illustrating our approach, we first present a brief survey of related work in Section II and review Gaussian and Dirichlet processes in Sections III and IV, respectively. We then present our new Dirichlet Mixture of Split-kernel Gaussian Processes Model (S-DPM) in Section V. In Section VI, we describe the MS prediction task in our motivating domain and evaluate our model on the task by comparing our model to a standard DP mixture model (DPM) often used in MTL and a non-mixture single Gaussian process model. The comparisons demonstrate that S-DPM consistently outperforms the other two approaches. Finally, we conclude in Section VII.

II. Related Work

Nonparametric Bayesian models have been extensively studied to facilitate learning with complex data. In earlier work, Rasmussen & Ghahramani introduced a Mixture of Experts (ME) model, where the individual experts are Gaussian Process (GP) regression models [5]. Using an input-dependent adaptation of the Dirichlet Process, the authors implemented a gating network for an infinite number of Experts. Tresp presented the mixture of Gaussian processes (MGP) model derived from the ME model and can also be used to model general conditional probability densities [6]. In more recent work, Lázaro-Gredilla et al. introduced a mixture of GPs to address the data association problem, i.e., to label a group of observations according to the sources that generated them [7]. Their novel mixture has the distinct characteristic of not using a gating function to determine the association of samples and mixture components. Ross and Dy explore an infinite mixture model on the entire input space with must-link and cannot-link constraints imposed among the data points [8]. Finally, Chatzis and Demiris use the Pitman-Yor process (a variation of DP process) to model the heavy tail behavior of the dataset [9].

Since exact posterior inference is intractable for Bayesian nonparametric models, another active vein of research in this domain is to improve the accuracy of the posterior inference. Sun and Xu proposed a new variational approximation for estimating hidden variables and hyperparameters, and successfully applied their model to the traffic prediction problem. [10]. In a recent study, Trapp et al. presented a deep structured mixture of GP experts that allows exact posterior inference with attractive computational and memory costs [11]. The framework further captures predictive uncertainties consistently better than previous expert-based approximations.

Lastly, nonparametric Bayesian approaches have been applied to various real-world applications. Jackson et al. developed a DPMGP model to classify a set of N signals into an unknown number k of classes for biological sequences (mRNA expression data) [12]. Abbasnejad et al. modeled user preferences as an infinite DP mixture of communities exploiting the observation that user populations often decompose into communities of shared preferences. [13]. In the medical domain, Rodriguez et al. applied a nested Dirichlet process prior to modeling the multi-distribution data collected from different centers in an application to quality of care in US hospitals [14]. Kottas et al. proposed Bayesian nonparametric spatial modeling approaches to study lung cancer incidences from 88 counties in the state of Ohio over an observation period of 21 years [15].

Our S-DPM model is motivated to address the physician subjectivity and patient bias in clinical data of Multiple Sclerosis patients. Our approach differs from the above studies in that prior work explored customizing infinite mixtures in the instance space while we aim to capture the individualized impact of feature subsets. It is also worth noting that our method differs from automatic relevant determination (ARD) [16]. ARD focuses on statistical pruning of irrelevant features and produces a sparse explanatory subset. S-DPM, on the other hand, exploits the fact that our feature space can be partitioned into two opposite groups in terms of their susceptibility to bias.

III. Gaussian Processes

Given n observations $\{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$, a regression problem tries to uncover the function $y = f(x)$ such that for a new input value x_*, we can accurately predict the corresponding value y_*. Often, a particular parametric form (such as linear) of f is stipulated and the parameters are inferred from the data (e.g., via the linear regression). In many practical situations, however, there is no natural way of identifying the form of the function f. A Gaussian process (GP) [3] avoids this difficulty by modeling the y values directly.

Formally, a Gaussian process is a distribution over a set of functions $f : X \rightarrow R$, with the property that when those functions are sampled and then evaluated on a finite set of inputs $\{x_1, x_2, \dots, x_n\} \in X$, the obtained values $\{y_1, y_2, \dots, y_n\} \in R^n$ are normally distributed. Alternatively, one can consider GP as a collection of random variables indexed by the input set X, any finite subset of which has a joint multivariate Gaussian distribution. When there is no information suggesting otherwise, the mean function of GP is often assumed to be 0. Under this assumption, a GP is completely determined by its covariance matrix K (often referred to as GP's "kernel"). We denote:

$$f(x) \sim GP(0, K)$$

Proper choice of the covariance function K determines a GP's flexibility and applicability to particular situations. One popular choice is the Radial Basis Function (RBF) kernel defined as:

$$k(x, x') = \sigma_f^2 exp[\tfrac{-(x-x')^2}{2l^2}]$$

where σ_f and l control the correlation strength and its rate of decay between two variables respectively.

Because observations are often noisy, it is common to consider Gaussian processes with added independent noise:

$$y \quad \sim GP(0, K + \sigma^2 I)$$

The joint normality of any finite set of variables in GP allows derivation of the following formula for the distribution of a sample $y = \{y_1, ..., y_n\}$ augmented by a new value $y* \in R^D$:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K + \sigma^2 I & K_*^T \\ K_* & K_{**} + \sigma^2 I \end{bmatrix}\right)$$

where

$$K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{bmatrix}$$

$$K_* = [k(x_*,x_1), k(x_*,x_2), \dots, k(x_*,x_n)]$$

$$K_{**} = k(x_*,x_*)$$

Standard properties of the multivariate normal distribution give the formula for the conditional expectation of y_* given the sample $Y = \{y_1, ..., y_n\}$:

$$y_*|Y \sim N(\mu_*, \sigma_*^2)$$

where

$$\mu_* = K_*(K + \sigma^2 I)^{-1} Y$$

$$\sigma_*^2 = K_{**} - K_*(K + \sigma^2 I)^{-1} K_*^T$$

Hence the best estimate of y_* and the uncertainty around it are captured by μ_* and σ_*^2. The hyper-parameters of the covariance function ($\{\sigma_f, l\}$ in the RBF case), and the noise variance σ can be estimated by maximizing the following likelihood function:

$$\log p(Y|X,\theta,\sigma^2) = -\tfrac{N}{2}\log 2\pi - \tfrac{1}{2}\log|K + \sigma^2 I| \\ -\tfrac{1}{2} Y^T (K + \sigma^2 I)^{-1} Y.$$

IV. Dirichlet Process Mixture Model

Dirichlet process (DP) is a family of of Bayesian nonparametric models in which the model representations grow as more data are observed [17] [18] [19]. In particular, DP used as a prior in a generative mixture model allows the number of mixing components adapt to the individual dataset automatically. DP can be interpreted as an extension to the traditional generative model with an arbitrary (infinite) number of mixing components. Formally, a Dirichlet process is an infinite dimensional discrete distribution with two parameters α and H denoted as:

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

where H is the base distribution and scalar α is the strength parameter. H serves as the mean of G and α controls the convergence of G towards H. A Dirichlet process can be constructed using the stick-breaking process [20] as follows:

$$\theta_k^* \sim H \qquad v_k \sim Beta(1,\alpha)$$

$$\pi_k^* = v_k \prod_{j=1}^{k-1}(1 - v_j) \qquad G = \sum_{k=1}^{\infty} \pi_k^* \delta(\theta_k^*)$$

where $k = 1, 2, \dots$. and δ is the Dirac function.

A DP mixture model uses $G(\alpha, H)$ as the prior under the Bayesian framework. The entire dataset is modeled as a mixture of components and each component is parameterized by a random draw (θ) from G. Each data observation belongs to one of the components and is modeled as a function of the parameter of its component, i.e., $f_i(\theta_i)$. Specifically,

$$G|\alpha, H \sim DP(\alpha, H) \qquad \theta_i|G \sim G$$

$$x_i|\theta_i \sim f_i(\theta_i)$$

Consider drawing N samples of θ_i (i = 1, 2, …, N) from G. Because G is a discrete distribution, the probability at any given point in the probability space can be non-zero. This implies that the values of the θ_i's will repeat with a positive probability. Hence, these θ_i's exhibit clustering behavior (Polya Urn Scheme). Given the first N samples of θ_i from G, we assume they have produced a set of k distinct values:

$$\Theta^* = \{\theta_1^*, \theta_2^*, \dots, \theta_k^*\} \text{ where } k < N.$$

It can be shown that the next new sample θ_{N+1} can be either a new value drawn from base distribution H with probability $\propto \alpha$ or can be taken from one of the existing members from Θ^* with probability $\propto c_i$, where c_i is the number of times θ_i^* has been repeated. Specifically,

$$\theta_{N+1}|\theta_1, \theta_2, \dots, \theta_N \sim \frac{\alpha}{\alpha + N} H + \sum_{i=1}^{n} \frac{c_i}{\alpha + N} \delta_{\theta_i} \qquad (1)$$

where δ_{θ_i} denotes the distribution concentrated at a single point θ_i. Equation (1) illustrates two important properties of Dirichlet process. First, the concentration parameter α controls the number of distinct values of θ_i's, i.e., the number of mixing components. Second, DP exhibits a "rich get richer" property: the more frequently a θ_i^* has been adopted (i.e., the larger the c_i), the more likely it will be chosen again as next θ_i value.

The learning process of a DP mixture model is to infer the maximum likelihood of θ_i assignments for the mixing components given the observed data. Various techniques such as MCMC sampling [21] and variational inference [22] have been developed. We adopt the latter in this study.

V. Dirichlet Mixture of Split-kernel Gaussian Processes Model

In this section, we outline our approach in detail by first presenting customized Gaussian process with a split kernel (S-GP) for our motivating domain. We then present the Dirichlet mixture of S-GP model (S-DPM) and provide the update equations for posterior inference based on the variational Bayesian framework.

A. *Split-kernel Gaussian Process (S-GP)*

Recall that in our motivating domain, physician subjectivities are reflected in the clinical features that are obtained by different health professionals whereas the patient biases appear in the demographic features of the patients. These two types of bias may not yield to a same form of parametric model. Hence, the relation (f) between the entire feature set (X) and the regression values (Y) is complex and we resort to non-parametric Gaussian process (GP) regression. Not only does GP allow a greater flexibility in the form of f, but also permits different treatments of the input space X. We take advantage of this flexibility and apply different kernels to subsets of features in X to capture their unique characteristics.

Let $X \in R^{N \times D}$ be the observed inputs and $Y \in R^N$ the corresponding regression values, where N is the number of instances, and D is the number of features. We model our data as generated by a mixture of components associated with an infinite set of regression functions $\{f_j\}_{j=1}^{\infty}$. It is convenient to introduce the latent indicator matrix $Z \in R^{N\times\infty}$ which ties the data pairs (x_i, y_i) with the regression functions f_j via the equation $y_i = f_j(x_i)$ (i.e., $Z(i,j) = 1$ and each row contains a single non-zero entry). Each of the regression functions is a path drawn from a GP:

$$f_i \sim GP(0, K_i)$$

We further divide our feature space into

$$X^{N\times D} = (X^{N\times D^1},\ X^{N\times D^2}) \text{ where } D = D^1 \cup D^2$$

For our domain, D^1 contains features capturing patient bias such as age, gender, etc., while D^2 contains features reflecting physician subjectivity (e.g., the interpreted clinical tests). In what follows we denote $X^1 = X^{N\times D^1}$and $X^2 = X^{N\times D^2}$.

To account for differences in our treatment of X^1 and X^2 we impose a special form on the covariance matrix:

$$\begin{aligned} & K = K^1 + K^2 \\ \text{where} \quad & K^1(X_1, X_2) = \sigma_1^2 exp[\tfrac{-(X_1^1 - X_2^1)^2}{2l_1^2}] \\ & K^2(X_1, X_2) = \sigma_2^2 exp[\tfrac{-(X_1^2 - X_2^2)^2}{2l_2^2}] \end{aligned}$$

The relative contribution of K^1 and K^2 is regulated by the magnitudes of σ^1 and σ^2. The Gaussian process generating each regression function f_j is then:

$$f_j \sim GP(0, K_j^1 + K_j^2)$$

B. Dirichlet Mixture of S-GP Models (S-DPM)

As discussed earlier, we do not have the domain knowledge to infer the number of distinct clusters in our data. To circumvent this difficulty, we impose a Dirichlet process as the prior in our mixture model. Since a GP can be viewed as a distribution over functions, we let the base distribution H in our DP to be a zero mean Gaussian process [23]. In our case we require, however, that H is an S-GP – a Gaussian process with a "split" kernel.

Following the notations we used in Sections III and IV, our model can be specified as follows:

$$G \sim DP(\alpha, H) \qquad f|K^1, K^2 \sim G$$
$$y|f, x, \sigma \sim N(f(x), \sigma^2)$$

Data generated by the above model are naturally partitioned (or clustered) by the distinct functions f drawn from H. Dirichlet process properties imply that the number of clusters (our mixing components) grows as new data are observed. The plate diagram in Figure 1 and the data generation process below describe our model in detail:

1. Draw $v_k|\alpha \sim Beta(1, \alpha), k = \{1, 2, ...\}$
2. Draw $f_k|K^1, K^2 \sim H, k = \{1, 2, ...\}$

For the jth data point:

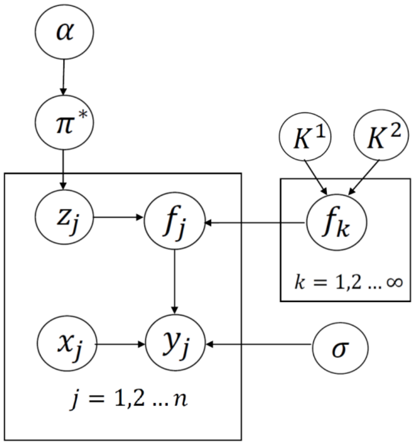

Fig. 1. Plate diagram for S-DPM model

(a) Draw $z_j|\{v_1, v_2, ...\} \sim Mult(\pi(v))$ where $\pi_k(v) = v_k \prod_{j=1}^{k-1}(1 - v_j)$

(b) Draw $y_j|f_j, z_j, x_j, \sigma \sim N(f_{z_j}(x_j), \sigma^2)$

C. Inference

Inference for a DP mixture model is typically conducted using MCMC [21] or variational approximations [22] because in most cases the desired posterior distributions are analytically intractable. The variational approach can be more advantageous due to its scalability and guaranteed local convergence. Here we use the mean-field variational inference outlined in [22]. A similar derivation of the posterior approximation and corresponding update rules via variational inference have been conducted in [8] and [9]. Here we follow their notation but replace the single kernel with a combination of two separate kernels.

As described above, our model employs the latent variables $Z = \{z_1, z_2, \dots\}$ (z_i's are the rows of the indicator matrix Z mentioned previously), $V = \{v_1, v_2, \dots, \}$ and $F = \{f_1, f_2, \dots\}$. Hyper-parameters are σ (the independent Gaussian noise) and $\theta_0 = \{\sigma_0^1, \sigma_0^2, l_0^1, l_0^2\}$ (the kernel parameters of H). The inference algorithm iteratively computes the values of latent variables until convergence. The joint distribution $P(Y, F, Z, V)$ can be factored as follows:

$$p(Y, F, Z, V) = p(Y|F, Z)p(F)p(Z|V)p(V|\alpha) \qquad (2)$$

where

$$p(Y|F, Z) = \prod_{n=1}^{N}\prod_{k=1}^{\infty} N(y_n|f_k^n, \sigma^2)^{Z_{n,k}}$$

$$p(F) = \prod_{k=1}^{\infty} N(f_k|0, K_k^1 + K_k^2)$$

$$p(Z|V) = \prod_{n=1}^{N}\prod_{k=1}^{\infty}\left(v_k \prod_{j=1}^{k-1}(1 - v_j)\right)^{Z_{n,k}}$$

$$p(V|\alpha) = \prod_{k=1}^{\infty} \text{Beta}(v_k|1, \alpha)$$

Our goal is to infer the posterior distribution $p(F, Z, V|Y)$. In variational inference, the posterior distribution $p(F, Z, V|Y)$ is approximated by a computationally convenient distribution $q(F, Z, V)$ which minimizes the Kullback–Leibler (KL) divergence $D_{KL}(q||p)$. Denoting $\Psi = \{F, Z, V\}$, we can write:

$$\begin{aligned} D_{KL}(q||p) &= -\sum_{\Psi} q(\Psi) \log \tfrac{p(\Psi|Y)}{q(\Psi)} \\ &= -\sum_{\Psi} q(\Psi) \log \tfrac{p(\Psi,Y)}{q(\Psi)} + \log p(Y) \\ &= -\mathcal{L}(q) + \log p(Y) \end{aligned}$$

The $\mathcal{L}(q)$ term in the above equation is often referred to as the *variational free energy*. We now have:

$$D_{KL}(q||p) + \mathcal{L}(q) = \log p(Y)$$

Because $p(Y)$ does not depend on q, maximizing $\mathcal{L}(q)$ minimizes the $D_{KL}(q||p)$. In the mean-field approach, maximization of $\mathcal{L}(q)$ is facilitated by the assumption that the posterior distribution q factorizes with respect to the latent variables, and that the factors have the same functional form as the factors of p. Specifically,

$$\begin{aligned} p(Z, V, F|Y) &\approx q(Z) * q(V) * q(F) \\ &= \prod_{n=1}^{N} q_{\phi_n}(z_n) \prod_{k=1}^{\infty} q_{\gamma_k}(v_k) \prod_{k=1}^{\infty} q_{\tau_k}(f_k) \\ &\approx \prod_{n=1}^{N} q_{\phi_n}(z_n) \prod_{k=1}^{T-1} q_{\gamma_k}(v_k) \prod_{k=1}^{T} q_{\tau_k}(f_k) \end{aligned} \tag{3}$$

where $q_{\gamma_t}(v_t)$ are beta distributions, $q_{\tau_t}(f_t)$ are S-GPs, and $q_{\phi_n}(z_n)$ are multinomial distributions. In the last line of (3) we truncated the infinite products to contain T factors. Hence the variational parameters are

$$\nu = \{\phi_1, \ldots, \phi_N, \tau_1, \ldots, \tau_T, \gamma_1, \ldots, \gamma_{T-1}\}$$

The truncation is performed on the variational distribution q only, and could be considered as an additional constraint on its form. The original model remains unaffected and retains the infinite number of components. The truncation parameter T can be adjusted according to to the needs of a particular application.

Using the calculus of variations that for each of the factors $q(Z)$, $q(V)$, $q(F)$, the optimal distribution q^* minimizing the KL divergence is given by:

$$\ln(q^*(\psi|Y)) = \mathrm{E}_{-\psi}[\ln(p(\Psi, Y))] + \mathit{constant}$$

where $\psi \in \Psi = \{Z, V, F\}$. Hence, from equation (1) we obtain:

$$\begin{aligned} \ln(q^*(F)) &= \mathrm{E}_{Z,V}\{\ln p(Z, V, F, Y)\} \\ &= \mathrm{E}_{Z,V}\{\ln p(F|X)p(Y|F, Z)\} + \mathit{const} \\ &= \ln p(F|X) + \mathrm{E}_Z\{\ln p(Y|F, Z)\} + \mathit{const} \end{aligned}$$

$$\begin{aligned} \ln(q^*(Z)) &= \mathrm{E}_{F,V}\{\ln p(Z, V, F, Y)\} \\ &= \mathrm{E}_{F,V}\{\ln p(Y|F, Z)p(Z|V))\} + \mathit{const} \\ &= \mathrm{E}_F\{\ln p(Y|F, Z)\} + \mathrm{E}_V\{\ln p(Z|V)\} + \mathit{const} \end{aligned}$$

$$\begin{aligned} \ln(q^*(V)) &= \mathrm{E}_{F,Z}\{\ln p(Z, V, F, Y)\} \\ &= \mathrm{E}_{F,Z}\{\ln p(Z|V)p(V|\alpha))\} + \mathit{const} \\ &= \mathrm{E}_{F,Z}\{\ln p(Z|V)\} + \ln p(V|\alpha) + \mathit{const} \end{aligned}$$

The above formulas lead to the following update rules for each of the variational distributions q:

$$q^*(F) = \prod_{k=1}^{T} \mathcal{N}(f_k|\mu_k, \Sigma_k)) \tag{4}$$

where

$$\begin{aligned} &\Sigma_k = (K_k(X, X)^{-1} + B_k)^{-1} \\ &K_k(X, X) = K_k^1(X, X) + K_k^2(X, X) \\ &\mu_k = \Sigma_k B_k Y \\ &B_k = \tfrac{1}{\sigma^2} \begin{bmatrix} \mathrm{E}_Z\{Z\}_{1,k} & 0 & \ldots & 0 \\ 0 & \mathrm{E}_Z\{Z\}_{2,k} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathrm{E}_Z\{Z\}_{N,k} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} q^*(V) = \prod_{k=1}^{T} \mathrm{Beta}(V_k|1 + \sum_{n=1}^{N} \mathrm{E}_Z\{Z\}_{n,k}, \\ \alpha + \sum_{j=k+1}^{T} \sum_{n=1}^{N} \mathrm{E}_Z\{Z\}_{n,j}) \end{aligned} \tag{5}$$

$$q^*(Z) = \prod_{n=1}^{N} \prod_{k=1}^{T} r_{n,k}^{Z_{n,k}} \tag{6}$$

where

$$\begin{aligned} r_{n,k} &= \frac{\rho_{n,k}}{\sum\limits_{k=1}^{T} \rho_{n,k}} \\ \ln \rho_{n,k} &= \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(Y_n^2 - 2Y_n \mathrm{E}_F(F_n^k) \\ &\quad + \mathrm{E}_F({F_n^k}^2) + \mathrm{E}_V(\ln V_T) \\ &\quad + \sum_{j=1}^{k-1} \mathrm{E}_V(\ln(1 - V_j)) \end{aligned}$$

We update (4), (5), (6) iteratively until convergence. The posterior distribution of $\{F, V, Z\}$ is given by (3).

Having calculated the latent variables, we can forecast the regression value y_* on a new input point x_* and provide a measure of uncertainty $\sigma^2_{y_*}$ around y_*:

$$y_* = \sum_{k=1}^{T} (V_t * f_k(x_*)) \qquad \sigma^2_{y_*} = \sum_{k=1}^{T} (V_k^2 * (\sigma_*^k)^2)$$

VI. Experimental results

In this section, we first describe the motivating task of predicting disease course in Multiple Sclerosis patients. We then present the performance evaluation of the S-DPM model.

A. Predicting Disease Course of Multiple Sclerosis

MS is an autoimmune disease of the central nervous system in which the immune system attacks the myelin sheath (a fatty layer of substance protecting the nerves), resulting in loss/blockage of signals from the brain [24]. Patients suffer from various levels of disability, and the rate of disability accumulation varies across patients. The machine learning goal for this domain is to predict which patients will accumulate disability and which are likely to remain without disability accumulation after five years from study entry. The level of MS disability is measured by the Expanded Disability Status Scale (EDSS) score [25], [26]. Patients who have a high likelihood of accumulating disability in five years should be treated more aggressively and monitored more closely. But, aggressive treatment carries significant potential side effects, thus it is critical to be able to make this prediction accurately. The specific goal of our project is to predict whether the patients' EDSS scores will increase by at least two at the five year mark using information from the first two years of clinical visits.

Our dataset consists of 574 patients currently enrolled in the CLIMB study [27], a large-scale, long-term study of patients with MS. These patients have been monitored for at least five years and each patient's data includes demographics (e.g., age and gender) and extracted MRI scan image data (e.g., lesion volume and BPF). In addition, patients in the study have a clinical visit every six months, which includes a complete neurological exam including a measurement of the subject's disability based on the EDSS. The EDSS is calculated by combining information from seven functional system (FS) scales that measure specific aspects of the patient using separate ordinal scales that range from 0-6. To demonstrate the potential subjectivity in scoring of the functional system, one physician might give a specific patient a score of "1" for his/her joint mobility whereas another physician would give the patient a "2". Increases of the EDSS score by at least 2 from initial visit to five year mark would classify the patients as the progressive ("P") class, otherwise, they would be classified as belonging to the non-progressive EDSS ("N") class.

B. Experimental Method

All experiments were conducted by running five independent 10-fold cross-validations, and the calculated accuracies are the averages of the five executions of the program. We predict each patient's EDSS level in five years using the GP components and then convert the predicted value into the "P" or "N" class by thresholding at value 2 (as in the class definition above). In our dataset, the ratio of the total number of "P" class patients to that of "N" class patients is 96:478. Because of the imbalance in the two classes of data, we applied a bagging technique to each training iteration in the cross-validation, and the final decisions were made according to the majority votes [28]. Specifically, we formed ten "bags" of data, each of which contained all minority class ("P") instances and an equal number of randomly sampled majority class ("N") instances. Ten classifiers were trained using these balanced "bags" of data and prediction on a new instance was obtained using a majority vote of the ten classifiers. Lastly, the hyper-parameters in this study were selected via grid search. Other methods for learning the hyper-parameters (e.g., empirical Bayes) could lead to further improvements and are currently under investigation.

TABLE I
Breakdown of subgroups by initial EDSS scores

	Initial EDSS score	# N	# P
Group I	< 2	309	61
Group II	$\geq$ 2 and <4	131	28
Group III	$\geq$4	38	7

C. Comparison to DPM and Single GP

We compare the S-DPM model's performance to that of the standard non-parametric Dirichlet Process Mixture (DPM) (DPM does not split its kernel). In addition to presenting the results of that comparison, Figure 2 compares the two mixture models' performance to the single Gaussian Process approach.

Our domain experts were interested interested in studying the patients' disease progress with respect to their initial disability level. The motivation for this distinction is that the baseline characteristics of subjects with different levels of disability are likely different (i.e., subjects with higher EDSS scores are likely older, have longer disease duration, and have been previously been exposed to treatment). In addition, the potential treatment options will differ for subjects based on their initial EDSS scores so that the application of the models in the clinic would be based on the initial EDSS score. To this extent, we separate our results into three groups I, II, III depending on their initial EDSS scores' being less than 2, between 2 and 4, and greater than 4 respectively. Table I shows the class distribution for each group.

From Figure 2, we observe in the "P" class (left plot) that the S-DPM model (leftmost bar) outperforms the two other models for all three groups. For the "N" class (right plot), the S-DPM (leftmost bar) outperforms the other two models for Group I and III, but there is no significant difference among the three models for Group II. The nature of Multiple Sclerosis makes prediction in class "P" more difficult than in class "N".

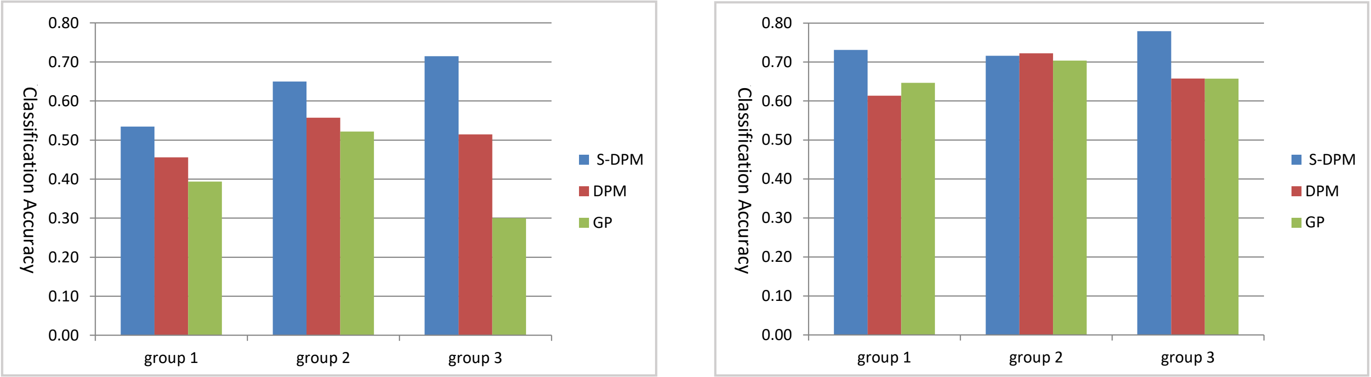

Fig. 2. Comparison of S-DPM, DPM and GP models applied to each group of data. The y-axis is the classification accuracy. The left graph is for the "P" class and the right graph is for the "N" class.

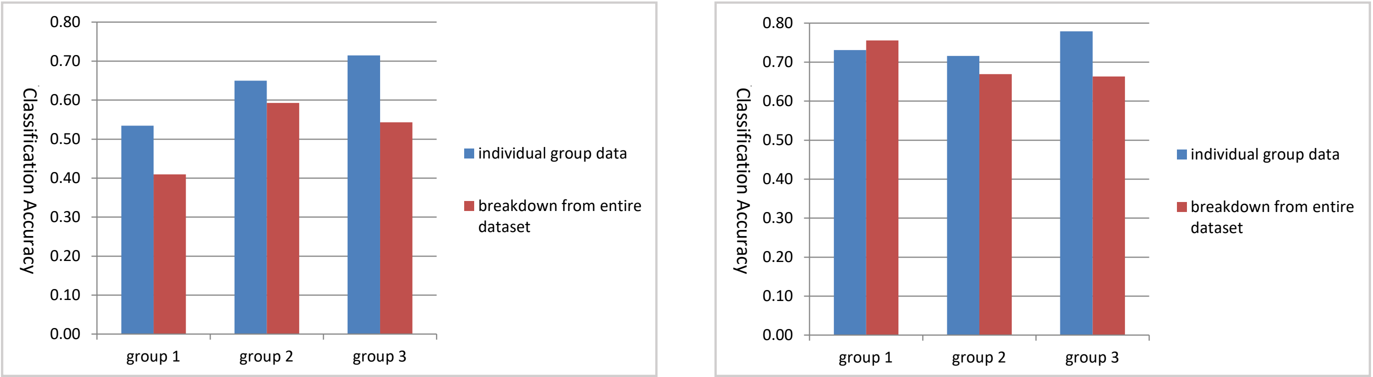

Fig. 3. Comparison of S-DPM applied to each individual group separately (left bars shown in blue) to S-DPM applied to the entire dataset (right bars shown in red). The y-axis is the classification accuracy. The left graph is for the "P" class and the right graph is for the "N" class.

Consequently, we observe significantly higher accuracies in predicting the non-progressive patients than the progressive ones. The difficulty in classifying the "P" patients is potentially linked to higher physician subjectivities in that class' data. The S-DPM model turns out to be more effective in this case; we observe an average of 12% and 23% gain over the DPM and GP models respectively. For the "N" class, the average gains are 8% and 7% respectively.

D. Disease Progression Subgroups versus Entire Dataset

We further conducted an experiment to test whether the DP based framework can cluster the disease progression subgroups better than our domain experts. We applied S-DPM to the entire dataset and calculated the performance achieved for each individual group under a homogeneous learner. Figure 3 compares the results of training the S-DPM model on the entire dataset to training one model for each group separately. The results of each are then shown for each group. We observe that the results from training each group separately (leftmost bars) are better than training on the entire dataset for both the "P" and "N" classes. Therefore, although the DP based clustering algorithm can in theory induce a grouping of the data automatically, it is not sufficient to successfully distinguish the disease progression subgroups in our case. One explanation is that the predictive features vary across patients with different level of disability and we designed our kernels according to the generation of biases in the dataset rather than the predictive strengths of the features.

VII. Conclusion

In this work, we developed a non-parametric mixture model (S-DPM) to address physician subjectivity and patient bias in medical data. Our model is a mixture of GP components that are responsible for the particular biases. In particular, instead of using a kernel that operates on the entire feature space, we partition the features into subsets and employ a separate kernel on each of the subsets in order to capture the individualized contributions. For MS progression prediction, we divide the features into two groups: features reflecting physician subjectivity and patient bias respectively. To have the data determine the number of mixing components, we used the Dirchlet process based clustering and estimated its parameters via variational inference.

One limitation of our approach is the requirement of a predefined partition of the feature space. In our study, there is an unambiguous division of actual and subjective features. In practice, not all datasets exhibit such straightforward distinctive partitions. For future work, we foresee a potential integration with effective tools to identify subjective attributes in real-world data to capitalize on the S-DPM approach presented in this paper.

Lastly, we successfully applied our S-DPM model to predict the disease course of MS patients. Our experimental results confirm that the technical labor needed to take advantage of the split kernel inference is rewarding when the feature space naturally decomposes into two distinct groups. Our approach can be extended to applications in which more than two partitions of feature space are necessary to capture individualized contributions from these sub-spaces.

References

[1] R. Caruana, "Multitask learning." *Springer US*, 1998.

[2] M. Fang, E. McCarthy, and D. Singer, "Are patients more likely to see physicians of the same sex? recent national trends in primary care medicine," *Am J Med. Volume 117, Issue 8*, pp. 575–581, 2004.

[3] C. E. Rasmussen, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71, 2004.

[4] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[5] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," *Advances in neural information processing systems*, vol. 2, pp. 881–888, 2002.

[6] V. Tresp, "Mixtures of gaussian processes," *Advances in neural information processing systems*, pp. 654–660, 2001.

[7] M. Lázaro-Gredilla, S. Van Vaerenbergh, and N. D. Lawrence, "Overlapping mixtures of gaussian processes for the data association problem," *Pattern recognition*, vol. 45, no. 4, pp. 1386–1395, 2012.

[8] J. C. Ross and J. G. Dy, "Nonparametric mixture of gaussian processes with constraints," *Proceeding of the 30th International Conference on Machine Learning*, 2013.

[9] S. P. Chatzis and Y. Demiris, "Nonparametric mixtures of gaussian processes with power-law behavior, ieee transactions on 23.12," *Neural Networks and Learning Systems*, pp. 1862–1871, 2012.

[10] S. Sun and X. Xu, "Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 466–475, 2010.

[11] M. Trapp, R. Peharz, F. Pernkopf, and C. E. Rasmussen, "Deep structured mixtures of gaussian processes," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2251–2261.

[12] E. Jackson, M. Davy, A. Doucet, and W. J. Fitzgerald, "Bayesian unsupervised signal classification by dirichlet process mixtures of gaussian processes," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 3. IEEE, 2007, pp. III–1077.

[13] E. Abbasnejad, S. Sanner, E. V. Bonilla, and P. Poupart, "Learning community-based preferences via dirichlet process mixtures of gaussian processes," in *Twenty-third international joint conference on artificial intelligence*, 2013.

[14] A. Rodriguez, D. B. Dunson, and A. E. Gelfand, "The nested dirichlet process," *Journal of the American statistical Association*, vol. 103, no. 483, pp. 1131–1154, 2008.

[15] A. Kottas, J. A. Duan, and A. E. Gelfand, "Modeling disease incidence data with spatial and spatio temporal dirichlet process mixtures," *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, vol. 50, no. 1, pp. 29–42, 2008.

[16] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[17] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *The Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[18] E. P. Xing, M. I. Jordan, and R. Sharan, "Bayesian haplotype inference via the dirichlet process," *Journal of Computational Biology*, vol. 14.3, pp. 267–284, 2007.

[19] J. Paisley, C. Wang, D. Blei, and M. I. Jordan, "A nested hdp for hierarchical topic models," *arXiv preprint arXiv:1301.3570*, 2013.

[20] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica sinica*, pp. 639–650, 1994.

[21] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of computational and graphical statistics*, vol. 9.2, pp. 249–265, 2000.

[22] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian analysis*, vol. 1.1, pp. 121–143, 2006.

[23] Y. Wang, R. Khardon, and P. Protopapas, "Shift-invariant grouped multi-task learning for gaussian processes," *Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg*, pp. 418–434, 2010.

[24] A. Compston and A. Coles, "Multiple sclerosis," *Lancet 372 (9648): 1502â17. doi:10.1016/S0140-6736(08)61620-7. PMID 18970977*, 2008.

[25] J. Kurtzke, "Rating neurologic impairment in multiple sclerosis: An expanded disability status scale (edss)," *Neurology 33 (11)*, pp. 1444–1452, 1983.

[26] J. Noseworthy, M. Vandervoort, C. Wong, and G. Ebers, "Interrater variability with the expanded disability status scale (edss) and functional systems (fs) in a multiple sclerosis clinical trial," *Neurology 40.6*, p. 971, 1990.

[27] S. Gauthier, B. Glanz, M. Mandel, and H. W. HL, "A model for the comprehensive investigation of a chronic autoimmune disease: The multiple sclerosis climb study," *Autoimmun Rev.*, vol. 5(8), pp. 532–536, 2006.

[28] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos, "Class imbalance, redux," *In Data Mining (ICDM), 2011 IEEE 11th International Conference on.*, pp. 754–763, 2011.