

# Online Regression for Data with Changepoints using Gaussian Processes and Reusable Models

Robert C. Grande, Thomas J. Walsh, Girish Chowdhary, Sarah Ferguson, and Jonathan P. How

**Abstract**—Many prediction, decision making and control architectures rely on online learned Gaussian Process (GP) models. However, most existing Gaussian Process (GP) regression algorithms assume a single generative model, leading to poor predictive performance when the data are nonstationary, i.e. generated from multiple switching processes. Furthermore, existing methods for GP regression over nonstationary data require significant computation, do not come with provable guarantees on correctness and speed, and many only work in batch settings, making them ill-suited for real-time prediction. We present an efficient online GP framework, GP-NonBayesian Clustering (GP-NBC), which addresses these computational and theoretical issues, allowing for real-time changepoint detection and regression using GPs. Our empirical results on two real-world datasets and two synthetic dataset show GP-NBC out-performs state of the art methods for nonstationary regression in terms of both regression error and computation. For example, it outperforms Dirichlet Process Gaussian Process clustering with Gibbs sampling by 98% in computation time reduction while the mean absolute error is comparable.

**Index Terms**—Gaussian Processes, Online, Changepoint Detection

## I. INTRODUCTION

IN many prediction and decision-making applications, it is necessary to create a model of the environment from stochastic measurements only. The Gaussian Process (GP) [1] is a Bayesian nonparametric framework for inference that has gained popularity in a number of applications in machine learning and decision-making, such as regression, [1]; classification, [2]; adaptive control, [3], [4], [5]; and reinforcement learning, [6]. However, while the majority of existing GP algorithms for online data assume a stationary, i.e. time-invariant, distribution, there are numerous online learning domains, such as stock predictions and aircraft control, for which the data may involve *changepoints*: points in time in which abrupt changes occur to the generating distribution itself (for example a market crisis or mechanical failure). When the data contains changepoints, i.e., is non-stationary, any framework modeling the underlying distribution, such as a GP, should be augmented to rapidly identify changepoints, learn new models online, and reuse old models if they become applicable again. Note that herein, non-stationary refers to data with abrupt temporal changes and not to spatially non-stationary GP kernels.

R. Grande, T. Walsh, S. Ferguson, and J. How completed this work with the Department of Aeronautics and Astronautics Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA

G. Chowdhary is an Assistant Professor at University of Illinois at Urbana Champaign, girishc@illinois.edu

A common way to address non-stationarity is to augment the GP input with time or a time-related function [7], [3]. However, using this method causes the GP to forget potentially useful information from earlier phases, even if no change has occurred in the generating distribution. It also cannot make use of previously learned models if they reappear later. The latter is important in several domains, such as pedestrian intent prediction [8] and robotic table tennis [9] in which different behaviors, or models, may be revisited and reused. Previous attempts at online GP changepoint detection algorithms [10], [11] are computationally demanding and thus are poorly suited for applications that require real-time prediction (see experiments section), or lack theoretical guarantees on accuracy and cannot reuse models after a changepoint occurs.

This paper presents a computationally efficient algorithm for online regression over data with changepoints: the Gaussian Process Non-Bayesian Clustering (GP-NBC) algorithm<sup>1</sup>. The speed and performance improvements in GP-NBC can be attributed to the decoupling of the problems of changepoint detection, regression, and model reuse. This decoupling results in efficient online learning in the presence of changepoints. That is, GP-NBC uses fast algorithms for solving each of these subproblems, providing order of magnitude speed-ups over previous methods that combine all three problems into the computation of a single posterior. For inference, GP-NBC uses a non-Bayesian test based on a Generalized Likelihood Ratio (GLR) test [12] to detect changepoints and to re-identify previously seen functions, and performs efficient inference using a budgeted online GP inference algorithm [13]. Non-Bayesian tests do not use a prior distribution and therefore do not require integration over the entire parameter space, resulting in magnitudes faster computation. Additionally, Non-Bayesian statistical tests are better suited for applications in which a prior density function cannot be well-defined over the changepoint frequency.

GP-NBC is efficient and uses orders of magnitude less computation than existing methods, and has been designed for online applications with streaming data. The contributions in this paper are describing GP-NBC and showing these properties both theoretically and empirically. On the theoretical side, we derive polynomial (in the accuracy and GP parameters, including the covering number of the input space) sample complexity bounds on the number of inaccurate predictions by GP-NBC, unlike competing algorithms. Empirically, GP-NBC is validated on two real-world and two synthetic datasets,

<sup>1</sup>In this work, the term a “cluster” refers to each previously defined model over the output domain, and not to a clustering over input domain variables, as is also common in the literature

where it is shown to outperform leading hierarchical GP and changepoint detection algorithms by orders of magnitude in computational efficiency and prediction accuracy. A preliminary version of GP-NBC was first introduced in [14], however the algorithm proposed here has been modified significantly and the theoretical results further developed.

## II. PRELIMINARIES

This section describes the online non-stationary mean prediction problem and reviews related work as well as preliminaries.

### A. Problem Definition

In an *online non-stationary mean prediction problem*, at each new instant  $t$ , a learning agent observes an input  $x_t \in U \subset \mathbb{R}^d$ , that is not i.i.d., and an output  $y_t$  drawn i.i.d. from  $y_t \sim p_i(y | x_t)$ . Before the agent receives  $y_t$ , it observes  $x_t$  and must make a prediction  $\hat{\mu}(x_t)$  of the expected value of the distribution  $\mathbb{E}[y_t | x_t] = f_i(x_t)$  with generating distribution  $p_i(y | x) \in \mathcal{P}$  that changes between changepoints, defined below. The agent then observes a noisy output drawn from the generative distribution  $y_t \sim p_i(y | x_t)$ . Our objective is to minimize the total number of *mistakes* in agent predictions below some tolerance  $\epsilon_E$ . More formally, we define a mistake as any timestep where  $|f_i(x_t) - \hat{\mu}(x_t)| > \epsilon_E$ . This work assumes the mean functions belong to a class of functions  $\mathcal{F}$  that is Lipschitz smooth and modeled by a *GP*. We additionally impose the restriction that the output range is bounded,  $V_m = y_{\max} - y_{\min}$  so that we may use statistical tools such as Hoeffding's inequality. In general, this is not a limiting assumption as for most real-world applications, measurement values are bounded.

Non-stationarity is introduced when the mean of the underlying process changes from  $f_i(x)$  to some  $f_j(x)$  at unknown *changepoints*. Specifically, it is said that a changepoint occurs at time  $t$ , if at time  $t$ , the underlying process mean changes from  $f_i(x)$  to  $f_j(x)$  such that there exists some region  $\tilde{U}$ :  $\forall x \in \tilde{U}, |f_i(x) - f_j(x)| > \epsilon_E$ . We call each period in between changepoints a *phase* and the objective of the agent is to minimize the number of mistakes in each phase. In order to accomplish this, the agent must detect changepoints, learn models, and potentially re-use previously learned models. To solidify the problem further, we restrict our attention to the subclass of problems for which there is a lower bound on the number of samples between changepoints. This ensures that there is sufficient time to learn a new model before detecting a change. Section IV provides sufficient conditions regarding this lower bound and the realizability of each  $f_i(x)$  (i.e., how well a GP captures  $f_i(x)$ ) that lead to a bounded number of mistakes during each phase.

### B. Related Work

Non-stationary data with changepoints are generally handled in the GP literature using one of two categories: online Changepoint Detection (CPD) algorithms, and batch hierarchical Bayesian clustering algorithms. We first review general

CPD algorithms and then focus on online CPD algorithms which use GPs. Then, we review batch hierarchical GP-clustering algorithms.

A variety of algorithms exist for Changepoint detection (CPD) in the field of time series analysis, [12], where often the goal is for the algorithm to infer points in time in which the generative parameters of the time series change rather than attempting to minimize regression error. Several algorithms for computationally efficient CPD exist for slowly drifting functions [15], and for abrupt changes [16]. However, for abrupt changes, existing frameworks either do not use or learn models, or require the set of all possible models be *given* to the algorithm a priori [15]. Alternatively, the framework proposed here considers the scenario where new models may need to be learned online.

Previous work on online CPD and time series monitoring using GPs include GP-Changepoint Detection (GP-CPD) [10] based on the Bayesian online changepoint Detection (BOCPD) algorithm [17] and the method described in [11]. GP-CPD learns a model for every changepoint possibility, and using a prior, performs posterior inference to obtain a distribution over changepoint times. Authors in [11] include the changepoint time in the kernel function and use optimization to determine the changepoint time. However, these approaches focus on the problem of instantaneous regression instead of model learning and require a large amount of computation that scales with the number of data points, which makes them ill-suited for real-time decision applications. The experiments in Section V demonstrate that using such approaches with GP models requires computation that is not suitable for real-time applications, and is outperformed by GP-NBC. Furthermore, there are no associated theoretical guarantees on regression accuracy with these methods.

On the other hand, batch algorithms using hierarchical mixture models mostly model changepoints using BNP methods, such as the Infinite Mixture of Experts (referred to in this paper as DP-GP) [18] and [19], referred to in this paper as MCMC-CRP. DP-GP uses a Dirichlet prior and Gibbs sampling over individual data points to obtain the posterior distribution over model assignments and the number of models. MCMC-CRP uses sampling as well except the sampling distribution is over changepoints. In this case, the Gibbs sampler creates, removes, and shifts changepoints instead of reassigning individual data points to models. While GP methods using sampling have had success in a variety of off-line applications, they require batch data and fail to meet the computational requirements for many real-time prediction applications, as shown in Section V. Furthermore, the Dirichlet prior may not converge to the correct number of models [20] and comes with no guarantees on regression accuracy or rates of convergence to the true posterior. It should be noted that the computational requirements could be reduced through a variational inference based approach [21], [22].

Contrasting with the aforementioned algorithms, GP-NBC is designed specifically for online decision making applications in which datapoints arrive sequentially in time; our theoretical results provide bounds on the number of inaccurate predictions GP-NBC may make per phase and our empirical results show

that GP-NBC is orders of magnitude faster than existing methods. In contrast to the sample efficiency results for GPs in the optimization context [23], the bounds given here are designed specifically for online prediction with non-stationary data. In our paper, we validate these claims over a synthetic and two real datasets as described in Section V.

### C. Gaussian Processes

We model the mean function  $f_i(x)$  using a Gaussian Process (GP). GPs can be viewed as a distribution over functions. That is, a draw from GP is not a single value, but rather a function  $f_i(x) : U \mapsto \mathbb{R}$  drawn from a distribution  $p(f)$ . In particular, a GP is defined as follows: if for any set of locations  $(x_1, \dots, x_n)$ , the resulting marginal distribution  $p(f(x_1), \dots, f(x_n))$  is Gaussian, then the process is a GP [24]. This process has mean  $\mu(x) : U \mapsto \mathbb{R}$  and covariance kernel  $k(x, x') : U \times U \mapsto \mathbb{R}$ . As is common in the GP literature, it is assumed that the prior is a zero mean normal distribution, although it should be noted that this is not limiting in that the posterior is not limited to zero. Additionally, the experimental section uses the Gaussian covariance kernel,  $k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\theta^2}\right)$ , however, the presented approach can be extended to other covariance kernels. The elements of the GP kernel matrix  $K(X, X)$  and kernel vector  $K(X, x_{t+1})$  are defined element-wise as  $K_{i,j} = k(x_i, x_j)$ . In our work, the measurements  $y(x)$  are modeled as being drawn from a normal distribution around the function mean  $f_i(x)$ ,  $y \sim \mathcal{N}(f_i(x), \omega^2)$ . Although different observation distribution models may be used, using a Gaussian model allows for exact inference over  $f_i(x)$ . Mathematically, the prior distribution over the union of the observations and the function mean at a new point  $x_{t+1}$ ,  $f(x_{t+1})$  is

$$\begin{bmatrix} \vec{y} \\ f(x_{t+1}) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \omega^2 I & K(X, x_{t+1}) \\ K^T(X, x_{t+1}) & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right) \quad (1)$$

The posterior distribution of the function mean  $f(x_{t+1})$  at location  $x_{t+1}$ , conditioned on observations  $\vec{y} = [y_1, \dots, y_t]^T$ , can be found using Bayes' Law (Sec. 2.1. of [1]) with mean

$$\hat{\mu}(x_{t+1}) = \alpha^T K(X, x_{t+1}), \quad (2)$$

where  $\alpha = [K(X, X) + \omega^2 I]^{-1} \vec{y}$  are the weights associated with each kernel, and covariance for set of points  $S = \{x'_1, \dots, x'_k\}$  is defined as

$$\begin{aligned} \Sigma(S) &= K(S, S) \\ &\quad - K^T(X, S)[K(X, X) + \omega^2 I]^{-1} K(X, S). \end{aligned} \quad (3)$$

Due to the addition of the positive definite matrix  $\omega^2 I$ , the matrix inversion in (2) and (3) is well defined.

In order to quantify the size of the domain  $U$  in relation to the GP model, we use the covering number and equivalent distance map, as defined below. These definitions are used in the theoretical results of Section IV.

**Definition 1** The **Covering Number**  $N_c(U, r)$  of a compact domain  $U \subset \mathbb{R}^d$  is the cardinality of the minimal set  $C = \{c_1, \dots, c_{N_c}\}$  s.t.  $\forall x \in U, \exists c_j \in C$  s.t.  $d(x, c_j) \leq r$ , where  $d(\cdot, \cdot)$  is some distance metric.

**Definition 2** The **equivalent distance**  $r(\epsilon_{tol})$  is the maximal distance s.t.  $\forall x, c \in U$ , if  $d(x, c) \leq r(\epsilon_{tol})$ , then the linear independence test  $\gamma(x, c) = k(x, x) - k(x, c)^2 / k(c, c) \leq \epsilon_{tol}$ .

It is important to note that while the GP model assumes the output is normally distributed around the mean, GPs are still consistent estimators of the function mean  $f_i(x)$  even when the output distribution is not Gaussian (see Section IV-B). This work is concerned primarily with accurate predictions of the mean function and not accurate characterization of the output distribution, so the GP model capability suffices. To ease exposition, we assume here that  $f_i(x) \in \mathbb{R}$ , multidimensional GP (Vector-GP) extensions are nontrivial, and considered in [25]

### D. Online Budgeted Inference

In general, traditional GP inference scales poorly with increasing number of data points. The main reason for this is because a new covariance kernel is added for every data point considered, that is, every time a new observation  $\langle x_t, y_t \rangle$  is obtained. Therefore, the matrix inversion typically used for prediction scales as  $O(t^3)$ . Hence, in order to enable efficient GP inference online on resource constrained platforms – for example robotic platforms with limited onboard computational power, such as Unmanned Aerial Vehicles – sparsification of kernels is required to ensure online tractability.

Csato and Opper's [13] budgeted online GP inference algorithm provides a recursive, rank 1 update, for the weights  $\alpha$  and covariance (3). The rank-1 updates restricts the prediction computation growth to  $O(t)$ . In addition, a budget is enforced on the number of kernel functions that are used by the algorithm, the data points retained online are referred to as the *active basis vector set*. The algorithm in [13] only adds a new point to the basis vector set,  $\mathcal{BV}$ , if the data point is sufficiently different from existing data points in  $\mathcal{BV}$ . To determine if a new point is novel, a linear independence test in the underlying RKHS is used. For the Gaussian kernel, the novelty of a data point can be captured in the variable  $\gamma$  using the closed form expression:  $\gamma_t = k(x_t, x_t) - K(X, x_t)^T K(X, X)^{-1} K(X, x_t)$ .

When  $\gamma_t$  exceeds some threshold  $\epsilon_{tol}$ , the data point is added to  $\mathcal{BV}$ . Otherwise, the weights  $\alpha$  and covariance  $\Sigma$  are updated using a rank 1 update, but the updates do not increase the dimension of  $\mathcal{BV}$ . If the budget is exceeded, then data points with highest  $\gamma_i$  are retained. The main benefit of this online sparse GP algorithm is that it allocates new basis vector locations dynamically, thus preventing ad-hoc a priori feature selection. This sparse approximation scales as  $O(|\mathcal{BV}|)$  for calculating the GP mean, and scales as  $O(|\mathcal{BV}|^2)$  for variance, resulting in significant computational savings. Algorithms are available for optimizing GP hyperparameters online as well [26], [10].

### E. Hypothesis Testing

In this paper, a hypothesis refers to the proposal that a set of points is generated from a given model. For two models,  $H_1$  and  $H_0$  with known priors, the decision rule that maximizes

the probability of correct model identification is a likelihood ratio test (LRT):

$$\frac{p(y|H_1)}{p(y|H_0)} \underset{\hat{H}=H_0}{\overset{\hat{H}=H_1}{\geq}} \exp(\eta) \quad (4)$$

where  $\eta = \log((1-p_1)/p_1)$ , and  $p_1 = p(H_1)$  is the prior probability of  $H_1$  occurring. If the left hand side is greater than  $\eta$ , then  $\hat{H} = H_1$  is chosen, otherwise  $\hat{H} = H_0$ . When a prior is not available, the problem is formulated as a *non-Bayesian* hypothesis test based on the probability of detecting an event  $H_1$  and the probability of a false alarm. The Neyman-Pearson lemma states that the decision rule that maximizes the probability of detection subject to some maximum probability of a false alarm is still a LRT. Other statistics exist for determining the distance between distributions, such as the Maximum Mean Discrepancy [27], Hilbert-Schmidt Independence Criterion [28], and Rényi divergence [29], however, the LRT is used as it is intuitively simple and has nice theoretical properties.

We utilize a variation of the Generalized Likelihood Ratio (GLR) [12], a non-Bayesian likelihood ratio test, to perform changepoint detection. The GLR tests for changepoints by comparing a windowed subset of observations  $\vec{y}_S$  and observation input locations  $S$  to a null hypothesis. After each new observation, a new model is created from the same class of models as the null hypothesis,  $\mathcal{H}$ , but with the maximum likelihood statistics corresponding to the data  $y_S$ ,  $H_1(S) = \arg \max_{H \in \mathcal{H}} p(y_S | H)$ . If the ratio of the likelihood of the windowed data with respect to the proposed model to that of the null hypothesis exceeds some ratio, a changepoint is detected. The joint log likelihood of a subset of points  $\vec{y}_S$  given a GP model is

$$\log p(\vec{y}_S | S, \Theta) = -\frac{1}{2}(\vec{y}_S - \hat{\mu}(S))^T (\Sigma(S) + \omega^2 I)^{-1} (\vec{y}_S - \hat{\mu}(S)) - \log |\Sigma(S)|^{1/2} + C \quad (5)$$

In the log-likelihood, the first term accounts for the deviation of points from the mean while the second term weighs the relative confidence (variance) in the prediction values.

### III. THE GP-NBC ALGORITHM

The full algorithm, Gaussian Process Non-Bayesian Clustering, GP-NBC, including a component for using previously identified models (or *clusters*), is presented in Algorithm 1. The main idea of our approach is to decouple the problems of prediction, CPD, and model re-identification. GP-NBC may also be initialized with a set of prior models.

The algorithm begins with a (newly initialized) working model  $GP_w$ . For the prediction problem, GP-NBC is presented with  $x_t$ . Utilizing this data,  $GP_w$  makes a prediction  $\hat{\mu}_w(x_t)$ , and is then updated with observation  $y_t$ . The subscript  $w$  refers not to a specific model number but rather the current model being used for predictions. After each observation, the GP model  $GP_w$  is saved and enqueued to a queue of finite depth  $m$ , the reason of which is described later.

Next, for the changepoint detection problem, a LRT (Algorithm 2) is used, so we need to construct a set of observations

$S$  that the LRT can compare to the current model (line 6). While a sliding window may suffice when successive  $f_i(x)$ ,  $f_j(x)$  differ over all  $x$ , if there is sufficient overlap between functions  $f_i(x)$ ,  $f_j(x)$  one of the following more nuanced approaches is needed:

- The *filtered sliding window* rule uses a buffer of size  $m_S$  for  $S$  and adds points when their log-likelihood is below a given threshold  $\theta$ .
- The *local sliding window (with delay)* rule adds observations to local version of  $S$ , denoted  $S_d(x)$ , that contains the most recent  $m_S$  measurements within a given distance  $d$  of  $x_t$ . To implement a delay, points are added to the set  $S$  after a delay of  $m$  timesteps.

The local sliding window  $S_d(x)$  is constructed by adding  $x_t$  to  $S_d(x)$  at each timestep. If there are more than  $m_S$  points in  $S_d(x)$  within distance  $d$  of  $x_t$ , then the point with the oldest timestamp in that region is discarded. The number of points  $m_S$  depends on the domain, but generally small  $m_S$  on the order of  $m_S \approx 10$  suffices. In practice, points may be added to the set  $S$  as soon they are observed, but in order to derive theoretical guarantees in Section IV, a delay of  $m$  must be used. This delay ensures that the LRT values are uncorrelated, which is required for using statistical tools such as Hoeffding's Inequality. In practice, this equates to adding each observation  $\langle x_i, y_i \rangle$  to  $S$  after  $m$  timesteps, at time  $i+m$ . It is worth emphasizing that  $m_S$  and  $m$  are not the same quantity although both perform similar utilities.  $m_S$  controls the number of points used in the GP model, which controls the variance and speed at which  $\hat{\mu}(x)$  changes over time;  $m$  averages over the LRT values to determine if a changepoint has occurred

At every timestep  $t$ , the value of the LRT  $L_t(y)$  is saved along with corresponding location  $x_t$ , when considering the last  $m$  LRT tests for Line 7 in Algorithm 2. We show in Sec. IV-C that using the latter approach with appropriate parameter settings results in a bound on the number of mistakes made by GP-NBC, even in the case of worst-case (adversarially chosen) inputs. However, we use the filtered sliding window in our experiments as it is simpler to implement and often approximates the local sliding window well, especially for real-world data with some (though not strictly i.i.d.) regularities in the inputs.

After obtaining this set, a new candidate GP model,  $GP_S$ , is created from these points,  $S$ , using the same basis vector set as  $GP_w$  (line 3), and the LRT is calculated for the *current point*  $\langle x_t, y_t \rangle$  against this model and the current GP. If using a delay,  $GP_w^{-m}$  is used for the LRT. If the mean over the last  $m$  LRT values ( $L_m$ ) increases substantially from the mean LRT values calculated so far, then a changepoint is declared and a new model is initialized. If a local sliding window is used, then when GP-NBC queries  $S_d(x_t)$  for the  $m_S$  most recent  $x_i$  close to  $x_t$ , the  $m_S$  LRT values associated with those points are used. Likewise, when a point  $x_t$  is added to  $S_d(x)$ , the LRT value  $L_i$  is saved with it.

If a changepoint is detected, the last  $m$  observations are deleted from  $GP_w$  before saving the model  $GP_w$  to memory. In practice, this is done by saving the  $GP_w$  model after each observation, and maintaining a queue of maximum length  $m$ . If a delay is used, then the last  $2m$  points are removed from

---

**Algorithm 1** GP-NBC
 

---

- 1: **Input:** (Optional) Set of models  $\{GP_1, \dots, GP_N\}$ , max queue length
- 2: Initialize new working GP,  $GP_w$ , with  $\hat{\mu}_w(x) = 0, \forall x$
- 3: **while** Input/Output  $\langle x_t, y_t \rangle$  available **do**
- 4:   Predict  $\hat{\mu}_w(x_t)$  using  $GP_w$
- 5:   Update  $GP_w$  with  $\langle x_t, y_t \rangle$  [13]
- 6:   Add  $\langle x_t, y_t \rangle$  to  $S$  using rules in Section III
- 7:   Add  $GP_w^t = GP_w$  to queue,  $Q$ ,  $\{GP_w^{t-1}, GP_w^{t-2}, GP_w^{t-3}, \dots\}$
- 8:   **if** Length(Queue) > max length **then**
- 9:     Dequeue last element
- 10:   **if**  $GP_w$  has >  $R$  measurements **then**
- 11:     Call Algorithm 2 with  $\langle S, \langle x_t, y_t \rangle, \text{Queue } Q \rangle$
- 12:     Call Algorithm 3 with  $GP_w$ , basis vectors  $BV_w$ , and set of models  $\{GP_1, \dots, GP_N\}$

---



---

**Algorithm 2** Changepoint Detection
 

---

- 1: **Input:** Set  $S$ , Observation  $\langle x, y \rangle$ , Queue  $Q$
- 2:  $l_1 = \log p(y | x, GP_w)$  or if using delay:  $l_1 = \log p(y | x, GP_w^{-m})$
- 3: Create new GP  $GP_S$  from  $S$
- 4:  $l_2 = \log p(y | x, GP_S)$
- 5: Store LRT  $L_i = (l_2 - l_1)$
- 6: **if** LRT was calculated for last  $m$  observations (i.e.  $i > m$ ) **then**
- 7:   Calculate average of last  $m$  LRT:  

$$L_m = \frac{1}{m} \sum_{j=i-m+1}^i L_j$$
- 8:   Calculate average of LRT after last changepoint:  

$$L_{ss} = \frac{1}{i-m-1} \sum_{j=1}^{i-m-1} L_j$$
- 9:   **if**  $L_m - L_{ss} \geq \eta$  **then**
- 10:     Restore  $GP_w = GP_w^{t-m}$  or to  $GP_w = GP_w^{t-2m}$  if using a delay, add to set of  $n$  models
- 11:     Initialize new model  $GP_{n+1}$ , set  $GP_w = GP_{n+1}$ , set  $i = 1$
- 12: **else**
- 13:    $i = i + 1$

---

$GP_w$ , and a queue of maximum length  $2m$  is used (line 8 of Algorithm 1). After a changepoint is detected, an element is dequeued restoring the current model  $GP_w^t$  to the  $GP_w$  model  $m$  or  $2m$  observations ago:  $GP_w^{t-m}$  or  $GP_w^{t-2m}$ . This prevents points that were potentially generated after a changepoint from being included in the old model. In applications where changepoints are relatively infrequent compared to  $m$ , the associated data loss of deleting  $m$  points is quite small. After detecting a changepoint, GP-NBC uses a burn-in period of approximately  $R \in [2m, 4m]$  before calculating the LRT again (line 10).

The CPD portion of GP-NBC tends to perform robustly in practical applications based on the following intuition. In the first step, the filtered sliding window selects only anomalous points while ignoring data that were likely generated by the current model. In the second step, a new candidate GP model,  $GP_S$ , is created from this data,  $S$ . If the data are anomalous simply due to noise in the output, then, on average, the new

---

**Algorithm 3** Compare to Previous Models
 

---

- 1: **Input:** Model  $GP_w$ , Basis vectors  $BV_w$ , and set of models  $\{GP_1, \dots, GP_N\}$
- 2:  $\hat{\mu}_w(BV_w) = GP_w \cdot \text{predict}(BV_w)$
- 3: **for** Each model  $j$  in set of models **do**
- 4:    $l_1 = \log P(\hat{\mu}_w(BV_w) | GP_j)$
- 5:    $l_2 = \log P(\hat{\mu}_w(BV_w) | GP_w)$
- 6:    $\gamma_j = \frac{1}{|BV_w|} (l_2 - l_1)$
- 7:   **if**  $\gamma_j \leq \eta$  **then**
- 8:     Store  $\langle j, \gamma_j \rangle$  in an array  $\Gamma$
- 9: Set  $i$  to be the index of the smallest value in the array  $\Gamma$
- 10: Delete current model and set  $GP_w = GP_i$

---

model created from these points will be similar to the current model, and the likelihood of the points given either model will be similar (in terms of the likelihood ratio test).

If the data are anomalous because they were drawn from a new process, then on average, the GP model created from these points will be substantially different from the current model. Then, the likelihood of these points will be much higher for the new model relative to the current model. Lastly, instead of deciding whether or not a changepoint has occurred based on a single LRT, GP-NBC uses the mean of the last  $m$  LRT's and compares this to the mean LRT values seen since the last changepoint. In particular, in Section IV, theoretical guidelines for setting  $m$  are given. In practice, the algorithm is robust to the selection of parameters  $m$  and  $\eta$ , as shown in the empirical section.

The GP-NBC algorithm is modified to accommodate the storage and reuse of previous models as such: after at least  $R$  data points are added to a model, GP-NBC uses a LRT to see if  $GP_w$  is significantly close to a previous GP model (Algorithm 3). For this test, the points  $\hat{\mu}_w(BV_w)$  are used as an artificial data set. If the values  $\hat{\mu}_w(BV_w)$  are not substantially different from an old model  $GP_i$ , i.e. the LRT is below some threshold, then the new model is deleted and the previously learned model with the lowest LRT score is used instead. The basis vectors are chosen in such a manner to adequately cover the input domain, which prevents over-fitting to one region of the input domain.

In summary, the computational complexity is  $O(|BV|^2m)$  for the LRT and  $O(|BV|^2N)$  for comparing a current model to all previous models, due to variance calculations. The algorithm designer may either limit  $|BV|$  for applications requiring fast computation, or simply allow the sparse GP algorithm to automatically add points to  $BV$  until the domain  $U$  is adequately covered, i.e.  $\forall x \in U, k(x, x) - k(x, BV)K(BV, BV)^{-1}k(x, BV)^T < \epsilon_{tol}$ . For most applications, the sparse GP algorithm will stop adding points to  $BV$  after approximating 10-200 points, depending on the size of the domain and accuracy required.

#### IV. THEORETICAL ANALYSIS

In this section, we prove bounds on the number of mistakes (predictions with large error) that GP-NBC will make with high probability. Since the process generating  $y_t$  is stochastic,

the theoretical statements all have a probability of failure  $\delta$  that contributes (polynomially) to the number of mistakes the algorithm might make. This is a necessity when proving the accuracy of an algorithm such as GP-NBC that is trained on stochastic values and is in line with traditional sample complexity analysis frameworks such as PAC [30]. It should, however, be noted that, the sample complexity bounds presented here are loose and cover the worst case analysis. They are provided to show the *scalability* of our solution, with increased problem size and are likely too conservative for choosing real-world application parameters.

We first motivate the choice to use hypothesis testing for nonstationary model detection. In Section IV-B, we determine the maximum number of mistakes a GP can make when learning a single mean function,  $f(x)$ , by determining a sufficient condition on the variance (Lemma 1) for accurate prediction and bounding the number of mistakes before the predictive variance meets this condition (Theorem 1). In Section IV-C, we analyze the nonstationary case by showing that if certain assumptions are met on the class of functions, we can lower bound the expected value of the LRT (Equation 12); and given a sufficiently large, but polynomially bounded (in relevant quantities, including the covering number of the input space), number of samples  $m$  from the new function, (Lemma 3), we show that GP-NBC will either detect a changepoint or return accurate predictions (Theorem 2).

#### A. Motivation

This section serves to motivate the use of the LRT for CPD using results from the theory of Large Deviations and to outline the proof structure of Section IV-C. Given two models  $H_0$  and  $H_1$ , the theory of Large Deviations states given  $m$  i.i.d. observations from a single model  $H_1$ , the average of the LRT values will tend towards the KL-Divergence  $D(H_1 | H_0)$  exponentially fast. When  $H_1$  is not known a priori, but there is an approximate model  $\hat{H}_1$  built from existing data, the LRT will tend towards

$$\mathbb{E}[L_{H_1 | H_0}(y)] = D(H_1 | H_0) - D(H_1 | \hat{H}_1) \quad (6)$$

which is the original KL-divergence minus the approximation error of using a model  $\hat{H}_1$  instead of  $H_1$ . See Appendix A for a proof.

Since the LRT values tend to (6) exponentially fast, it follows that even for small numbers of  $m$ , the LRT will perform well. Furthermore, since (6) can be calculated explicitly for many real distributions, values of  $\eta$  can be set explicitly beforehand and given intuitive interpretation in terms of distributional differences and modeling error. While  $H_1$  and  $H_0$  are unknown beforehand in this application, bounds on its value can still be determined. In particular, Lemma 2 of Section IV-C determines a lower bound on the the KL divergence between two consecutive generating distributions  $D(H_1 | H_0)$ , and assuming an upper bound on the approximation error  $D(H_1 | \hat{H}_1)$ , a lower bound on (6) can be found. Lemma 3 then uses Hoeffding's Inequality to derive a bound on the number of LRT values,  $m$ , such that with high probability the average of  $m$  LRT values will be within some tolerance of the expected value. At this point, a changepoint will be detected.

In order to apply the theory of large deviations to our application in which  $x_i$  may not be i.i.d., the expectation operator is conditioned on  $x$  such that if  $H_1$  is implicitly a function of  $x$ , (6) holds for each  $x$ . Furthermore, the assumptions of boundedness and function smoothness from Section II-A are used in the proofs. For any arbitrary distribution satisfying these assumptions, the theorems in the stationary analysis section, section IV-B hold, but for nonstationary analysis, further requirements are used. The analysis assumes that  $x_t$  are chosen adversarially in order to derive a maximum bound on the number of mistakes.

#### B. Stationary Analysis

We begin our analysis by bounding the number of mistakes, as defined in Section II-A, within a given phase without a changepoint (that is when the generating distribution is stationary). This value ultimately determines the rate at which a new GP model can be learned, and therefore how frequent changepoints can occur while still learning new separate models. In the following lemma, a relationship between the variance of a GP and the probability of making a mistake is derived.

**Lemma 1** Consider a GP trained on samples  $\vec{y} = [y_1, \dots, y_t]$  which are drawn from  $p(y | x)$  at input locations  $X = [x_1, \dots, x_t]$ , with  $\mathbb{E}[y | x] = f(x)$ ; if the predictive variance of the GP at  $x' \in X$  is

$$\sigma^2(x') \leq \sigma_{tol}^2 = \frac{2\omega^2 \epsilon_E^2}{V_m^2 \log(\frac{2}{\delta_1})} \quad (7)$$

then a mistake at  $x'$  is bounded in probability:  $Pr\{|\hat{\mu}(x') - f(x')| \geq \epsilon_E\} \leq \delta_1$ .

*Proof sketch:* The proof appears in Appendix B and applies McDiarmid's inequality to the general GP equations to bounds the changes in predicted values for a given level of predictive variance. ■

The inequality condition in lemma 1 is sufficient to ensure, with high probability, that the mean of the GP is within  $\epsilon_E$  of a stationary function  $f(x)$ . Conversely, for variances greater than  $\sigma_{tol}^2$ , no guarantees can be made that the GP will not make a mistake. Next, Theorem 1 determines a maximum number of samples a GP may make erroneous predictions on before the condition on variance from Lemma 1 holds true everywhere in  $U$ , and the GP returns accurate predictions, with high probability. In order to account for error of sparsification, Theorem 1 uses a finer tolerance of  $\frac{1}{4}\sigma_{tol}^2$  (corresponding to  $\epsilon_E = \frac{\epsilon_E}{2}$  from (7)). The following theorem assumes adversarial inputs.

**Theorem 1** Consider a sparse GP model with linear independence test threshold  $\epsilon_{tol} = \frac{\epsilon_E}{2V_m}$ , trained in the online setting (with  $\langle x_t, y_t \rangle, t = 1..$  as described earlier) over a compact domain  $U \subset \mathbb{R}^d$ . Let the set of observations  $\vec{y}$  over all timepoints  $t$  be drawn from  $p(y_t | x_t)$ , with  $\mathbb{E}[y_t | x_t] = f(x_t)$ , and  $x_t$  drawn adversarially. Then, with probability  $1 - \delta_1$ , the number of mistakes where:

$$|\hat{\mu}(x_t) - f(x_t)| \geq \epsilon_E \quad (8)$$

is at most

$$n = \left( \frac{4V_m^2}{\epsilon_E^2} \log \left( \frac{2}{\delta_1} \right) \right) N_c \left( U, r \left( \frac{\omega^2 \epsilon_E^2}{4V_m^2 \log(\frac{2}{\delta_1})} \right) \right). \quad (9)$$

Furthermore, the covering number grows polynomially with  $\frac{1}{\epsilon_E}$ ,  $V_m$ ,  $\frac{1}{\delta_1}$ , but exponentially in the input dimension  $d$  for the Gaussian Radial Basis Function (RBF) kernel.

*Proof sketch:* The proof breaks up the domain into Voronoi regions around the covering set and determines an upper bound on the number of points required reduce the variance everywhere in the region below  $\frac{1}{4}\sigma_{tol}^2$ . In order to show  $N_c \left( U, r \left( \frac{\omega^2 \epsilon_E^2}{4V_m^2 \log(\frac{2}{\delta_1})} \right) \right)$  grows polynomially in  $\frac{1}{\epsilon_E}$ ,  $V_m$ , and  $\frac{1}{\delta_1}$ , the proof bounds the number of Voronoi regions required to cover the input domain by using volumetric arguments. This is done by dividing the volume of the region into hypercubes of volume less than or equal to that of the Voronoi regions, and showing that the number of hypercubes grows polynomially in  $\frac{1}{\epsilon_E}$ ,  $V_m$ , and  $\frac{1}{\delta_1}$ , although the number grows exponentially in  $d$ . See the proof in Appendix C for further details. ■

In theory, the covering number grows exponentially with the dimensionality of the space, which limits the practical capabilities of using GPs without sparsification, but in practice sparse GPs have been shown to work well in relatively high dimensional spaces [31], [13]. Note that in the adversarial setting, mistakes may occur at any time, but, Theorem 1 states that the cumulative sum of those mistakes may not exceed some value  $n$ .

### C. Nonstationary Analysis

Building on the results for the stationary case, we now consider the nonstationary case in which  $f_i(x)$  changes between phases. These changes in  $f_i(x)$  must be inferred online, using only sequential observations. Additional assumptions are made here about the class of functions  $\mathcal{F}$  that may be detected using the nonstationary algorithm: *realizability*, *separability*, and *Gaussianity*. It is assumed that each function  $f_i \in \mathcal{F}$  can be exactly modeled by a GP given infinite data and the expected approximation loss of using a GP model with finite data  $S$  is approximately the same over all functions, functions are well separated regionally, and the distributions  $p_i(y|x)$  conditioned on  $x$  are Gaussian, although possibly heteroskedastic. The first and third assumptions are required to enforce some regularity among the generating distributions, so that bounds on the KL-divergence and LRT can be derived, while the second assumption enforces that successive generating distributions have different enough means to allow changepoint detection.

**Assumption 1 (Realizability)** The approximation error between a GP model learned from a subset  $S$  of the data obtained by the local sliding rule and the true generative distribution does not change between phases by more than  $\epsilon_{DS}$ , where

$$\sup_{i,j} |D(p_i||GP_S) - D(p_j||GP_S)| \leq \epsilon_{DS} \quad (10)$$

**Assumption 2 (Separability)** All functions  $f \in \mathcal{F}$  differ by some  $\epsilon_F$  over at least one compact input region  $\tilde{U}$ :

$$\forall i, j : \exists \tilde{U} \subset U \text{ s.t. } |f_i(x) - f_j(x)| \geq \epsilon_F > \epsilon_E \quad (11)$$

**Assumption 3 (Gaussianity)** For all  $i$ ,

$$p_i(y_i|x) \sim \mathcal{N}(f_i(x), \omega_i^2(x))$$

with nonzero variance  $\inf_x \omega_i^2(x) \geq \omega_{min}^2 > 0$  and Lipschitz constant  $K$  associated with  $f_i(x)$ .

Assumption 3 is a regularity condition since the KL-divergence may be ill-defined in the case of a deterministic function.

The next lemma relates the absolute distance between distribution means to a lower bound in the KL-divergence between a  $GP_w$  and the generating distribution. This is the same as finding a lower bound on  $D(H_1|H_0)$  from (6).

**Lemma 2** Consider a GP,  $GP_w$ , trained on  $p_i(y|x)$  with  $\sigma_x^2(x) \leq \frac{1}{4}\sigma_{tol}^2$ ,  $\forall x \in \tilde{U}$ , and a set of distributions  $p_j(y|x) \in \mathcal{P}$  satisfying assumptions 1,2,3; then  $\forall x \in \tilde{U}$ , then we can ensure for a given probability  $1 - \delta_d$

$$D(p_j(y|x)||GP_w) - D(p_i(y|x)||GP_w) \geq \eta \quad (12)$$

where  $\eta$  equals

$$\frac{1}{2} \left( \frac{\omega^2 + \sigma_{tol}^2 - \omega_{min}^2}{\omega^2 + \sigma_{tol}^2} - \frac{\epsilon_E^2}{\omega^2} + \log \left( \frac{\omega_{min}^2}{\omega^2} \right) + \frac{(\epsilon_F - \epsilon_E)^2}{\sigma_{tol}^2 + \omega^2} \right) \quad (13)$$

*Proof:* The KL-divergence between two normal variables [32] is given by

$$D(p_0||p_1) = \frac{1}{2} \left( \frac{\sigma_0^2}{\sigma_1^2} - \log \frac{\sigma_0^2}{\sigma_1^2} - 1 + \frac{(\mu_0 - \mu_1)^2}{\sigma_1^2} \right) \quad (14)$$

The minimum that the first two terms can equal 1 is when  $\sigma_1 = \sigma_0$ ; the maximum is when  $\sigma_1^2 = \omega^2 + \sigma_{tol}^2$  and  $\sigma_0 = \omega_{min}^2$ . By minimizing the variance related terms in  $D(p_2(y|x)||GP_w)$ , maximizing the variance related terms in  $D(p_1(y|x)||GP_w)$ , and bounding  $|\hat{\mu}_w(x) - f_1(x)| \leq \epsilon_E$ , w. p.  $1 - \delta_d$ , (substituting  $\delta_d$  for  $\delta_1$  in Lemma 1) where  $f_1$  is the mean of  $p_1$ , (13) is obtained. ■

The expected value of the LRT, from (6), is the KL-divergence between the current distribution and the working model, shifted by the model error, which is bounded from assumption 1. In order to detect a changepoint, one can simply determine the point in time where the mean of the time series  $L(y)$  changes significantly. This equates to finding the number of observations  $m$  such that the average of  $m$  LRT values will be within some tolerance of the expected LRT value from Lemma 2, and therefore above the LRT threshold for declaring a changepoint. Lemma 3 finds  $m$  allowing for some probability of false alarm or failed CPD,  $\delta_L$ , conditioned on Lemma 2 holding true.

**Lemma 3** Consider a  $GP_w^{t-m}$  trained on  $n_1 > \frac{8V_m^4}{\omega^4 \eta^2} \log \left( \frac{4}{\delta_L} \right)$  samples from  $p_1(y|x)$ ,  $x \in \tilde{U}$ . Consider a second GP trained from a set of  $m \ll n_1$  samples, which are drawn from  $p_2(y|x)$  from region  $\tilde{U}$ , with property,  $D(p_2(y|x)||GP_w) - D(p_1(y|x)||GP_w) \geq \eta + \epsilon_{DS}$ ,  $\forall x \in \tilde{U}$ .

Then, if the window size,  $m$ , and  $n_1$  satisfy the inequalities stated below, then w.p.  $1 - \delta_d - \delta_L$ ,  $L_m(y) - L_{ss}(y) \geq \eta$ ,

$$m \geq \frac{\frac{8V_m^4}{\omega^4} \log\left(\frac{4}{\delta_L}\right) n_1}{\eta^2 n_1 - \frac{8V_m^4}{\omega^4} \log\left(\frac{4}{\delta_L}\right)} \quad (15)$$

For large  $n_1$

$$m \geq \frac{8V_m^4}{\omega^4 \eta^2} \log\left(\frac{4}{\delta_L}\right). \quad (16)$$

Furthermore, if these window size conditions are met, Algorithm 2 will detect a changepoint w.p.  $1 - \delta_d - \delta_L$ .

*Proof:* Since a delay  $m$  is used, the observations used in the LRT are not included in the models  $GP_w$  or  $GP_S$ , making the LRT values independent of each other. Furthermore, since the output domain is bounded, the LRT values  $L(y)$  are bounded. These conditions are sufficient to use Hoeffding's inequality to bound the variation of the average of the LRT values from their expected value, the KL-divergence. To proceed, the maximal range of the LRT is bounded, and is given by  $L(y_i) = \frac{1}{2} \left( \log \frac{\sigma_w^2(x_i) + \omega^2}{\sigma_s^2(x_i) + \omega^2} + \left( \frac{(y_i - \hat{\mu}_w)^2}{\sigma_w^2(x_i) + \omega^2} - \frac{(y_i - \hat{\mu}_s)^2}{\sigma_s^2(x_i) + \omega^2} \right) \right)$ , where the subscript  $w$  and  $s$  refer to the working GP model and GP model based on subset  $S$ . The variance is not affected by  $y_i$  so the first term is a constant. The last two terms are bounded by the  $V_m^2/\omega^2$  each, and so the maximal range of each  $L(y_i)$  is  $c_i = \frac{2V_m^2}{\omega^2}$ . Therefore, using Hoeffding's Inequality, one can bound the distance of the averages  $L_m$ ,  $L_{ss}$  from their respective expected values,  $|L_{(\cdot)}(y) - (D(p_i || GP_w) - D(p_i || GP_S))| \leq \epsilon_{(\cdot)}$  w. p.  $1 - \delta_L/2$ , for  $(\cdot) \in \{m, ss\}$ .

Given  $n_1$  and  $\delta_L$ , the accuracy of the current GP,  $\epsilon_w$ , can be determined and the required number of samples  $m$  to drive  $\epsilon_S \leq \eta - \epsilon_w$  can be solved for through algebraic manipulation. The total probability of (15) not holding true is obtained by the union bound of probability of failure of (12) not holding true,  $\delta_d$ , and the probability,  $\delta_L$ , of  $L_m(y) - L_{ss}(y) \geq \eta$  not holding true. This yields a final probability of  $1 - \delta_d - \delta_L$ . ■

Lemma 3 states that if the generating function switches such that the working model  $GP_w^{t-m}$  now predicts erroneously in some region  $\tilde{U}$ , then a set of observations built from the new distribution of size  $m$  will detect the change with high probability. A delay of  $m$  is used, so that the LRT values  $L_t$  are independent of each other. If the observations at times  $t - m$  to  $t$  were used in the models  $GP_w$  and  $GP_S$ , then the LRT values would be correlated and Hoeffding's Inequality could not be applied. However, in practice, these correlations tend to be quite small, and so a delay is not required. Lemma 3 considers only predictions in regions, and so now Theorem uses Lemma 3 to bound the number of erroneous predictions across a domain before GP-NBC either 1) detects a changepoint or 2) predicts the new function accurately in all regions, as is the case when the two mean functions are similar in certain regions or if the GP had no data points in a region (and so the GP mean can change).

For the next theorem, consider a KL-divergence change threshold  $\eta$  associated through (13) with an  $\epsilon_F$  strictly smaller than  $\epsilon_E$ ,  $\epsilon_F < \epsilon_E$ . Define  $U_E = \{x \in U \text{ s.t. } |f_i(x) - f_j(x)| > \epsilon_E\}$ , and  $U_F = \{x \in U \text{ s.t. } |f_i(x) - f_j(x)| > \epsilon_F\}$  as the

regions where the mean functions differ by at least  $\epsilon_E$  and  $\epsilon_F$ , respectively. The next theorem bounds the total number of mistakes that GP-NBC can make per phase. It accomplishes this by adding the number of mistakes a GP can make due to high predictive variance,  $\frac{1}{4}\sigma_{tol}^2$ , and adding the maximum number of mistakes that the GP can make before detecting a changepoint, with high probability. It uses the covering number, Lipschitz constant  $K$ , probability of (12) not holding,  $\delta_d$ , (Lemma 2), and probability of  $L_m(y) - L_{ss}(y) < \eta$  not holding,  $\delta_L$  (Lemma 3).

**Theorem 2** Consider  $GP_w$  built on samples from the generating distribution  $p_i(y | x)$ . Assume that at some time instant  $t$ , there is a changepoint and the generating distribution switches to  $p_j(y | x)$  satisfying assumptions 1,2,3. Consider a local sliding window  $S_d(x)$  with  $d \leq \frac{1}{K}(\epsilon_E - \epsilon_F)$  and  $L_m(s)$  considering the last  $m$  LRTs performed within distance  $d$  of  $x_i$ . After  $m$  LRT calculations within the region of  $x$ , Algorithm 2 will detect a changepoint in that region w. p.  $1 - \delta_d - \delta_L$  or, after a sufficient number of mistakes (see below) predict  $f_j(x)$  accurately in this region until the next changepoint. The total number of mistakes by Algorithm 2 per phase is bounded by

$$n + (2m - 1)N_c(U_E, r(d)), \quad (17)$$

w. p.  $1 - \delta_d - \delta_L$ .

*Proof:* Let  $\sigma_t^2(x)$  denote the predictive variance of the GP at a point  $x$  in the input domain at time  $t$ . Note, the predictive variance in a region decreases as points are added to a GP. There are three types of regions, or conditions, under which a mistake may be made after a changepoint, 1) the GP mean differs from  $f_j(x)$  by at least  $\epsilon_E$ , and the variance of the GP is low,  $\sigma_t^2(x) \leq \frac{1}{4}\sigma_{tol}^2$ , 2) the GP mean and  $f_j(x)$  differ by at least  $\epsilon_E$  and the variance is high  $\sigma_t^2(x) > \frac{1}{4}\sigma_{tol}^2$ , and 3) the mean functions may not differ by more than  $\epsilon_E$ , but the variance is high  $\sigma_t^2(x) > \frac{1}{4}\sigma_{tol}^2$ , and so we cannot say with certainty that no mistake has been made. We have from Lemma 1 combined with Theorem 1 that the number of mistakes that can be made before  $\sigma_t^2(x) \leq \frac{1}{4}\sigma_{tol}^2$  everywhere is  $n$ . That is, Theorem 1 states that the maximum number of mistakes is in the stationary case is  $n$  and its proof links to Lemma 1 to show mistakes only occur when there is high variance. After  $n$  mistakes have been made, the variance everywhere is low, and so no more errors may occur in types of regions 2) and 3).

After the variance has reduced everywhere, the domain can be divided into the region in which the GP makes an error  $U_E$ , and its complement, where the GP prediction is accurate. The adversary may choose  $x_t$  from  $U_E$ , but after the adversary chooses  $m - 1$  observations from each region of the covering set, the next observation will yield  $m$  samples in a sliding window  $S_d(x)$ , by the Pigeon Hole Principle. At this point, the GP variance is below  $\frac{1}{4}\sigma_{tol}^2$ , and the GP prediction and mean function differ by at least  $\epsilon_E$ , satisfying the requirements of Lemma 3. Lemma 3 then guarantees that a changepoint will be detected w.p.  $1 - \delta_d - \delta_L$ . Accounting for  $m$  mistakes per region due to the delay, we have that  $2m - 1$  mistakes may be made per region. Combining these mistakes with the number of mistakes possible due to high variance,  $n$ , we have that

the total number of mistakes per changepoint is  $n + (2m - 1)N_c(U_E, d)$ , w. p.  $1 - \delta_d - \delta_L$ . ■

Note that in the above result, we enforce  $\epsilon_F < \epsilon_E$ , since, without it, points may be sampled arbitrarily close on either side of the boundary of the region  $U_E$ , and the nonstationary shift will not be detected.

## V. EXPERIMENTS

The following empirically compares GP-NBC to several state-of-the-art techniques on simulated and real-world datasets. GP-NBC is used as described earlier, except that the joint likelihood of the last  $m$  points is used when calculating likelihoods in Algorithm 2, rather than the likelihood of single points. This approach, which is more difficult to analyze theoretically, seems to have empirical benefits. The competing algorithms are: GP-CPD [10], the Infinite Mixture of Gaussian Process Experts (DP-GP) [18], and a MCMC - sampling based Chinese Restaurant Process (MCMC-CRP) algorithm [19] which, unlike DP-GP, samples new changepoint configurations, rather than sampling over individual data points. We also compare to a naïve implementation of GP-regression with a forgetting factor, GP-Forget. All experiments are run on a Macintosh Pro with 2.3 GHz Intel Core i7 processor and 8 GB RAM. The results demonstrate that GP-NBC achieves a similar or better classification and regression error compared to the batch algorithms, even though it is online. The ability of GP-NBC to sequentially process data and its demonstrated computational efficiency makes GP-NBC a feasible algorithm for real-time implementation on resource constrained platforms, such as Unmanned Aerial Vehicles with limited computational power. GP-NBC also requires several orders of magnitude less calculation time compared to the batch algorithms and GP-CPD. This section uses two synthetic examples and two real domains to compare performance across the algorithms. In the last domain, GP-NBC augments DP-GP by using the models learned by DP-GP as a set of prior models. We show that GP-NBC's ability to detect and learn new clusters as opposed to only using the prior models, leads to a substantial improvement.

### A. Synthetic Datasets

The first experiment uses two very similar functions in order to test subtle change points:  $f_1(x) = x^2 - \frac{1}{2}$ , and  $f_2(x) = \frac{1}{2}x^2$  over the domain  $x \in [-1, 1]$ . Observations are corrupted with white Gaussian noise,  $y = f_i(x) + \nu$ , where  $\nu \sim \mathcal{N}(0, 0.2^2)$  is Gaussian white measurement noise, and  $x$  are drawn i.i.d. from a uniform distribution. Figure 1 shows these two functions plotted together. In the second experiment, the functions are well separated,  $f_1(x) = x^2$ , and  $f_2(x) = -x^2 + 2$ , with  $x$  drawn i.i.d. Changepoint detection becomes much easier, but this experiment demonstrates a situation in which using a naïve sliding window approach would lead to significant regression errors. For both datasets, the generating function switches from  $f_1$  to  $f_2$  at some time  $\tau$  drawn uniformly from  $(75, 125)$ , and switches back to  $f_1$  at some  $\tau$  drawn uniformly from  $(175, 225)$ . The run ends at  $t = 400$ . GP-NBC parameters were set to window threshold  $\theta = 1$ , detection parameter  $\eta = 0.5$ ,

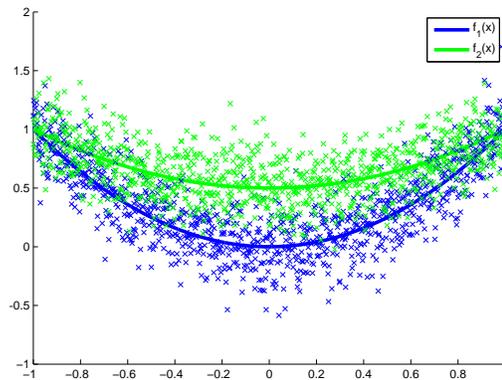


Fig. 1.  $f_1$  (lower curve) and  $f_2$  for the first experiment.

roll-back parameter  $R = 5$ , and  $|S| = 10$ . Parameters for the other algorithms were chosen by grid search optimizing over mean absolute regression error.

Table I compares 75 runs of each algorithm on both experiments. For both experiments, GP-NBC has the best performance in terms of average prediction error and identifying the correct number of clusters, and also achieves this with a runtime one to two orders of magnitude faster than the batch methods. When MCMC-CRP and DP-GP correctly identify the changepoints, they achieve the minimum error, however, they often converge to local optima and misidentify the changepoints, leading to much higher error. Since GP-NBC can reuse models, it outperforms GP-CPD in terms of prediction error, and also takes approximately 30 times less time to run. Lastly, GP-Forget has good prediction performance for the first example. However, the low error is an artifact of the functions being very close. In the second example, aliasing due to the sliding window results in extremely large errors for GP-Forget while GP-NBC performs equally well in both cases. Results are statistically significant: GP-NBC and the closest competitor, GP-CPD, are separated by 4 standard deviations in both experiments in prediction error.

Figure 2 compares GP-NBC on example 1 over a grid of parameters from  $\eta \in [0.1, 1.6]$  and  $m \in [3, 40]$ . For  $\eta \geq 2$ , the changepoint does not trigger a detection, and for  $\eta \leq 0.1$ , false positives occur. For reasonable values of  $\eta \approx 0.5$ ,  $m \in [3, 30]$  results in similar performance. These results demonstrate that GP-NBC is fairly robust to parameter selection.

### B. Robot Interaction

The next experiment highlights the ability of GP-NBC to identify multiple models in a real data set and leverage these for improving prediction. In this experiment, a iRobot Create, named GPUC-1, is driven manually while a second Create, named GPUC-2 responds to GPUC-1's movements in a variety of ways: 1) attempting to encircle GPUC-1 clockwise or 2) counterclockwise, 3) attempting to follow GPUC-1, or 4) tracking a point 1m behind GPUC-1. An overhead VICON motion tracking system measures the position and orientation of the Creates at 5Hz, and the velocities are estimated using a fixed point smoother. Every 200 seconds, GPUC-2 elects a different, random behavior, possibly reusing a previously

TABLE I  
 ERRORS, CLUSTERS AND RUNTIMES ON SYNTHETIC DATA

Experiment 1				Experiment 2			
	Mean Abs. Error	Max Error	No. Clusters	Mean Abs. Error	Max Error	No. Clusters	Runtime (s)
GP-NBC	<b>0.0484±0.0050</b>	<b>0.0624</b>	<b>2.00±0</b>	0.0436±0.0050	<b>0.0662</b>	<b>2.0±0</b>	3.16
GP-CPD	0.0679±0.0074	0.0848	–	0.0591±0.0044	0.0674	–	108.4
DPGP	0.0802±0.0167	0.1109	3.066±0.47	0.0755±0.0156	0.1104	4.38±0.70	217.5
MCMC-CRP	0.0647±0.0469	0.1369	1.88±0.49	<b>0.0403±0.0156</b>	0.0835	3.8±1.61	158.6
GP-Forget	0.0685±0.0055	0.0843	–	0.1556±0.022	0.2032	–	<b>0.71</b>

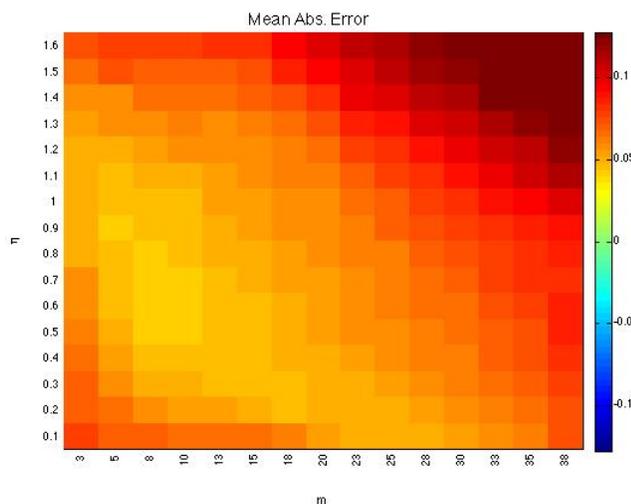


Fig. 2. Heat map of the Mean Abs Error (MAE) for various parameter settings for GP-NBC. Green corresponds to better MAE than the optimal parameter settings of GP-CPD, black to equivalent performance, and red to worse performance.

exhibited behavior as well. The difference in the  $(x, y)$  position between GPUC-1 and GPUC-2, and the heading of GPUC-1 is used as the input to the algorithms, and the velocity of GPUC-2 is the output. Figure 3 shows GP-NBC robustly identifying changepoints in GPUC-2's behavior as well as reclassifying previously seen behaviors. In this experiment,  $R = 20$  yielded a good model reidentification without negative transfer. The GP-NBC algorithm took 2.9min to run and had a mean error of 0.0661; GP-CPD took 6.8h and had mean error 0.0785. GP-NBC outperforms GP-CPD since it is able to reuse previous models. After 6 hours, we stopped MCMC-CRP and DP-GP. GP-NBC outperformed GP-CPD in terms of regression error as well as taking two orders of magnitude less time.

### C. Pedestrian Behavior Classification

In the final experiment, we demonstrate the ability of GP-NBC to utilize prior models while simultaneously detecting anomalous (new) functions. The setting we consider is modeling pedestrian behaviors, i.e. typical trajectory patterns, at an intersection and predicting the position of the pedestrian some time in the future. This experiment is modeled after work by [8], which considers pedestrian intent, or goals. In this setup, the algorithm may be given a set of observed behaviors, or trajectory patterns, but during testing, may observe a new behavior or observe a human changing behavior patterns. This

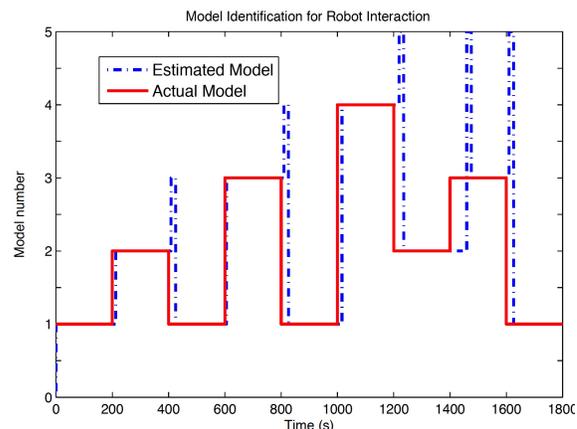


Fig. 3. GP-NBC detects changepoints as well as reclassifying old models of robot interaction dataset.

has many applications to pedestrian avoidance in autonomous driving.

In this experiment, a pedestrian walks along one of four trajectories at an indoor intersection, as seen in Figure 4. The pedestrian position is measured using a LIDAR sensor placed in front of the intersection. For more experimental setup details, please refer to [33] or Chapter 5 and 6 of [34]. The algorithm is trained on three of the four behaviors (all but the teal trajectories) in batch using DP-GP. The training data can be seen in the top right subfigure of Figure 4. In the testing phase, a pedestrian may follow one of the three previously observed trajectories, or may follow a new trajectory, as seen in the lower left subfigure of Figure 4. After testing, GP-NBC correctly identifies a change in behavior and reclassifies the new trajectory pattern. The output of GP-NBC after the testing phase is seen in the lower right subfigure of 4.

The experiment has five trajectories, with approximately 600 samples per trajectory. The input is the pedestrian location and velocity, and the output is the velocity at the next time step. Figure 4 shows that GP-NBC correctly identifies when a pedestrian deviates from a behavior learned in the training phase and reclassifies the anomalous behavior as the fourth model.

GP-NBC is compared with DP-GP using the one-timestep ahead and ten-timestep ahead predictions of the pedestrian's future location. Prediction error is computed as the root mean square (RMS) difference between the true position and mean predicted position, where the 10-step ahead prediction utilizes both the GP mean and variance for propagation [35]. The prediction error, averaged over all trajectories and time, was similar for both algorithms for trajectories within the three training clusters (mean difference of 10.25% and 2.47% for

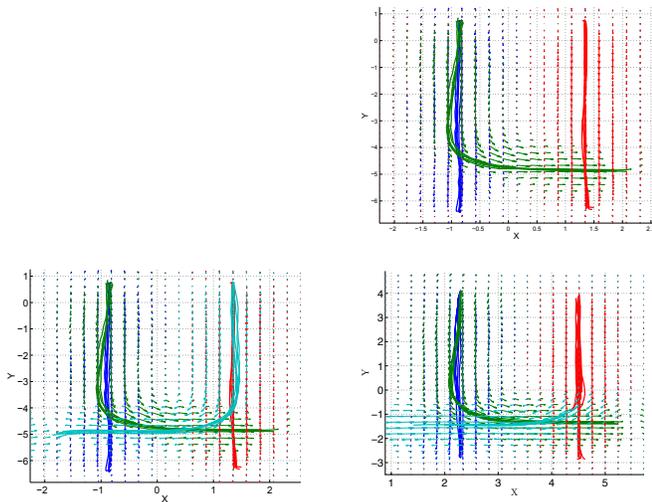


Fig. 4. Left: Test data. Top Right: Training data. Bottom Right: Output of GP-NBC on test data. GP-NBC detects the new behavior and successfully reclassifies it.

one and ten timesteps ahead), however, for the previously unseen behavior, GP-NBC reduced the error by 27% and 44% for the one and ten timestep ahead predictions.

## VI. CONCLUSIONS

We presented a computationally efficient algorithm, GP-NBC, for GP regression in the presence of changepoints. Unlike previous attempts to model changepoints using GPs, GP-NBC requires orders of magnitude less computation and so is fast enough to be used in real-time decision making applications. We derived polynomial (in relevant quantities including the covering number of the input space) sample complexity bounds on the number of inaccurate predictions by GP-NBC. Although the bounds are conservative, such bounds are not shared by competing algorithms considered in this paper. These results advance the state-of-the-art in non-stationary predictive modeling and enable fast online clustering for online decision making and learning architectures.

### APPENDIX A

#### PROOF OF EQUATION 6

The expected value of the log LRT test is given with respect to the true distribution  $p(y | H_1)$ ,  $\mathbb{E}_{p(y|H_1)}[\text{LRT}(y)]$ , however the log LRT is taken with respect to the model of the distribution  $\hat{H}_1$  and the null hypothesis  $H_0$ .

*Proof:*

$$\begin{aligned} \mathbb{E}[\text{LRT}(y)] &= \int_{p(y|H_1)} p(y | H_1) \log \frac{p(y | \hat{H}_1)}{p(y | H_0)} \\ &= \int_{p(y|H_1)} p(y | H_1) \log \frac{p(y | \hat{H}_1)}{p(y | H_1)} \\ &\quad + \int_{p(y|H_1)} p(y | H_1) \log \frac{p(y | H_1)}{p(y | H_0)} \\ &= D(H_1 \| H_0) - D(H_1 \| \hat{H}_1) \end{aligned}$$

### APPENDIX B

#### PROOF OF THE LEMMA 1

*Proof:* McDiarmid's Inequality states that the probability of a multivariate function of random variables deviating from its expected value is bounded as:

$$Pr\{|f(\theta_1, \dots, \theta_n) - \mathbb{E}[f(\theta_1, \dots, \theta_n)]| \geq \epsilon\} \leq \delta \quad (18)$$

$$\delta = 2\exp\left(-\frac{2\epsilon^2}{\sum_i c_i^2}\right) \quad (19)$$

where  $c_i = \sup f(\theta_1, \dots, \theta_i, \dots, \theta_n) - f(\theta_1, \dots, \hat{\theta}_i, \dots, \theta_n)$ . That is, different values of  $\theta_i$  can affect the output  $f(\theta_1, \dots, \theta_n)$  by no more than  $c_i$ . McDiarmid's Inequality becomes Hoeffding's Inequality if the function is a simple mean. We will now use McDiarmid's Inequality to bound the amount of change caused by adding a new datapoint to the GP.

Consider the GP mean equation

$$\mu(X) = K(X, X)(K(X, X) + \omega^2 I)^{-1} \bar{y} \quad (20)$$

where  $y \in [0, V_m]$  and  $\text{Var}(y) \leq V_m^2$ .  $K(X, X)$  is symmetric and positive semi-definite, so its eigenvectors are orthonormal to each other and all of its eigenvalues are nonnegative. It can be shown that  $K(X, X)$  and  $(K(X, X) + \omega^2)^{-1}$  have the same eigenvectors. Performing eigendecomposition,

$$\mu(X) = Q\Lambda Q^T Q(\Lambda + \omega^2 I)^{-1} Q^T \bar{y} \quad (21)$$

$$\mu(X) = Q\Lambda(\Lambda + \omega^2 I)^{-1} Q^T \bar{y} \quad (22)$$

Consider performing prediction only at the first input location  $x_1$  by pre-multiplying using a unit coordinate vector  $e_1 = [1, 0, \dots, 0]^T$ .

$$\mu(x_1) = e_1^T Q\Lambda(\Lambda + \omega^2 I)^{-1} Q^T \bar{y} \quad (23)$$

This is just a weighted sum of the observations  $y$ , with weights given by

$$\alpha = e_1^T Q\Lambda(\Lambda + \omega^2 I)^{-1} Q^T \quad (24)$$

It follows that  $\sum_i c_i^2 = \|\alpha\|_2^2 V_m^2$ . Then,

$$\|\alpha\|_2^2 = e_1^T Q\Lambda(\Lambda + \omega^2 I)^{-1} Q^T Q(\Lambda + \omega^2 I)^{-1} \Lambda Q^T e_1 \quad (25)$$

$$\|\alpha\|_2^2 = q_1 \Lambda(\Lambda + \omega^2 I)^{-1} (\Lambda + \omega^2 I)^{-1} \Lambda q_1^T \quad (26)$$

where  $q_1 = [Q_{11} \dots Q_{1n}]$  is the first row of  $Q$ . Therefore,

$$\|\alpha\|_2^2 = \sum_i q_{1i}^2 \left(\frac{\lambda_i}{\lambda_i + \omega^2}\right)^2 \quad (27)$$

However, by evaluating (24), the weight  $\alpha_1$  which corresponds to  $(x_1, y_1)$  is given by

$$\alpha_1 = \sum_i q_{1i}^2 \frac{\lambda_i}{\lambda_i + \omega^2}. \quad (28)$$

Since every term in (28) is greater than every respective term in the sum of (27), it follows that,

$$\|\alpha\|_2^2 \leq \alpha_1 \quad (29)$$

In order to finish the proof, an upper bound  $\alpha_1$  is derived. Note that for a Gaussian likelihood (that is assuming the measurement noise is Gaussian), the GP mean prediction



distance map is given by

$$r(\nu) = \theta \left( \log \left( \frac{1}{1-\nu} \right) \right)^{\frac{1}{2}}. \quad (36)$$

$\frac{1}{r(\nu)^d}$  grows polynomially with  $\frac{1}{\nu^d}$ . It follows  $N_c(U, r(\nu)) \sim O\left(f^p \left( V_m^2, \frac{1}{\epsilon_{\mathcal{E}}^2}, \log \left( \frac{1}{\delta_1} \right) \right)\right)$  where  $f^p$  is some polynomially bounded function of degree  $p \propto d$ . ■

#### ACKNOWLEDGMENT

The research was funded in part by Aerojet-Rocketdyne, ONR #N000141110688, AFOSR #FA9550-14-1-0399, and Ford.

#### REFERENCES

- [1] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.
- [2] K.-R. Müller, S. Mika, G. Rätsch, S. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–202, 2001.
- [3] G. Chowdhary, H. Kingravi, J. P. How, and P. Vela, "Bayesian nonparametric adaptive control of time varying systems using Gaussian processes," in *American Control Conference (ACC)*. IEEE, 2013.
- [4] H. A. Kingravi, G. Chowdhary, P. A. Vela, and E. N. Johnson, "Reproducing kernel Hilbert space approach for the online update of radial bases in neuro-adaptive control," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1130–1141, July 2012.
- [5] G. Chowdhary, H. Kingravi, J. P. How, and P. A. Vela, "Bayesian Nonparametric Adaptive Control using Gaussian Processes," *IEEE Transactions on Neural Networks*, 2013 (to appear).
- [6] M. P. Deisenroth, "Efficient reinforcement learning using Gaussian processes," Ph.D. dissertation, Karlsruhe Institute of Technology, 2010.
- [7] F. Pérez-Cruz, S. Van Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing," *arXiv preprint arXiv:1303.2823*, 2013.
- [8] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic Foundations of Robotics X*. Springer Berlin Heidelberg, 2013, pp. 475–491.
- [9] Z. Wang, K. Mülling, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human-robot interaction," *The International Journal of Robotics Research*, 2013.
- [10] Y. Saatçi, R. D. Turner, and C. E. Rasmussen, "Gaussian process change point models," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 927–934.
- [11] R. Garnett, M. A. Osborne, S. Reece, A. Rogers, and S. J. Roberts, "Sequential bayesian prediction in the presence of changepoints and faults," *The Computer Journal*, vol. 53, no. 9, pp. 1430–1446, 2010.
- [12] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [13] L. Csató and M. Opper, "Sparse on-line gaussian processes," *Neural Computation*, vol. 14, no. 3, pp. 641–668, 2002.
- [14] R. C. Grande, "Computationally efficient gaussian process changepoint detection and regression," Master's thesis, Massachusetts Institute of Technology, 2014.
- [15] N. Vaits and K. Crammer, "Re-adapting the regularization of weights for non-stationary regression," in *Proceedings of the 22nd international conference on Algorithmic learning theory*, 2011, pp. 114–128.
- [16] C. Monteleoni and T. Jaakkola, "Online learning of non-stationary sequences," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.
- [17] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.
- [18] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," *Advances in neural information processing systems*, vol. 2, pp. 881–888, 2002.
- [19] F. Stimberg, A. Ruttner, and M. Opper, "Bayesian inference for change points in dynamical systems with reusable states - a chinese restaurant process approach," *Journal of Machine Learning Research - Proceedings Track*, vol. 22, pp. 1117–1124, 2012.
- [20] J. W. Miller and M. T. Harrison, "A simple example of dirichlet process mixture inconsistency for the number of components," in *Advances in Neural Information Processing Systems*, 2013, pp. 199–206.
- [21] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–144, 2006.
- [22] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. Jordan, "Streaming variational bayes," in *Advances in Neural Information Processing Systems*, 2013, pp. 1727–1735.
- [23] T. Desautels, A. Krause, and J. W. Burdick, "Parallelizing exploration-exploitation tradeoffs with gaussian process bandit optimization," in *International Conference on Machine Learning (ICML)*, 2012.
- [24] Z. Ghahramani, "Nonparametric bayesian methods," in *Tutorial presentation at the UAI Conference*, 2005.
- [25] H. Maske, M. Matthews, A. Axelrod, H. Mohomadipanah, G. Chowdhary, C. Crick, and P. Pagilla, "Collaborative goal and policy learning from human operators of construction co-robots," in *NIPS workshop on Autonomous Learning Robots*. Montreal, CA: NIPS, 2015.
- [26] R. Grande, G. Chowdhary, and J. P. How, "Nonparametric adaptive control using gaussian processes with online hyperparameter estimation," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2013. [Online]. Available: [http://acl.mit.edu/papers/Grande13\\_CDC.pdf](http://acl.mit.edu/papers/Grande13_CDC.pdf)
- [27] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2006, pp. 513–520.
- [28] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Algorithmic learning theory*. Springer, 2005, pp. 63–77.
- [29] T. Van Erven and P. Harremoës, "Rényi divergence and kullback-leibler divergence," *arXiv preprint arXiv:1206.2459*, 2012.
- [30] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.
- [31] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.
- [32] W. Penny and S. Roberts, "Bayesian multivariate autoregressive models with structured priors," Oxford University, Tech. Rep., 2000.
- [33] S. Ferguson, B. Luders, R. Grande, and J. How, "Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions," in *Workshop on Algorithmic Foundations of Robotics*. Springer, 2014 (submitted).
- [34] S. Ferguson, "Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions," Master's thesis, Massachusetts Institute of Technology, 2014.
- [35] A. Girard, C. E. Rasmussen, J. Quintero-Candela, and R. Murray-smith, "Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting," in *Advances in Neural Information Processing Systems*. MIT Press, 2003, pp. 529–536.