Scalable Nonparametric Bayesian Learning for Dynamic Velocity Fields

Sunrit Chakraborty^{*1} **Aritra Guha** ^{2} **Rayleigh Lei**^{3} **XuanLong Nguyen**^{1}

^{1}Department of Statistics, University of Michigan, Ann Arbor, MI, USA
^{2}Data Science & AI Research, Chief Data Office, AT&T, Bedminster, NJ, USA
^{3}Department of Statistics, University of Washington, Seattle, WA, USA

Abstract

Learning and understanding heterogeneous patterns in complex spatio-temporal data is an important and challenging task across domains in science and engineering. In this work, we develop a model for learning heterogeneous and dynamic patterns of velocity field data, motivated by applications in the transportation domain. We draw from basic nonparametric Bayesian modeling elements such as the infinite hidden Markov model and Gaussian process and focus on making the learning of such a stochastic model scalable for voluminous and streaming data. This is achieved by employing sequential MAP estimates from the infinite HMM model, an efficient sequential sparse GP posterior computation, and refinement of the estimates using the Viterbi algorithm, which is shown to work effectively on a careful simulation study. We demonstrate the efficacy of our techniques to the NGSIM dataset of complex multi-vehicle interactions.

1 INTRODUCTION

A common challenge arising in modern applications is the presence of a large amount of data available via spatio-temporal dynamics generated in a highly heterogeneous and potentially fast-paced environment, yet there is a need to extract meaningful and interpretable patterns out of such complexities in a computationally efficient way. While there are numerous examples in a variety of domains, e.g., Sarkar et al. [2019], Nelson and Widrow [2022], Angell and Sheldon [2018], McDowell et al. [2017], Rubenstein and Hobson [2004], Hooten and Johnson [2017] to name a few, what motivates our present work is the analysis of traffic flow patterns out of high-volume and streaming measurements of vehicles passing through a busy thoroughfare.

A visitor to a large city may be initially shocked upon observing a bewildering range of individual driving behaviors and of cars moving in varying speeds and directions, competing and challenging for an open lane at any given moment. Underneath this seemingly intractable complexity, one may eventually find the calming ebbs and flows of movements regulated by traffic control and the rhythm of the day. Such patterns of traffic flows can be represented by a vector field indexed on a two-dimensional plane.

Define a vector field (interchangeably *velocity field*) as a function $f : \mathcal{X} \to \mathbb{R}^2$, such that $f(x)$ records the velocity vector for a car in location $x \in \mathcal{X}$. Unless there is an unusual disruption, one expects that the velocity vector varies smoothly, both in direction and magnitude, through the spatial domain. Thus, we adopt the viewpoint that a smooth vector field is a useful mathematical device to describe the current state of traffic flow at any given moment. Gaussian process (GP) Rasmussen and Williams [2006] is a useful tool for modeling such vector fields, and has been utilized in recent work in motion modeling [Barão and Marques, 2017, Klinger et al., 2016, Ellis et al., 2009] or traffic data analysis [Guo et al., 2019, Chen et al., 2016]. However, these works do not explicitly capture the temporal dynamics, even though the daily time or season may be important factors for a driver to consider for safe and efficient driving. This calls for stochastic modeling tools to explicitly represent the temporal nature of spatial traffic patterns. Moreover, such a model must be learned from potentially high-volume, heterogeneous, and streaming data.

We focus on NGSIM traffic data at Lankershim Boulevard (LB), Los Angeles for our application (http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm). Figure 1 illustrates the notion of velocity fields in this context — in particular Fig. 1(C) captures a left turn from LB to University Hollywood Dr. The data comprises 1.5 million observations spread over 30 minutes. Understanding the traffic patterns is essential in many applications, including urban planning, construction, and real-time traffic regulation. This requires interpretable methods to

[*] Corresponding author. Email: sunritc@umich.edu

Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), PMLR 216:282–292.

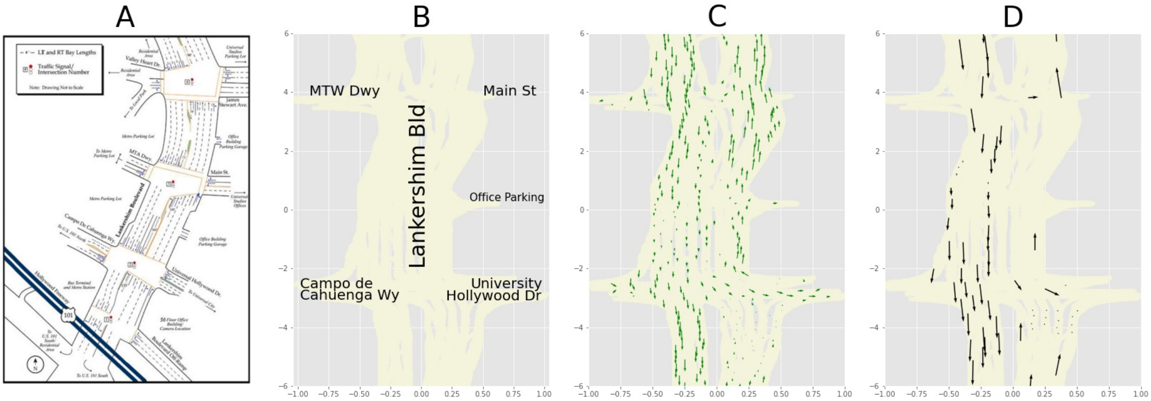


Figure 1: Notion of velocity fields: (A) Image of Lankershim Boulevard, the region under study; (B) Representation of a subset of this region under local coordinate system; (C) A particular velocity field (traffic pattern); (D) Real observations at a specific time point, believed to arise from the velocity field in (C) (Arrow lengths not comparable across images).

meaningfully extract information on driving behavior. However, the sheer size of the data makes many existing model-based interpretable inference methods inapplicable.

Contributions We aim to achieve fast and accurate inference underlying a stochastic model for smooth vector field patterns arising in a heterogeneous and dynamic environment. Our contributions are two-fold. First, we propose a Bayesian nonparametric model, which at the high level can be cast as an infinite hidden Markov model (iHMM) on a state space of multi-dimensional vector fields described by smooth Gaussian processes. Second, we develop a novel algorithm for scalable Bayesian inference on the proposed model. The model and algorithm are demonstrated via extensive simulations and application to the NGSIM data.

More specifically, a discrete-time hidden Markov chain that operates on the state space of the latent velocity fields is constructed to capture the temporal dynamics of spatial patterns. This modeling brings forward a novel aspect to the application perspective, which is potentially useful in improving autonomous vehicles based on interpretable learned patterns. Moreover, to account for the highly heterogeneous environment, we allow the number of hidden states to be unbounded. This is achieved by drawing from the powerful nonparametric Bayesian techniques of iHMM and hierarchical Dirichlet processes (HDP) [Beal et al., 2002, Teh et al., 2006]. On the algorithmic front, our contribution includes deriving sequential MAP estimates from the infinite HMM model and efficient sequential GP posterior computation techniques. The two novel steps of the proposed method consist of (1) forward pass step - sequentially updating the state labels by MAP estimates based on the sequential posterior of the model and (2) refinement step - to remove redundant clusters found by the greedy forward pass. Simultaneous observations at a large number of spatial locations in the dataset suggest the involvement of large covariance for GP computations, inverting which is computationally prohibitive and overcome by a Sparse GP approximation technique with fixed knots to replace numerical optimization with closed-form updates. These innovations allow us to analyze over 50,000 total observations in under 2 minutes.

Related work Prior works that combine both Dirichlet Process and GP modeling elements to extract an unknown number of traffic patterns include [Guo et al., 2019, Joseph et al., 2011, Chen et al., 2016], but these models do not consider the temporal nature of the data and are not scalable for large datasets. On the other hand, [Jung and Park, 2020] consider HMM dynamics with GP emission at scale, but as we show in Section 5, may drastically end up underestimating the number of underlying clusters thereby (over)generalizing patterns too much; some other similar works ([Henter et al., 2012, Nakamura et al., 2017, Nagano et al., 2018] which combine HMM and GP elements also suffer from the heavy computational burden. While some of the aforementioned works employ Bayesian inference techniques via MCMC [Gelfand and Smith, 1990, Fox et al., 2009], which may lead to the computational inefficiency for large datasets, a number of them including [Jung and Park, 2020] use stochastic variational inference (VI)([Blei et al., 2003, Foti et al., 2015], [Jordan et al., 1999, Blei and Jordan, 2006, Hoffman et al., 2013, Mandt et al., 2017]) techniques which may favor simpler models unable to capture the underlying heterogeneity in the data.

The remainder of the paper is as follows. Section 2 briefly describes the modeling elements employed in this work. Section 3 formalizes the data representation and describes our model. Section 4 describes our proposed algorithm. An extensive simulation study is given in Section 5, followed by experimental results on the NGSIM traffic data in Section 6.

Notations The set $\{1, \ldots, n\}$ is denoted by $[n]$. For a function $\phi : \mathcal{X} \to \mathbb{R}$ for $\mathcal{X} \subset \mathbb{R}^D$ and $\boldsymbol{X} = (x_1, \ldots, x_n) \subset \mathcal{X}$, let $\phi(\boldsymbol{X})$ denote the column vector $(\phi(x_1), \ldots, \phi(x_n))^\top$.

For a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and given $\boldsymbol{X}_1, \boldsymbol{X}_2 \subset \mathcal{X}$, where $\boldsymbol{X}_i = (x_{i,1}, \ldots, x_{i,n_i})$, we use $K(\boldsymbol{X}_1, \boldsymbol{X}_2)$ to denote the $n_1 \times n_2$ matrix, whose (i,j)th element is $K(x_{1,i}, x_{2,j})$. If $\boldsymbol{X}_1 = \boldsymbol{X}_2$, we write it simply as $K(\boldsymbol{X}_1)$.

2 BACKGROUND

We discuss the key modeling elements that we use in this work. Further details are provided in Appendix A.

Gaussian Process (GP): A stochastic process $\{f(x) : f(x) \in \mathbb{R},\ x \in \mathcal{X}\}$ is called a GP [Rasmussen and Williams, 2006] with mean function $m(\cdot)$ and covariance kernel $K(*,*)$ if for any finite $\boldsymbol{X} := \{x_1, \ldots, x_k\} \subset \mathcal{X}$,

$$f(\boldsymbol{X}) \sim \mathcal{N}_k(m(\boldsymbol{X}), K(\boldsymbol{X})). \tag{1}$$

This is denoted as $f \sim \text{GP}(m, K)$. It is common practice to assume $m = 0$. In our work, the index space is the spatial region. To incorporate measurement error, it is common practice to model the observations as corrupted by white noise in which case the (n i.i.d.) observations y_i corresponding to spatial region x_i are modeled as $y_i = f(x_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, with $f \sim \text{GP}(m, K_\theta)$ (where θ includes all the kernel parameters); then the posterior distribution of f can be expressed in terms of its finite-dimensional distributions. For $\boldsymbol{X}^* = \{x_i^*\}_{i \in [n^*]}$, we have

$$f(\boldsymbol{X}^*)|\boldsymbol{X}, \mathbf{y} \sim N(\mu^*, \Sigma^*)\,, \text{where}$$

$$\begin{aligned}
\mu^* &= \mu + K_\theta(\boldsymbol{X}^*, \boldsymbol{X}) A_\theta^* (\mathbf{y} - \mu) && (2)\\
\Sigma^* &= K(\boldsymbol{X}^*) - K_\theta(\boldsymbol{X}^*, \boldsymbol{X}) A_\theta^* K_\theta(\boldsymbol{X}, \boldsymbol{X}^*) && (3)\\
A_\theta^* &= \left[K_\theta(\boldsymbol{X}) + \sigma^2 I\right]^{-1}. && (4)
\end{aligned}$$

Note that computation of A_θ^* requires the inversion of a $n \times n$ matrix. The parameters (θ, σ^2) can be estimated by maximizing the marginal likelihood, which is also challenging for large n.

Sparse Gaussian Process (SGP): This is a variational approximation to GP, which can overcome the computational bottleneck of traditional GP [Titsias, 2009, Hensman et al., 2013]. The idea is quite simple: consider a set of spatial points $\boldsymbol{Z}$ (of size m), to be referred to as knots or inducing points and a variational distribution of $f(\boldsymbol{Z}) \sim \phi(f_Z) \equiv N(\mu_m, \Sigma_m)$, using which we can get an approximation for the true posterior. The objective is to maximize the evidence lower bound by optimizing over the inducing points $\boldsymbol{Z}$, the variational parameters μ_m, Σ_m, and other parameters (like those of kernel or likelihood). Given these parameters, the posterior of f over $\boldsymbol{X}^*$ is given by

$$f(\boldsymbol{X}^*)|\boldsymbol{X}, \mathbf{y} \sim N(\mu^*, \Sigma^*)\,, \text{where}$$

$$\begin{aligned}
\mu^* &= K(\boldsymbol{X}^*, \boldsymbol{Z}) K(\boldsymbol{Z})^{-1} \mu_m && (5)\\
\Sigma^* &= K(\boldsymbol{X}^*) - K(\boldsymbol{X}^*, \boldsymbol{Z}) K(\boldsymbol{Z})^{-1} K(\boldsymbol{Z}, \boldsymbol{X}^*)\\
&\quad + K(\boldsymbol{X}^*, \boldsymbol{Z}) K(\boldsymbol{Z})^{-1} \Sigma_m K(\boldsymbol{Z})^{-1} K(\boldsymbol{Z}, \boldsymbol{X}^*). && (6)
\end{aligned}$$

Given the inducing points $\boldsymbol{Z}$ and the kernel, the variational lower bound for the marginal likelihood can be optimized to obtain analytical solutions for the variational parameters μ_m, Σ_m:

$$\begin{aligned}
\mu_m &= \frac{1}{\sigma^2} K(\boldsymbol{Z}) \tilde{A} K(\boldsymbol{Z}, \boldsymbol{X}) \mathbf{y} && (7)\\
\Sigma_m &= K(\boldsymbol{Z}) \tilde{A} K(\boldsymbol{Z}), \quad \text{where} && (8)\\
\tilde{A} &= \left(K(\boldsymbol{Z}) + \frac{1}{\sigma^2} K(\boldsymbol{Z}, \boldsymbol{X}) K(\boldsymbol{X}, \boldsymbol{Z})\right)^{-1}. && (9)
\end{aligned}$$

Note that for SGP, we need to invert matrices of size $m \times m$, which is much faster when $m << n$. Typically, batch gradient-based methods can be employed to optimize all the parameters together, which include kernel parameters, σ^2, $\boldsymbol{Z}, \mu_m, \Sigma_m$.

Infinite Hidden Markov Model (iHMM): The infinite HMM model of [Beal et al., 2001] is a Bayesian nonparametric model which allows a countably infinite number of components. It was subsequently shown to be an instance of the Hierarchical Dirichlet process HMM model of [Teh et al., 2006]. iHMM uses a local allocator (which can select one of the already explored states using the current state and the transition count matrix N_t) and a global allocator (which can select one of the observed states or a new unseen state) at each time point t to decide the state at the next time point s_{t+1}. The oracle variable o_t (binary-valued) is the indicator of whether the global allocator was used at time t and M_t is the count vector capturing the number of times till time t that a particular state was visited using the global allocator. The global allocator requires s_t and M_t. Note that the assignment of state at every time depends on the current oracle variable and the chosen allocator. A new state can be reached only using the global allocator.

We briefly describe the infinite HMM prior structure. Given parameters (α, β, γ), to draw a sample of a sequence of states $\{s_t\}$ from this prior, we start by setting $s_1 = 1$. Initialize the oracle variable $o_1 = 1$. Given $\{s_{1:t}\}$ and $\{o_{1:t}\}$, let K_t be the number of distinct elements in $\{s_{1:t}\}$ and N_t and M_t denote the transition count matrix (i.e. $(N_t)_{i,j} = \sum_{r \in [t-1]} \mathbf{1}(s_r = i, s_{r+1} = j)$) and the oracle count vector (i.e. $(M_t)_i = \sum_{r \in [t]} \mathbf{1}(s_r = i, o_r = 1)$) respectively. Given $s_t = i$, s_{t+1} and o_{t+1} are generated as follows. For convenience, we use the short hands $N_{i\cdot} = \sum_j N_{ij}$ for N_t

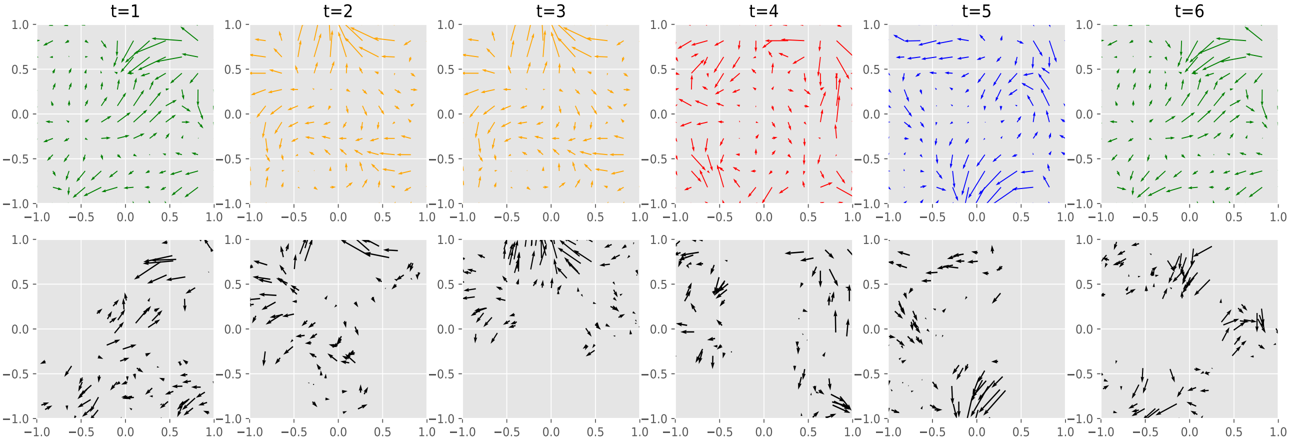


Figure 2: Example of data generated from our model with $T = 6$ time points and $K = 4$ true latent functions (shown by colors). (top) true vector fields at each time and (bottom) noisy observations at corresponding times over different locations.

(similarly $M_\cdot$), dropping subscript t.

$$p\left(\begin{array}{c|c} s_{t+1}=j & s_t=i, \\ o_{t+1}=0 & N_t, M_t \end{array}\right) = \begin{cases} \frac{\alpha+N_{ii}}{N_{i\cdot}+\alpha+\beta} & j = i \\ \frac{N_{ij}}{N_{i\cdot}+\alpha+\beta} & j \neq i; j \leq K_t \end{cases} \tag{10}$$

$$p\left(\begin{array}{c|c} s_{t+1}=j & s_t=i, \\ o_{t+1}=1 & N_t, M_t \end{array}\right) = \begin{cases} \frac{\beta}{N_{i\cdot}+\alpha+\beta}\,\frac{M_j}{M_\cdot+\gamma} & j \leq K_t \\ \frac{\beta}{N_{i\cdot}+\alpha+\beta}\,\frac{\gamma}{M_\cdot+\gamma} & j = K_t + 1. \end{cases} \tag{11}$$

Under this mechanism, starting at the current state $s_t = i$, the system can jump to one of the previously explored states $[K_t]$, either directly (with $o_{t+1} = 0$) or through the oracle (with $o_{t+1} = 1$) or might explore a new state $K_t + 1$, for which it must go through the oracle. See Figure 1 in Appendix A for an illustration. We write $s_t \sim \text{iHMM}(\alpha, \beta, \gamma)$, to indicate the infinite HMM as the prior on $\{s_t\}$.

3 THE MODEL

We assume there are (an unknown number) K^* underlying functions $f_1, \ldots, f_{K^*} : \mathcal{X} \to \mathbb{R}^P$, each function modeling a velocity field. The temporal dynamics of the system are controlled through an HMM, in particular, assume $\{s_t\}_{t\in[T]}$ follows Markov dynamics with transition matrix $\Pi_{K^*\times K^*} = (\pi_{ij})_{i,j}$, i.e. $p(s_t = j|s_{t-1} = i) = \pi_{ij}$. Given the *state* s_t of the Markov chain at time t, the system follows the velocity field f_{s_t} and hence, the observations at time t are given by

$$y_{t,j} = f_{s_t}(x_{t,j}) + \epsilon_{t,j}, \quad j \in [n_t], t \in [T] \tag{12}$$

where $\epsilon_{t,j} \overset{iid}{\sim} N_P(\mathbf{0}, \Sigma)$ is the noise and $\{x_{t,j}\}_{j\in[n_t]}$ is the set of fixed spatial locations where observations are available. We take $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_P^2)$ as a diagonal matrix. When $P = 1$, we place an **infinite HMM** prior on the state sequence $\{s_t\}$ and independent mixture of **Gaussian process** priors on the (infinite) sequence $f_1, f_2, \ldots$ of functions $f_k \sim \mathcal{GP}(\mathbf{0}, K_{\theta_k})$ (with respective kernel parameters θ_k). Guided by our requirement to extract sufficiently *smooth* functions, we choose the popular Radial Basis Function (RBF) kernel $K_\theta(x, x') = \sigma_0^2 \exp\left\{-\|x - x'\|^2 / 2\ell_0^2\right\}$ where $\theta = (\sigma_0^2, \ell_0)$ embodies the kernel parameters. For $P > 1$, we place *independent* such Gaussian process priors across different output dimensions. For computational efficiency, we approximate the Gaussian Process prior with **Sparse Gaussian Process**. The complete model, referred to as iHMM-GP, is thus given as follows.

$$\begin{aligned} \{s_t\} &\sim \text{iHMM}(\alpha, \beta, \gamma) \\ f_k &\sim \otimes_P \text{SGP}(\mathbf{0}, K_{\theta_k}), \quad \text{ind. } k \geq 1 \\ (\boldsymbol{Y}_t)_j | \boldsymbol{X}_t, s_t &\sim N_{n_t}\left((f_{s_t})_j(\boldsymbol{X}_t), \sigma_j^2 I\right), \quad \text{ind. } j \in [P] \end{aligned} \tag{13}$$

where the second line indicates that each f_k consists of P functions (one for each output dimension), each drawn independently from a Sparse Gaussian process and the last line indicates that the jth dimension of the observations is modeled as Gaussian, independently across these dimensions, based on the current state. We treat (α, β, γ) as hyperparameters of the model and estimate $\{\sigma_j^2\}_{j\in[P]}$ and $\{\theta_k\}$ from the data. Figure 2 shows a simulated example from our model. We mention the following remarks.

Remark: (i) The model assumes that Σ represents the underlying error in measurement and therefore it is reasonable to assume the noise level Σ to be spatio-temporally invariant. (ii) The model is flexible to allow observed positions to vary across time points. This in turn also enables efficient prediction at unobserved locations present in test data. This is a key aspect of the model that helps capture population driving behavior.

4 INFERENCE

Our method of inference comprises of two steps, the first of which is a novel two-pass algorithm over the data. The two steps after initialization can be summarized as (1) performing a forward pass by updating the parameters using sequential greedy MAP estimation, followed by a refinement step, which uses the Viterbi algorithm to reassign states with the goal of removing redundant clusters; and (2) iterating between updating the latent states given the current components, using Viterbi, and updating the posteriors of the estimated components, given the current states. The outline of the algorithm is given in Algorithm 1. For notational simplicity, we write the steps for $P = 1$ and take $n_t = n$ for all t.

Algorithm 1 Proposed algorithm for iHMM-GP model

```
Input: Data $\mathcal{D}_{1:T} = \{(\boldsymbol{X}_t, \boldsymbol{Y}_t)\}_{t=1}^{T}$.
Require Tuning parameters $\boldsymbol{Z}, m_0, (\alpha, \beta, \gamma), n_0, L_{\max}$.

Initialization:
Get $\hat{\Sigma}$ and $\{\tilde{\theta}_t\}$ and $p(\boldsymbol{Y}_t|\boldsymbol{X}_t)$ $\forall t \in [T]$ (see Section 4.1)

Step 1 Using forward pass and refinement
  1. Create blocks: $\mathbf{B}_j = \{\mathcal{D}_t : (j-1)m_0+1 \leq t < m_0 j\}$
  2. $\forall\, j$, fit forward pass (see Section 4.2) on $\mathbf{B}_j$
  3. $\forall\, j$, use refinement step (Section 4.3)
  4. Combine the results for different $j$ (see last paragraph
     of 4.3) to obtain $K$ and $s_{1:T}$
Step 2 Iterate $L_{\max}$ times or till convergence:
  • Update $s_{1:T}$ given the clusters, using Viterbi
  • Update the SGPs for each cluster, given $s_{1:T}$
Output: $s_{1:T}$.
```

4.1 INITIALIZATION

We first fit GP to $(\boldsymbol{X}_t, \boldsymbol{Y}_t)$ separately for each $t \in [T]$. We extract the following information from these fitted models: (1) the noise variance $\hat{\sigma}_t^2$ for each t, (2) the estimated optimal kernel parameters $\tilde{\theta}_t$ for each t and (3) the marginal log-likelihoods $p(\boldsymbol{Y}_t|\boldsymbol{X}_t, \tilde{\theta}_t)$ for these T models. From (1) we compute the overall estimated noise variance $\hat{\sigma}^2$ as the empirical mean of $\{\hat{\sigma}_t^2\}$. We fix and use this $\hat{\sigma}^2$ throughout. The following steps after initialization are performed on the meta-model comprised of the output of initial GP estimates.

4.2 FORWARD PASS

The idea of the forward pass is to traverse the data from $t = 1$ to $t = T$, sequentially making a greedy decision, based on the estimated variables so far, whether to add the current $\mathcal{D}_t$ to an existing cluster or create a new one, i.e., choose $\hat{s}_{t+1}, \hat{o}_{t+1}$ as

$$\arg\max p(s_{t+1}, o_{t+1}|s_{1:t} = \hat{s}_{1:t}, o_{1:t} = \hat{o}_{1:t}, \mathcal{D}_{1:(t+1)}).$$

At time $t+1$, based on the current estimates till time point t, we use the iHMM prior and the GP models to make this decision. By Bayes theorem,

$$\begin{aligned} &p(s_{t+1}, o_{t+1}|s_{1:t}, o_{1:t}, \mathcal{D}_{1:(t+1)}) \\ &\quad \propto p(s_{t+1}, o_{t+1}|s_{1:t}, o_{1:t}) p(\mathcal{D}_{t+1}|s_{1:(t+1)}, \mathcal{D}_{1:t}) \end{aligned} \quad (14)$$

where the first term on the right is given by the iHMM prior structure, given in Equation (10) and (11). Note that the N_t, M_t used in (10) and (11) only depends on $s_{1:t}$ and $o_{1:t}$.

For the second term, denote $\mathcal{D}_{1:t}^{(k)} = \{\mathcal{D}_r : r \leq t, s_r = k\}$ and similarly for other quantities and let K_t be the number of clusters found using data $\mathcal{D}_{1:t}$.; then, for $k \leq K_t$ (one of the existing clusters)

$$\begin{aligned} p(\mathcal{D}_{t+1}|s_{t+1} = k, s_{1:t}, \mathcal{D}_{1:t}) &= \int p(\mathcal{D}_{t+1}|f) Q(f|\mathcal{D}_{1:t}^{(k)}, \theta_k) \\ &= \mathcal{N}(\boldsymbol{Y}_{t+1}|\mu_t^{(k)}, \Sigma_t^{(k)} + \sigma^2 I) \end{aligned} \quad (15)$$

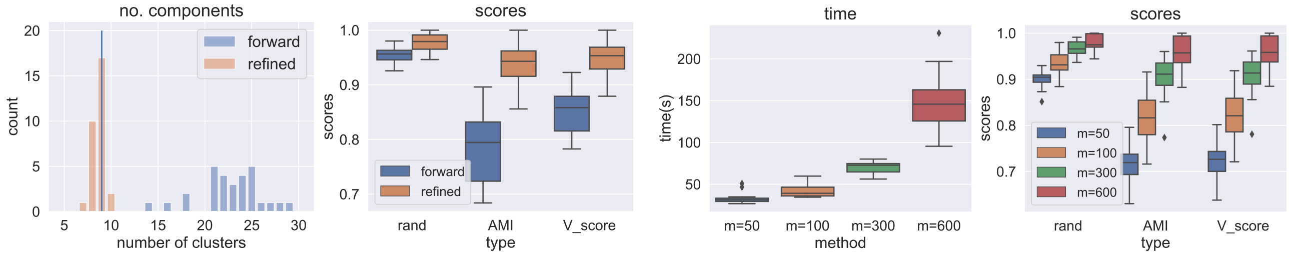
where Q is the law of the GP. Here $\mu_t^{(k)}$ and $\Sigma_t^{(k)}$ can be obtained using Equations (2) and (3), by replacing $\boldsymbol{X}^*$ with $\boldsymbol{X}_{t+1}$, $\boldsymbol{X}$ with $\boldsymbol{X}_{1:t}^{(k)}$ and $\mathbf{y}$ with $\boldsymbol{Y}_{1:t}^{(k)}$. For $k = K_t + 1$ (new cluster), there is no previous time point data to update the posterior and hence we have

$$\begin{aligned} p(\mathcal{D}_{t+1}\ |s_{t+1} = k, s_{1:t}, \mathcal{D}_{1:t}) &= \int p(\mathcal{D}_{t+1}|f) Q(f|\theta_k) \\ &= \mathcal{N}(\boldsymbol{Y}_{t+1}|\mathbf{0}, K_{\theta_k}(\boldsymbol{X}_{t+1}) + \sigma^2 I) \end{aligned} \quad (16)$$

which is the marginal likelihood computed during initialization. In this case, we use $\theta_k = \tilde{\theta}_{t+1}$. Using the above, we sequentially estimate the state and oracle variables, starting with $\hat{s}_1 = 1, \hat{o}_1 = 1$. Note that computation of this second term is costly as it requires the inversion of a large matrix. In particular, consider Equation (15), whose computation involves inverting a matrix of size $\tilde{n}_t^{(k)} \times \tilde{n}_t^{(k)}$, which requires $O((\tilde{n}_t^{(k)})^3)$ computations. Here $\tilde{n}_t^{(k)} = \sum_{r \leq t : s_t = k} n_t$. This keeps growing for each t and requires K_t (also growing relative to t) such computations. Therefore, we use **sparse Gaussian process** with a fixed set of inducing points $\boldsymbol{Z}$ of size $m \lesssim n_t$ to speed up the process (see Appendix B for details). This reduces the computational complexity of the key matrix inversion step to $O(m^3)$ instead, thus reducing the overall computational complexity of the algorithm.

4.3 REFINEMENT STEP

After finishing the forward pass, we have estimates of $s_{1:T}$, based on which we have K_T components, each with an SGP.



(a) Compare forward pass with refinement

(b) Effect of block size ($m = 600$ refers to no use of blocks)

Figure 3: Simulation Study 1.

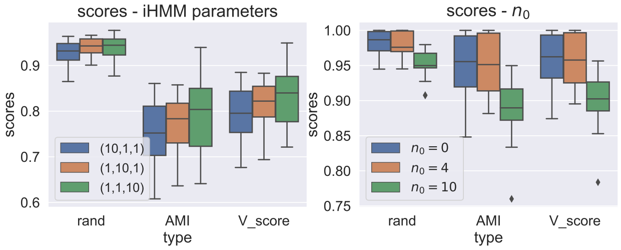


Figure 4: Simulation Study 2 - Effect of iHMM parameters (left) and parameter n_0 (right).

We propose a refinement step with the goal of identifying and removing redundant clusters, if any.

For cluster k and K current clusters, to decide if it is redundant given the others, we temporarily remove it, this gives new $N^{(-k)}, M^{(-k)}$. Let $\tau_k = \{t \in [T] : s_t = k\}$ and $\tau_{-k} = [T] \setminus \tau_k$. We propose to use Viterbi to reassign $\{s_t\}_{t \in \tau_k}$ given $\{s_t\}_{t \in \tau_{-k}}$, treating it as a HMM with K states ($K - 1$ remaining ones and one extra *new* state). The transition probabilities are constructed using iHMM with $N^{(-k)}$ and $M^{(-k)}$ and the emission probabilities are based on Equation 15 for remaining states and marginal likelihood for the *new* state. If the number of times a new cluster is required exceeds a threshold n_0, we retain it; otherwise, we dissolve this and reassign $\{s_t\}_{t \in \tau_k}$ to the other $K - 1$ clusters using Viterbi. This tuning parameter n_0 incorporates prior knowledge about the sizes of the clusters and works as a truncation mechanism to reduce the number of clusters and allows the user to ignore smaller clusters (by increasing the value of n_0). In the absence of prior knowledge, we set it at 0. This implies that for a cluster (from the forward pass) even if there is a single time point that requires a new cluster after removing it (during the refinement step), we choose to retain it as a separate cluster. In practice, it may be tuned using out-of-sample log-likelihood of the fitted model by starting from a smaller value followed by a gradual increment based on the log-likelihood score. This parameter has a similar role as *minPts* parameter in DBSCAN clustering [Ester et al., 1996]. We go through the clusters in their increasing size.

Use of Blocks: Following the forward pass and refinement steps we note that (1) the parameter estimates may depend on the particular order of the training data, and (2) the computational burden is high when T is high since the forward pass cannot be enabled in parallel loops. To mitigate this, we propose to partition the data into distinct blocks, $\mathbf{B}_j = \{\mathcal{D}_t : (j-1)m_0 + 1 \leq t < m_0 j\}$, of size m_0 and perform forward pass and refinement on each of these blocks independently in parallel. We combine the results from different blocks based on K-Means and Silhoutte coefficient. See Appendix C for additional details.

The tuning parameters include the set of inducing points $\boldsymbol{Z}$, the block size m_0, the iHMM parameters (α, β, γ), the threshold parameter n_0 in the refinement step, and $L_{\max}$, the maximum number of iterations in Step 2.

5 SIMULATION STUDY

Next, we present simulation studies to explicate the performance of our model and proposed algorithm. Mean of results for each experiment (over 30 replications) is reported.

5.1 SIMULATION SETTINGS

Each of the experiments is controlled by T (total number of time points), n (the average number of observations per time point), and σ^2 (the noise level). Given K^* true functions $f_k^* : \mathbb{R}^D \rightarrow \mathbb{R}^P$ and a transition matrix Π^*, we generate data from an HMM model with observations arising as in Equation (12). A particular instance of the training data for $D = P = 2$ is demonstrated in the bottom row of Figure 2. We refer to the proposed method in this work as *iHMM-GP*.

For evaluating performance, we consider the prediction of labels, for which we compare the estimated labels on training data with the true labels based on (a) RAND index, (b) Adjusted Mutual Information (AMI), and (c) V Score. The Rand Index [Hubert and Arabie, 1985] has a value between 0 and 1, with 0 indicating that the two data clusterings do not agree on any pair of points and 1 indicating they are the same. Mutual information of a clustering indicates the

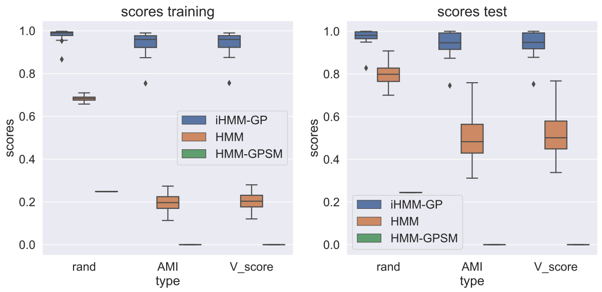


Figure 5: Label estimation performance in Experiment 1.

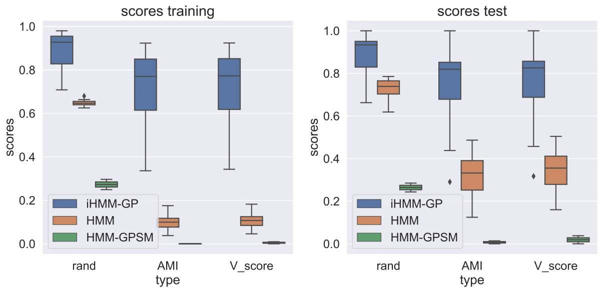


Figure 6: Label estimation performance in Experiment 2.

reduction in the entropy of class labels if the cluster labels are known. Adjusted mutual information [Vinh and Epps, 2010] accounts for chance - it is a value between 0 and 1, taking 1 when clusterings are identical and 0 when the mutual information between them is equal to the value expected due to chance alone. V Score [Rosenberg and Hirschberg, 2007] is also a similarity index, taking values between 0 (dissimilar) and 1 (identical clusters), and can be seen as the average of two other scores - (a) completeness and (b) homogeneity. To evaluate the estimation of the number of components, we use $d(K, K^*)$ as the mean of absolute deviations $|K - K^*|$, across multiple repetitions for each setting. Additional details about all experiments along with a few others are provided in Appendix D.

5.2 EFFECT OF TUNING PARAMETERS

In this section, we study the effects of different tuning parameters and steps on the algorithm — under $D = P = 2$ with $K^* = 9$ setting. For tuning parameters, we set the defaults as $n_0 = 3, (\alpha, \beta, \gamma) = (3, 3, 3)$ and $m_0 = 200$ (when $T > 200$) and no blocking for smaller T. We chose a 10×10 uniform grid as the inducing points for the SGP.

- **Effect of Refinement Step**: We study how the refinement step improves the performance after the forward pass, under the setting $T = 100, n = 50, \sigma^2 = 1$. Figure 3a demonstrates that the refinement step addresses the overestimation of the number of components, while also improving training accuracy drastically.

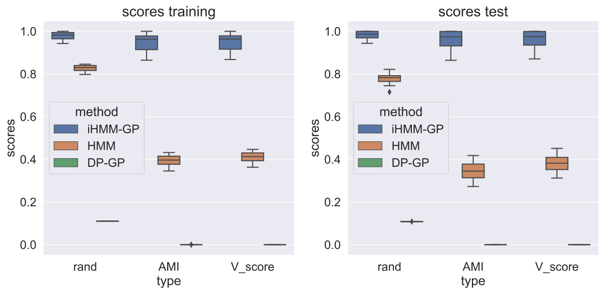


Figure 7: Label estimation for $D = P = 2$ case, for $n = 30$.

- **Effect of block size** m_0: Under $T = 600, n = 50, \sigma^2 = 2$ setting, we investigated the effect of block size m_0 on the performance of the algorithm, comparing between $m_0 = 50, 100, 300$ and no blocking. Figure 3b suggests that higher m_0 provides better performance at the cost of time. While no blocking has the best training accuracy, it takes a significantly longer time to fit the model.
- **Effect of iHMM parameters**: Here we study the effect of the iHMM hyperparameters on the performance of the algorithm under $T = 100, n = 50, \sigma^2 = 2$ setting. We compare the following 3 settings for (α, β, γ): (a) $(10, 1, 1)$, (b) $(1, 10, 1)$ and (c) $(1, 1, 10)$. Figure 4 (left) shows that performances for the 3 settings are similar; however, higher γ or β (which promote creating more clusters) have slightly better performance. For the number of clusters, we found that the results after the refinement step are comparable.
- **Effect of** n_0: We study the effect of the parameter n_0, used in the refinement step. Under $T = 150, n = 50, \sigma^2 = 2$, we compare between the choices $n_0 = 0, 5, 10$. Figure 4 (right) demonstrates that performances for the first two are similar but worsen in the last case, thus indicating that, while performance is similar for smaller values (with values closer to 0 allowing small clusters to be retained), performance will deteriorate significantly if n_0 is chosen too large.

5.3 COMPARISONS FOR $D = P = 1$

We use $K^* = 4$ true components, which are functions on $(0, 1)$. We compared our results with HMM with Gaussian emission (G-HMM) and HMM-GPSM [Jung and Park, 2020] (both setting $K \leq 8$). While exact-fitted setting for these baselines provided similar results, we present the overfitted case which is the typical case in practice. For our method, we use $m_0 = 100, n_0 = 3$, the iHMM parameters were set to $(3, 2, 1)$ and $\boldsymbol{Z}$ as 60 equi-spaced points.

For *Experiment 1*, we take $T = 400, n = 60, \sigma^2 = 4$ without any kind of spatial censoring. The results are given

Table 1: Performance comparison for simulations in terms of time, number of components, and average log-likelihood.

	Model	**time(s)**	d(K,K*)	**log lik**
Exp 1	iHMM-GP	15.185	**0.1333**	**-127.61**
	HMM-GPSM	403	1.00	-130.73
	G-HMM(8)	**5.93**	4.00	-129.68
Exp 2	iHMM-GP	13.853	**1.13**	**-84.47**
	HMM-GPSM	375	3.00	-87.08
	G-HMM(8)	**5.76**	4.00	-86.54
Exp 3	iHMM-GP	53.90	**0.9**	**-84.46**
	DP-GP	600.97	3.00	-95.77
	G-HMM(15)	**6.32**	6.00	-87.64

in Figure 5. For *Experiment 2*, we keep the same T, σ^2 but censor observations in one-third spatial region to reduce n to 40. The results are given in Figure 6. Results in Table 1 show that iHMM-GP is able to provide a scalable and efficient estimate of the number of components in comparison to the other methods, thereby lending credibility to it as a scalable and interpretable algorithm. HMM-GPSM identifies a much lower number of clusters than the truth. Additionally, the likelihood values of the optimal estimator for HMM-GPSM are also lower than that of iHMM-GP. This indicates that the output of HMM-GPSM is unable to capture the generating distribution effectively, thereby compromising the statistical interpretation of the generating mechanism.

5.4 COMPARISONS FOR $D = P = 2$

In *Experiment 3* we compare our method with DP-GP [Guo et al., 2019] and G-HMM ($K \leq 15$). We take $K^* = 9$ true vector fields on $\mathcal{X} = (-1, 1)^2$ and we use spatially censored data for training. For our proposed iHMM-GP, we used $m_0 = 200, n_0 = 3$, $\boldsymbol{Z}$ as a uniformly spaced grid on $\mathcal{X}$ of size 100 and $(3, 2, 1)$ as the iHMM parameters. In this case, we do not use a fixed number of observations at every time, the n in the settings below indicates the mean number of observations per time. For DP-GP, the current implementation cannot accommodate more than 30 observations per time frame, hence we use the setting (***EXP 3***) $T = 600, n = 30, \sigma^2 = 1$, for which we compare the three methods. The results are shown in Fig 7. DP-GP was allowed to run for 3 Gibbs iterations and suffers from scalability and interpretability, see Table 1.

Results from using iHMM-GP with varying n and σ^2 are in Table 2. We see that prediction quality worsens for $n = 50$ as the noise level increases from $\sigma^2 = 1$ to $\sigma^2 = 3$. However, if we increase n to 120, then even at this noise level, the prediction quality is good again.

Table 2: Label prediction accuracy on test data for different settings for $D = P = 2$ simulations for iHMM-GP.

Setting		Test accuracy				time
n	σ^2	Rand	NMI	V	$d(K, K^*)$	(sec)
50	1	0.962	0.924	0.927	1.28	65.09
50	3	0.872	0.586	0.611	2.08	83.73
120	3	0.944	0.899	0.903	1.26	105.4

6 APPLICATION TO NGSIM DATASET

We chose a real-world traffic dataset collected as part of Federal Highway Administration's (FWHA) Next Generation SIMulation (NGSIM) project. The dataset contains detailed multi-vehicle trajectories at multiple intersections and freeways. The selected subset of the data was collected at Lankershim Boulevard in the Universal City neighborhood of Los Angeles, CA. Figure 1 (A) and (B) shows the spatial region under study. Each time frame is 0.1 seconds in duration and contains the locations x and velocities y of all vehicles in the spatial region under consideration at that time. We consider 8 minutes worth of data, with $T = 4800$ frames, each with a varying number of observations n_t (318,751 total observations). We applied our algorithm to extract the latent traffic patterns. For space constraints, we use the following acronyms: NB/SB/EB/WB (North/South/East/West bound), LB (Lankershim Blvd), UHD (Universal Hollywood Dr), CCW (Campo Cahuenga Wy).

We fixed $m_0 = 1200$ and $(\alpha, \beta, \gamma) = (3, 3, 3)$. The estimated noise variance was $\hat{\sigma}_1^2 \approx 0.07, \hat{\sigma}_2^2 \approx 36.26$. Since the LB is laid along the $y-$axis, most of the variation comes from that component. To select the inducing points, we collected all $\{\boldsymbol{X}_t\}_{t \in T}$ and performed a Kmeans++ to collect 400 centers. They were well spread out over the road sections. The algorithm took around **150 minutes**. A total of $\boldsymbol{K = 44}$ traffic patterns were estimated. From the estimated state labels, we computed the estimated transition matrix. Figure 8 (right) in Appendix E shows that it is sparse and carries high values along the diagonal, which suggests a high amount of self-transitions (indeed, each frame is 0.1 seconds and a pattern typically lasts longer).

Figure 8 shows 6 prominent velocity fields, each pattern is represented by the posterior mean of the SGP related to that field, on the inducing points. On closer inspection, one can see the traffic patterns captured by our model — it is important to note, in a completely unsupervised fashion, without any other spatial or temporal information. As an example, pattern 2 involves SB vehicles on the LB, going either straight through that intersection or taking a left turn towards UHD. The figure also shows the 100-step estimated transition probabilities, restricted to these states (100 steps correspond to 10 seconds). The single-step transition and 1200-step transition matrix restricted to these 6 states are shown in the heatmaps in Figure 9, the latter shows the

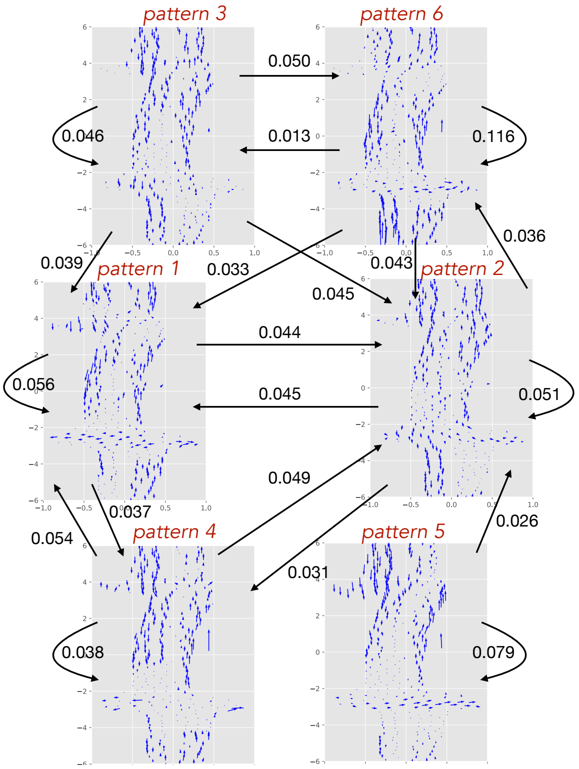


Figure 8: Prominent motion patterns and their associated 100-step (10 sec) transition probabilities.

approximate stationary behavior of the chain at these states.

To demonstrate the usefulness of such a model, we present a simple outlier detection scheme, explained in detail in Appendix E. Two examples are shown in Figure 10. Each plot is a specific time frame, the blue arrows show the estimated velocity field at that time and the red arrow is the particular vehicle in that time frame, whose velocity has a high deviation from the predicted field value at that location. Consider Eg 1 (at $t = 427$), the red vehicle is seen to be at a position where it should not be - it could be a vehicle taking a left turn toward the residential area (Valley Heart Dr), which is outside the spatial region shown. As another example, looking at Eg 2, the red arrow is very strange since it is a part of the road where traffic flows in the other direction. Upon inspection, we found that there is a bus terminal and metro station in the region to the left of that red arrow, and possibly, this vehicle was actually inside that region.

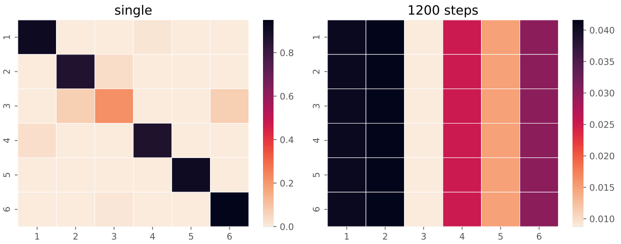


Figure 9: Estimated transition probabilities for the 6 patterns shown in Fig 8. (left) one-step transition and (right) 1200-steps transition, equivalent to 2 minutes

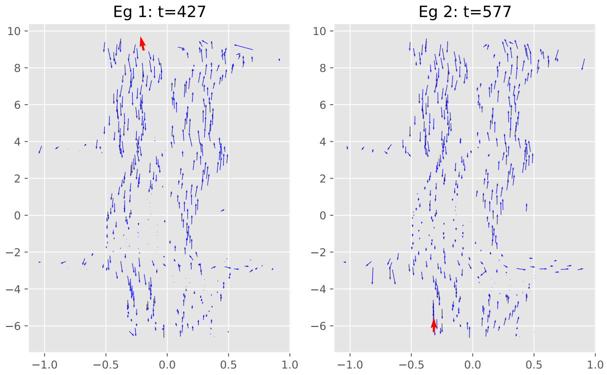


Figure 10: Examples of outlier detection.

7 CONCLUSION

Motivated by learning and understanding heterogeneous patterns in spatio-temporal dynamic data, we introduced a stochastic model for velocity fields, drawing from Bayesian nonparametric modeling elements. We developed a fast inference method for this model involving a sequential greedy estimation step combined with novel refinement post-processing, and an application of sparse Gaussian process techniques. Through an extensive simulation study, we demonstrated the effectiveness of the proposed methodology, which outperforms existing baseline methods in both accuracy and speed. We successfully applied our method to the NGSIM dataset to efficiently extract interpretable traffic patterns from the large volume of data. We demonstrated how the results can be used for outlier detection, in the context of abnormal vehicle behavior.

There are several venues that we aim to explore as part of future work. First, it would be interesting to include other covariate information, available in the dataset, like lane ID, vehicle ID, or intersection ID for each vehicle at each time. In our study, such information has been ignored. Second, the GP-induced spatial dependence result in abnormal flow patterns in certain regions at specific times — this could be addressed by considering each lane (or consecutive lanes with traffic moving in one direction) and using a mixture of GP to capture the patterns separately for each such zone. Lastly, due to the intricacies associated with traffic motion, the choice of the kernel could be studied in more detail.

Acknowledgement

This research is partially supported by the NSF grant DMS-2015361 and a research gift from Wells Fargo.

References

Rico Angell and Daniel R Sheldon. Inferring latent velocities from weather radar data using gaussian processes. *Advances in Neural Information Processing Systems*, 31, 2018.

Miguel Barão and Jorge S Marques. Gaussian random vector fields in trajectory modelling. In *Irish Machine Vision and Image Processing Conference-IMVIP*, 2017.

Matthew Beal, Zoubin Ghahramani, and Carl Rasmussen. The infinite hidden markov model. *Advances in neural information processing systems*, 14, 2001.

Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584, 2002.

D.M. Blei and M.I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2006.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res*, 3:993–1022, 2003.

Y. F. Chen, M. Liu, S.Y. Liu, J. Miller, and J. How. Predictive modeling of pedestrian motion patterns with bayesian nonparametrics. *AIAA 2016-1861*, 2016.

David Ellis, Eric Sommerlade, and Ian Reid. Modelling pedestrian trajectory patterns with gaussian processes. In *2009 IEEE 12th international conference on computer vision workshops, ICCV workshops*, pages 1229–1234. IEEE, 2009.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

Nicholas Foti, Jason Xu, Dillon Laird, and Emily B. Fox. Stochastic variational inference for hidden markov models. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2015.

E. Fox, E. Sudderth, M. I. Jordan, and A. Willsky. The sticky hdp-hmm: Bayesian nonparametric hidden Markov models with persistent states. Technical Report P-2777, MIT LIDS, 2009.

A.E. Gelfand and A.F.M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85 (410):398–409, 1990.

Yaohui Guo, Vinay Varma Kalidindi, Mansur Arief, Wenshuo Wang, Jiacheng Zhu, Huei Peng, and Ding Zhao. Modeling multi-vehicle interaction scenarios using gaussian random field. *arXiv preprint arXiv:1906.10307*, 2019.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Gustav Eje Henter, Marcus R Frean, and W Bastiaan Kleijn. Gaussian process dynamical models for nonparametric speech representation and synthesis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4505–4508. IEEE, 2012.

M. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, May 2013.

Mevin B. Hooten and Devin S. Johnson. Basis function models for animal movement. *Journal of the American Statistical Association*, 112(518):578–589, 2017.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2): 183–233, 1999.

Joshua Joseph, Finale Doshi-Velez, Albert S Huang, and Nicholas Roy. A bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4): 383, 2011.

Yohan Jung and Jinkyoo Park. Scalable hybrid hmm with gaussian process emission for sequential time-series data clustering. *arXiv preprint arXiv:2001.01917*, 2020.

Tobias Klinger, Franz Rottensteiner, and Christian Heipke. A gaussian process based multi-person interaction model. In *XXIII ISPRS Congress, Commission III 3 (2016), Nr. 3*, volume 3, pages 271–277. Göttingen: Copernicus GmbH, 2016.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18(134): 1–35, 2017.

Ian C. McDowell, Dinesh Manandhar, Christopher M. Vockley, Amy K. Schmid, Timothy E. Reddy, and Barbara E. Engelhardt. Clustering gene expression time series data using an infinite gaussian process mixture model. *bioRxiv*, 2017.

Masatoshi Nagano, Tomoaki Nakamura, Takayuki Nagai, Daichi Mochihashi, Ichiro Kobayashi, and Masahide Kaneko. Sequence pattern extraction by segmenting time series data using gp-hsmm with hierarchical dirichlet process. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4067–4074. IEEE, 2018.

Tomoaki Nakamura, Takayuki Nagai, Daichi Mochihashi, Ichiro Kobayashi, Hideki Asoh, and Masahide Kaneko. Segmenting continuous motions with hidden semi-markov models and gaussian processes. *Frontiers in neurorobotics*, 11:67, 2017.

Patrick Nelson and Lawrence M Widrow. Gaussian process model for the local stellar velocity field from gaia data release 2. *Monthly Notices of the Royal Astronomical Society*, 516(4):5429–5439, 2022.

C.E. Rasmussen and C.K.I Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

Dustin R Rubenstein and Keith A Hobson. From birds to butterflies: animal movement patterns and stable isotopes. *Trends in ecology & evolution*, 19(5):256–263, 2004.

Dripta Sarkar, Michael Osborne, and Thomas Adcock. Spatiotemporal prediction of tidal currents using gaussian processes. *Journal of Geophysical Research: Oceans*, 124, 04 2019. doi: 10.1029/2018JC014471.

Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 101: 1566–1581, 2006.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

Nguyen Xuan Vinh and Julien Epps. Bailey, j2738784: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. vol. 11. *J Mach Learn Res*, pages 2837–2854, 2010.