Online Student-t Processes with an Overall-local Scale Structure for Modelling Non-stationary Data

Taole Sha
University of Hong Kong
u3577089@connect.hku.hk

Michael Minyi Zhang
University of Hong Kong
mzhang18@hku.hk

Abstract

Time-dependent data often exhibit characteristics, such as non-stationarity and heavy-tailed errors, that would be inappropriate to model with the typical assumptions used in popular models. Thus, more flexible approaches are required to be able to accommodate such issues. To this end, we propose a Bayesian mixture of student-t processes with an overall-local scale structure for the covariance. Moreover, we use a sequential Monte Carlo (SMC) sampler in order to perform online inference as data arrive in real-time. We demonstrate the superiority of our proposed approach compared to typical Gaussian process-based models on real-world data sets in order to prove the necessity of using mixtures of student-t processes.

1 Introduction

In modelling dynamical systems, it is common that the data will exhibit non-stationarity, where the trend changes across the input space. Kernel methods like the Gaussian process (GP) are a popular choice of prior distribution over real-valued functions in Bayesian models of time series data (Rasmussen and Williams, 2005). However, in the non-stationary time series setting that this paper focuses on, they face several challenges: 1.) The calculation of the likelihood in GP inference requires inverting an $N \times N$ matrix, which generally incurs a computational complexity of $\mathcal{O}(N^3)$ where N is the number of observations; 2.) Updating the model in real-time is not trivial; 3.) Stationarity is often assumed by covariance kernels, while non-stationary kernels typically lead to computationally intractable GPs, especially when the sample size is large. As a related stochastic process, the student-t process (TP) is an attractive alternative prior distribution over function space compared to the GP which has heavy tails controlled by the degree of freedom parameter, allowing more modelling flexibility (Shah et al., 2014).

Similar to the GP, the TP has consistent marginals and closed-form conditionals which make it as convenient as the GP to use in statistical modelling without any additional computational cost. However, TPs are still liable to suffer from the aforementioned three issues that GPs face when modelling real-world data. *Hence, we introduce in this paper a mixture of TPs with an SMC sampler, so that we may take advantage of the additional flexibility of a mixture-of-experts model with a convenient online inference algorithm.* To derive the TP, we assume a latent GP and integrate out an inverse gamma prior on the kernel amplitude and the noise parameters. Moreover, we add an additional parameter of the noise term to control the heteroscedasticity. Lastly, we model the level of heavy-tailedness by automatically controlling the TPs' degree of freedom using an efficient slice sampling scheme.

Our paper proceeds as follows: In Section 2, we discuss some previous work about online GP models. The online TP inference algorithm is detailed in Section 3. We use the experiment results to compare it with GP-based models in Section 4. Finally, we conclude the paper in Section 5 with a discussion of future work.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Related Work

While the GP is a convenient choice of prior as it leads to tractable posterior inference in many classes of models, GPs suffer from the typical cubic computational that other kernel methods face. Numerous scalable methods have been developed to tackle the computational issue of GPs: Sparse inducing point methods are a popular technique for reducing the computational complexity of GP methods (Snelson and Ghahramani, 2006; Titsias, 2009; Bauer et al., 2016). In the sparse GP methods, they form a low-rank approximation of the kernel function using a collection of M "pseudo-inputs" which reduce the computational complexity of the GP to $O(NM^2)$ from $O(N^3)$.

Product-of-expert models employ a block diagonal approximation of the full covariance matrix in order to reduce the complexity of the full covariance matrix inversion to individually inverting each smaller block (Deisenroth and Ng, 2015; Cohen et al., 2020). While not necessarily faster, mixture-of-expert models use a mixture of GPs to model functions with greater flexibility compared to a single GP (Rasmussen and Ghahramani, 2001; Meeds and Osindero, 2005).

For fast online GP methods, Csató and Opper (2002) used variational inference to approximate the posterior in a sparse online GP model, however, the hyperparameters are assumed to be fixed in their method. Nguyen-tuong et al. (2008) proposed a product-of-experts local GP method for online fitting, where the weights are based on the distance of the new observation to the local models. Though, in such methods, ignoring the correlation between experts when adopting the local assumption can lead to poor uncertainty quantification. Bui et al. (2017) developed a sparse variational GP regression approach that allows for online updating of the hyperparameters, called OSVGP. However, OSVGP has a tendency to be numerically unstable and, empirically, is liable to underfit the data. Stanton et al. (2021) developed an exact sparse online model called WISKI, where a structured and sparse covariance matrix approximation developed by Wilson and Nickisch (2015) is used, leading to constant computational complexity with respect to the number of observations.

Regarding SMC methods in GPs, Svensson et al. (2015) proposed an SMC sampler with the purpose of marginalizing the kernel hyperparameters and Gramacy and Polson (2011) proposed an SMC sampler for sequential design in GPs. While these SMC methods allow for updating the GP model sequentially, they cannot account for non-stationarity in the data, nor are they able to limit the computational cost of the model as the complexity still scales $O(N^3)$. However, Zhang and Williamson (2019) proposed an importance sampling method for scaling up a mixture-of-experts GP model to an average complexity of $O(N^3/K^2)$ for non-stationary data. Later, Zhang et al. (2023) and Härkönen et al. (2022) developed an online SMC and SMC^2 sampler for mixture of GPs. But despite the advances in online mixtures of GPs, little attention has been paid to online mixtures of the student-t process.

3 Online Student-t Processes For Non-stationary Data

The data generating process for our proposed model is:

$$
\begin{aligned}
&\mathbf{x}_i \sim \mathcal{T}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Psi}_{z_i}, \nu_{z_i}), \;\; \alpha \sim \text{Gamma}(a_0, b_0), \;\; z_i|\alpha \sim \text{CRP}(\alpha), \\
&\boldsymbol{\theta}_k \sim log\mathcal{N}(m_0, s_0^2\mathbf{I}), \;\; \nu_k \sim \text{Gamma}(2, 0.1) \;\; h_k \sim \mathcal{N}(0, k_0^2), \;\; k_0^2 \sim \text{Inv-Gamma}\left(\frac{1}{2}, \frac{1}{2}\right), \\
&\mathbf{y}_k|\mathbf{X}_k, \boldsymbol{\theta}_k, \sim \mathcal{T}(\nu_k, 0, \mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}).
\end{aligned} \tag{1}
$$

where the i-th input $\mathbf{x}_i$ comes from an infinite Dirichlet process Gaussian-inverse Wishart mixture model (Antoniak, 1974): $\mathcal{N}(\mathbf{M}_{z_i}, \mathbf{C}_{z_i})$. The latent parameters $(\mathbf{M}_{z_i}, \mathbf{C}_{z_i})$, for $z_i \in \{1, 2, \ldots\}$, are integrated out over a normal-inverse Wishart prior, $\mathcal{NIW}(\boldsymbol{\mu}_{z_i}, \lambda_{z_i}, \boldsymbol{\Psi}_{z_i}, \nu_{z_i})$. The x_i marginally follows a student-t distribution $\mathcal{T}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Psi}_{z_i}, \nu_{z_i})$.

Since student-t distributions are not closed under addition, we cannot analytically obtain a latent TP and independent student-t noise for modelling regression problems. Instead, a previous solution is incorporating the noise into the kernel function. According to Shah et al. (2014), Zhang and Yeung (2010) wrongly assumed the noise to be independent when raising this model. Shah et al. (2014) stated that the noise term is uncorrelated but dependent, and argued it to behave similarly to a sum of a latent TP with independent noise.

However, directly incorporating the noise term into the kernel may not be sufficiently flexible for modelling real data. In our model, the outputs y_i from cluster k are denoted as $\mathbf{y}_k$, which we obtain:

$$\sigma_k^2|\nu_k \sim \text{Inv-Gamma}\left(\frac{\nu_k}{2}, \frac{\nu_k}{2}\right), \;\; \mathbf{y}_k|\sigma_k^2, \mathbf{X}_k \sim \mathcal{N}(0, \sigma_k^2(\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I})),$$
$$\int P(\mathbf{y}_k|\sigma_k^2, -)P(\sigma_k^2)d\sigma_k^2 \sim T(\nu_k, 0, \mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}). \tag{2}$$

It is assumed to be a sum of a GP and dependent Gaussian noise. The covariance is determined by an overall scale parameter σ_k^2 for both the kernel and the noise, a local scale parameter h_k for the noise only to control this heteroscedasticity, and kernel parameters $\boldsymbol{\theta}_k$. The overall scale parameter σ_k^2 is integrated out over an inverse gamma prior, and a TP can be derived. When the i^{th} streaming data $(\mathbf{x}_i, y_i)$ comes, we assign it to cluster k according to the predictive distribution of the DP, the Chinese restaurant process (Aldous, 1985):

$$P(z_i = k|\alpha, \mathbf{X}_k) \propto \begin{cases} N_k' \cdot \mathcal{T}(\boldsymbol{\mu}_k', \boldsymbol{\Psi}_k', \nu_k') & k \in K^+. \\ \alpha \cdot \mathcal{T}(\boldsymbol{\mu}_0, \boldsymbol{\Psi}_0, \nu_0) & \text{o.w.} \end{cases} \tag{3}$$

K^+ refers to the existing clusters, and all $(\cdot)'$ represent summary statistics calculated with first $i-1$ observations. The student-t likelihood's parameters $(\mu_k^{'}, \boldsymbol{\Psi}_k^{'}, \nu_k^{'})$ of inputs can be updated by:

$$\begin{aligned}
\boldsymbol{\mu}_k' =& \frac{\lambda_0\boldsymbol{\mu}_0 + N_k'\bar{\mathbf{x}}_k}{\lambda_k'}, \;\; \nu_k' = \nu_0 + N_k' - D + 1, \;\; \boldsymbol{\Psi}_k' = \frac{\lambda_k' + 1}{\lambda_k'\nu_k'}\left(\boldsymbol{\Psi}_0 + \mathbf{S}_k' + \mathbf{S}_{\bar{\mathbf{x}}_k}'\right), \\
\bar{\mathbf{x}}_k' =& \frac{\sum_{i':(z_{i'}=k, i'<i)} \mathbf{x}_{i'}}{N_k'}, \;\; N_k' = \sum_{i'=1}^{i-1} I(z_{i'} = k), \;\; \lambda_k' = \lambda_0 + N_k', \\
\mathbf{S}_k' =& \sum_{i':(z_{i'}=k, i'<i)} (\mathbf{x}_{i'} - \bar{\mathbf{x}}_k')(\mathbf{x}_{i'} - \bar{\mathbf{x}}_k')^T, \;\; \mathbf{S}_{\bar{\mathbf{x}}_k}' = \frac{\lambda_0 N_k'}{\lambda_k'}(\bar{\mathbf{x}}_k' - \boldsymbol{\mu}_0)(\bar{\mathbf{x}}_k' - \boldsymbol{\mu}_0)^T.
\end{aligned} \tag{4}$$

Also, a Gamma prior is placed on the Dirichlet process concentration parameter α. We can use a variable augmentation scheme to sample its full conditional posterior up to observation i (Escobar and West, 1995).

$$\begin{aligned}
\rho|\alpha \sim& \text{Beta}(\alpha + 1, i), \;\; K = |\{k : N_k > 0\}|, \;\; \frac{\pi_\alpha}{1 - \pi_\alpha} = \frac{a_0 + K - 1}{N(b_0 - \log\rho)}, \\
\alpha|\mathbf{z}_{1:i}, \pi_\alpha, \rho =& (1 - \pi_\alpha) \cdot \text{Gamma}(\alpha_0 + K - 1, b_0 - \log\rho) + \pi_\alpha \cdot \text{Gamma}(\alpha_0 + K, b_0 - \log\rho).
\end{aligned} \tag{5}$$

The Gamma$(2, 0.1)$ prior is commonly used when inferring the degree of freedom, which puts mass on a large range of reasonable values for the degrees of freedom (Juárez and Steel, 2010). We sample the degrees of freedom parameter through an efficient variable augmentation scheme. Given the latent overall scale σ_k^2, the degree of freedom ν_k will be independent of all other parameters and data. Due to the conjugacy between the Gaussian likelihood and the inverse Gamma prior, we can directly Gibbs sample the σ_k^2 from its full conditional. Then, conditioned on σ_k^2, we sample ν_k using the slice sampler from $P(\nu_k|\sigma_k^2)$ (Neal, 2003; Damien et al., 1999).

We assume a hierarchical structure on the local heteroscedasticity parameter, $|h_k|$, where global scale k_0^2 is shared over all mixtures. Here, we will share scale data from other clusters to inform the posterior sampling of h_k. Because the likelihood is Gaussian and the prior over k_0^2 is an inverse Gamma, we can again sample the full conditional of k_0^2 in closed form. And we sample h_k and the TP parameters $\boldsymbol{\theta}_k$ using the elliptical slice sampler (ESS), which is an efficient sampling algorithm for non-conjugate models with Gaussian priors (Murray et al., 2010).

3.1 SMC for Online TP-MOE

In our proposed method, we use a sequential Monte Carlo sampler in order to update the model as new data arrive (Del Moral et al., 2006). SMC follows from importance sampling (IS) and sequential importance sampling (SIS) algorithms in Monte Carlo methods, and can avoid the degeneracy problem the two can encounter, where one proposal weight $w^{(j)}$ dominates the rest of the proposals.

For $j = 1, \ldots, J$ particles, the particles $(\mathbf{z}^{(j)}, \boldsymbol{\theta}^{(j)}, \mathbf{h}^{(j)}, \alpha^{(j)})$ are updated as described before when a new observation arrives. Then, we calculate the particle weights, which results in a posterior weighted sample TP product-of-experts models. Initially when $i = 1$, the particle j's weight is:

$$w_1^{(j)} \propto P(y_1|z_1^{(j)}, \mathbf{x}_1, \boldsymbol{\theta}^{(j)}, h^{(j)}, \nu^{(j)}) P(\mathbf{x}_1|z_1^{(j)}, \alpha^{(j)}). \tag{6}$$

Then the updating procedure for $i > 1$ is shown in Algorithm 1.

Algorithm 1: SMC Sampler for TP-MOE

Input: New observation $(\mathbf{x}_i, y_i)$
for $j = 1, \cdots, J$ *in parallel* **do**
  Sample $z_i^{(j)} = k$ from $P(z_i^{(j)}|\alpha^{(j)}, \mathbf{X}_{1:i-1})$
  Sample $\alpha^{(j)}$ from the full conditional $P(\alpha^{(j)}|\mathbf{z}_{1:i})$
  Sample $\theta_k^{(j)}$ and $h_k^{(j)}$ jointly by using the elliptical slice sampler
  Sample $(k_0^2)^{(j)}$ from $P((k_0^2)^{(j)}|h_1^{(j)}, \ldots, h_K^{(j)})$
  Sample $(\sigma_k^2)^{(j)}$ from $P(\sigma_k^2|\mathbf{X}_k, \mathbf{y}_k, \nu_k^{(j)})$
  Sample $\nu_k^{(j)}$ by using the slice sampler
  Update particle weight:

$$w_i^{(j)} = w_{i-1}^{(j)} P(\mathbf{x}_i|\alpha^{(j)}, z_i^{(j)}) \times \frac{P\left(\mathbf{y}_{1:i}|\mathbf{X}_{1:i}, \theta_{k,i}^{(j)}, h_k^{(j)}, \nu_k^{(j)}\right)}{P\left(\mathbf{y}_{1:i-1}|\mathbf{X}_{1:i-1}, \theta_k'^{(j)}, h_k'^{(j)}, \nu_k'^{(j)}\right)} \tag{7}$$

end
Normalize weights:

$$w_i^{(j)} := \frac{w_i^{(j)}}{\sum_{j=1}^J w_i^{(j)}}$$

if $N_{eff} < \frac{J}{2}$ **then**
  Resample particles $(\mathbf{z}_{1:i}^{(\mathbf{j}^*)}, \theta_k^{(\mathbf{j}^*)}, h_k^{(\mathbf{j}^*)}, \nu_k^{(\mathbf{j}^*)}, \alpha^{(\mathbf{j}^*)})$, where
   $\mathbf{j}^* \sim$ Multinomial$(J, w_i^{(1)}, \cdots, w_i^{(j)})$
  Set $w_i^{(j)} := \frac{1}{J}$ for $j = 1, \cdots, J$
end
Output: Particle weights $(w_i^{(1)}, \cdots, w_i^{(j)})$ and particles $(\mathbf{z}_{1:i}^{(1:J)}, \theta^{(1:J)}, \mathbf{h}^{(1:J)}, \nu^{(1:J)}, \alpha^{(1:J)})$

The computational complexity is dominated by the inversion of a $N_k \times N_k$ matrix. If we assume that the average size of N_k is N/K, the number of data divided by the number of clusters, the n the computational complexity will be $\mathcal{O}(JN^3/K^2)$. Under the basic setting of our sampler, the complexity of the sampler still grows as new data arrive so the method cannot truly be considered "online". To this end, we adopt the "minibatched" stochastic approximation that is widely used as a method for substantially reducing the computational complexity of posterior inference (Zhang et al., 2023; Zhang and Williamson, 2019; Minsker et al., 2014; Srivastava et al., 2015). A subsample of size B from the mixture with N_k observations is drawn uniformly without replacement, then their likelihood is calculated and upweighted by N_k/B power to approximate the full likelihood. The stochastic approximation method leads us to:

$$\begin{aligned}
&\mathbf{u}_k^{(j)} = (u_1, \ldots, u_B) \sim \text{HyperGeometric}\big(B, \big\{i : z_i^{(j)} = k\big\}\big), \\
&(\mathbf{y}_{\mathbf{u}_k}, \mathbf{X}_{\mathbf{u}_k}) = \big(y_u, \mathbf{x}_u : u \in \mathbf{u}_k^{(j)}\big). \\
&P\big(\mathbf{y}_{\mathbf{u}_k^{(j)}}|\mathbf{X}_{\mathbf{u}_k^{(j)}}, \boldsymbol{\theta}_k^{(j)}, h_k, \sigma_k^2\big) \sim \mathcal{N}\left(0, \sigma_k^2(\mathbf{K}_{\boldsymbol{\theta}_k^{(j)}} + \frac{N_k|h_k|}{B}\mathbf{I})\right), \\
&P\big(\mathbf{y}_{\mathbf{u}_k^{(j)}}|\mathbf{X}_{\mathbf{u}_k^{(j)}}, \boldsymbol{\theta}_k^{(j)}, h_k, \nu_k\big) \sim \mathcal{T}\left(\nu, 0, \mathbf{K}_{\boldsymbol{\theta}_k^{(j)}} + \frac{N_k|h_k|}{B}\mathbf{I}\right).
\end{aligned} \tag{8}$$

With minibatching, the complexity is reduced to $\mathcal{O}(J\min\{N_k, B\}^3/K^2)$. As each particle can be updated independently, the parallel computation can be adopted to further reduce the complexity to $\mathcal{O}(\min\{N_k, B\}^3/K^2)$. Then, we calculate the effective sample size, $N_{eff} = 1/\sum_{j=1}^{J}(w_i^{(j)})^2$, based on the particle weights. If it is lower than a threshold set by the user, typically $J/2$, the particles are resampled to only preserve the high-weighted ones and avoid the degeneracy problem. For a future output, a particle predicts it by combining results from mixtures within it weighted by the CRP probabilities of it belonging to each mixture. And we weighted average predictions from all particles according to their weights.

4 Experiments

In this section, we proceed to study the advantages of the heavy tails by implementing the TP-MOE and other Gaussian-based models on different non-stationary datasets and analysing their performances in terms of one-step-ahead predictions. The GP models include a Gaussian mixture-of-experts model (GP-MOE) (Zhang et al., 2023), a sparse online GP method using the Woodbury identity and structured kernel interpolation (WISKI) (Stanton et al., 2021), and an online sparse variational GP method (OSVGP) (Bui et al., 2017) [1].

For the experiments, we sequentially predict the next future observation and update the model with the real data point. The one-step predictive mean squared error (MSE) is adopted to evaluate the results. The 3 datasets used include: 1.) An accelerometer measurement of a motorcycle crash (N=94). 2.) The price of Brent crude oil (N=100). 3.) The annual carbon dioxide output in Canada (N=215). [2]. They exhibit non-stationarity in both length-scale and noise, and have been pre-processed to have zero mean and unit variance.

To make the results comparable, The TP-MOE and the GP-MOE share the same particle number $J = 100$ and the same 16 cores used on a shared memory process based on OpenMP, and the number of inducing points for all models is set to be 50. The OSVGP's number of optimization iterations is set to the default value of 1. The radial basis function kernel for all models is:

$$\Sigma(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\theta}{2}\sum_{d=1}^{D}(x_d - x_d')^2\right\} \tag{9}$$

The plots of four algorithms' sample runs are shown in Figure 1-3, which contain data points, one-step predictive mean (plotted with solid red lines) and 95% predictive interval (plotted with dashed black lines). The data points in TP-MOE's and GP-MOE's plots are coloured according to the cluster assignment given by the particle with the highest weight. The results in terms of the predictive MSE are listed in Table 1.

Table 1: One-step Predictive MSE. One Standard Error Reported in Parentheses

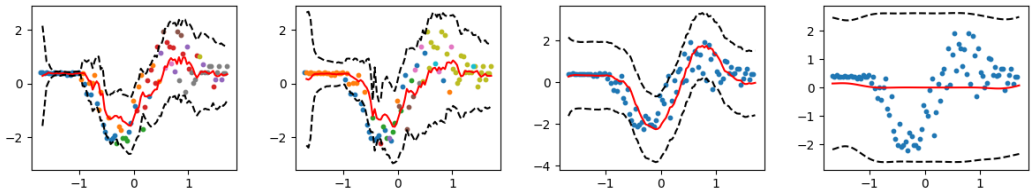
	Motorcycle	Brent	Canada
TP-MOE	**0.363 (0.028)**	**0.146 (0.014)**	**0.015 (0.003)**
GP-MOE	0.381 (0.038)	0.160 (0.019)	0.016 (0.004)
WISKI	0.631 (0.000)	0.220 (0.000)	0.048 (0.000)
OSVGP	0.998 (0.002)	0.782 (0.021)	0.711 (0.030)

From the comparisons in Table 1, we observe that our TP-MOE performs better than the GP-based models and achieves lower predictive MSE. According to the plots of sample runs (Figure 1-3),we can see that the MOE models can better capture the heterogeneity of the underlying function better than the stationary models. Moreover, we can see that the TP-MOE produces tighter predictive credible intervals compared to the GP-MOE, which sometimes produces overly conservative predictive intervals. This suggests that the TP-MOE has better uncertainty quantification capabilities. The

[1]The implementation for GP-MOE is available at `https://github.com/michaelzhang01/GPMOE`. The code of OSVGP and WISKI are available at: `https://github.com/wjmaddox/online_gp`. Our code is at `https://github.com/stllllll/TP-MOE`

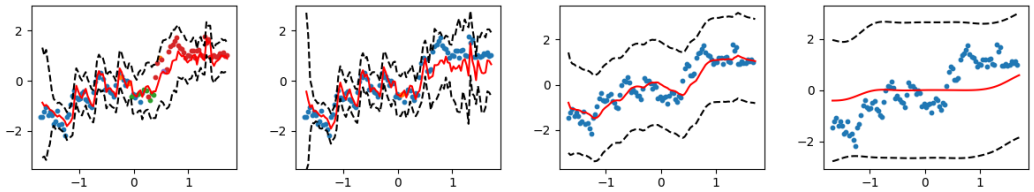
[2]The motorcycle dataset can be found in the R package `VarReg`. The Brent, Canada CO_2, datasets are available at: `https://github.com/alanturing-institute/TCPD`.

OSVGP tends to underfit as expected, while the WISKI cannot quantify the uncertainty as well as the mixture-of-experts models due to its assumption of stationarity.



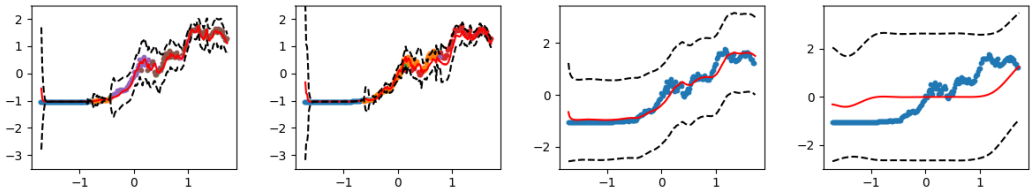
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 1: Sample Runs on the Motorcycle Dataset. N = 94.



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 2: Sample Runs on the Brent Dataset. N = 100.



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 3: Sample Runs on the Canada Dataset. N = 215.

5 Conclusion

Heavy-tailed data sets appear in a wide variety of applied settings. However, devising models that can adequately handle their noise structure is not trivial. In this paper, we build a Bayesian mixture of student-t processes model with an overall-local scale structure for noisy data, which can be inferred by an SMC online algorithm. We have shown that TP-MOE has advantages over the Gassian-based models when facing commonly encountered non-stationary data.

In future work, we are interested in applying the TP-MOE in optimization and reinforcement learning tasks. For such tasks, the learning, prediction, and decision making aspects of the model occur in sparse, noisy environments that require heavy-tailed models in order for a learning agent to properly handle the problem at hand. Modelling data with a mixture of Student-t processes is a natural method for dealing with non-stationarity and heavy-tailed errors yet their popularity has still eluded the machine learning community. We seek to fill that gap with the method proposed in this paper.

References

Aldous, D. (1985). Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, pages 1152–1174.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse Gaussian process approximations. *Advances in Neural Information Processing Systems*, 29.

Bui, T. D., Nguyen, C., and Turner, R. E. (2017). Streaming sparse gaussian process approximations. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Cohen, S., Mbuvha, R., Marwala, T., and Deisenroth, M. (2020). Healing products of Gaussian process experts. In *International Conference on Machine Learning*, pages 2068–2077. PMLR.

Csató, L. and Opper, M. (2002). Sparse On-Line Gaussian Processes. *Neural Computation*, 14(3):641–668.

Damien, P., Wakefield, J., and Walker, S. (1999). Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):331–344.

Deisenroth, M. and Ng, J. W. (2015). Distributed Gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490. PMLR.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Gramacy, R. B. and Polson, N. G. (2011). Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1):102–118.

Härkönen, T., Wade, S., Law, K., and Roininen, L. (2022). Mixtures of Gaussian process experts with SMC^2. *arXiv preprint arXiv:2208.12830*.

Juárez, M. A. and Steel, M. F. J. (2010). Model-based clustering of non-gaussian panel data based on skew-t distributions. *Journal of Business & Economic Statistics*, 28(1):52–66.

Meeds, E. and Osindero, S. (2005). An alternative infinite mixture of Gaussian process experts. *Advances in Neural Information Processing Systems*, 18.

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. (2014). Scalable and robust bayesian inference via the median posterior. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1656–1664, Bejing, China. PMLR.

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548, Chia Laguna Resort, Sardinia, Italy. PMLR.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics*, 31(3):705–767.

Nguyen-tuong, D., Peters, J., and Seeger, M. (2008). Local gaussian process regression for real time online model learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Rasmussen, C. and Ghahramani, Z. (2001). Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 14.

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

Shah, A., Wilson, A., and Ghahramani, Z. (2014). Student-t Processes as Alternatives to Gaussian Processes. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 877–885, Reykjavik, Iceland. PMLR.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1259–1266.

Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 912–920, San Diego, California, USA. PMLR.

Stanton, S., Maddox, W., Delbridge, I., and Gordon Wilson, A. (2021). Kernel interpolation for scalable online gaussian processes. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3133–3141. PMLR.

Svensson, A., Dahlin, J., and Schön, T. B. (2015). Marginalizing Gaussian process hyperparameters using sequential Monte Carlo. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 477–480. IEEE.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574. PMLR.

Wilson, A. and Nickisch, H. (2015). Kernel interpolation for scalable structured gaussian processes (kiss-gp). In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1775–1784, Lille, France. PMLR.

Zhang, M. M., Dumitrascu, B., Williamson, S. A., and Engelhardt, B. E. (2023). Sequential gaussian processes for online learning of nonstationary functions. *IEEE Transactions on Signal Processing*, 71:1539–1550.

Zhang, M. M. and Williamson, S. A. (2019). Embarrassingly parallel inference for gaussian processes. *Journal of Machine Learning Research*, 20(169):1–26.

Zhang, Y. and Yeung, D. (2010). Multi-task learning using generalized t process. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 964–971, Chia Laguna Resort, Sardinia, Italy. PMLR.