Multiscale Gaussian Process Regression

Abhinav Gupta [1]

Abstract

In this project, we compare the abilities of ARD kernel, Gibbs kernel, additive GPR and a weighted mixture of GPs method in handling multiscale features using synthetic data. Application to real data is shown for some time-series data of fish-catch, and sea-surface-temperature image marred due to cloud cover.

1. Introduction

Gaussian Processes (GPs) provide a very general and flexible framework for defining a distribution over functions, while respecting properties such as smoothness by using appropriate covariance kernels. The relevant data might have spatially varying length-scales, noise, or involve correlations spanning multiple scales. Thus it becomes very important to ensure that the prior model —a GP with appropriate form of mean and covariance functions— has the ability to represent the observed data. In this project, I focus on modeling data which exhibit multiple correlation length-scales (multiscale) using different Gaussian Process Regression (GPR) models. Such data is relevant to physical processes where a cascade of interactions happens at different scales; for example, ocean flows where micro-turbulence happens at $\mathcal{O}$(1cm) to geostrophic eddies at $\mathcal{O}$(1000km) (Cushman-Roisin & Beckers, 2011).

The literature on GPs is rich, and there exist methods to handle most conceivable data features. For multiscale data, common techniques involve use of ARD (Automatic Relevance Determination) kernel, non-stationary kernels (Williams & Rasmussen, 2006), additive models (Duvenaud et al., 2011), hierarchical models (Fox & Dunson, 2012), mixture of experts (Rasmussen & Ghahramani, 2002), and clustering (Zhang et al., 2015), among others. Inspired by the Gaussian Mixture Model (GMM) based Kalman filters (Alspach & Sorenson, 1972; Sondergaard & Lermusiaux, 2013a;b), I start with a weighted mixture of GPs for the prior, and then derived a closed-form solution for the posterior weights as well as for the mean and covariance function for each mixture model, which I will refer to as the *weighted mixture of GPs* method. In this project, I compare the performance of some of these methods GP — ARD kernel, Gibbs kernel (Gibbs, 1998), additive model and weighted mixture of GPs— using synthetic 1-dimensional and 2-dimensional data. I compare computational costs, and lastly, perform regression using actual data for daily fish catch in the Lakshadweep islands of India and a sea-surface-temperature (SST) satellite measurement near the east coast of the US.

[*]Equal contribution [1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA. Correspondence to: Abhinav Gupta <guptaa@mit.edu>.

Final project report for the course: 6.435 Bayesian modeling and inference, Spring 2019, MIT.

2. Methodology

In the subsequent sub-sections, I will discuss and review the different GP forms/kernels being used in this project, starting from a review of standard GPR in order to form the base for terminologies and notations.

2.1. Gaussian Process Regression

Let us say, we have a observational dataset $\mathcal{D}$ consisting of N input vectors $\mathbf{X} = \{\mathbf{x}\}_{n=1}^{N}$ of dimension D and corresponding real-valued targets $\mathbf{y} = \{y\}_{n=1}^{N}$. Assume that this data is coming from some underlying function $f(\mathbf{x})$, according to the observation model $y = f(\mathbf{x}) + \varepsilon$, where ε is some form of additive noise. The overall goal of GPR is to infer the function $f(\mathbf{x})$ based on the information provided by the observed dataset. For this, we assume a Gaussian process prior for $f(\mathbf{x}) \sim \mathcal{N}(f(\mathbf{x})|m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, which could be interpreted as a multivariate Gaussian distribution for any finite set of locations $\{\mathbf{X}, \mathbf{f}\}$, $p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|m(\mathbf{X}), \mathbf{K_{XX}})$, where $m(\mathbf{X})$ is a mean function and $\mathbf{K_{XX}}$ a covariance matrix constructed using a kernel: $[\mathbf{K}]_{nn'} = k(\mathbf{x}_n, \mathbf{x}_{n'})$. We also assume that the noise in our observation model is independent Gaussian, given by $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. Together the parameters defining the mean function, kernel and noise are called hyperparameters, and denoted with $\boldsymbol{\theta}$. Based on the observation mode, we can write marginal likelihood of the observed data as $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|m(\mathbf{X}), \mathbf{K_{XX}} + \sigma_n^2\mathbf{I})$. Using Bayes's rule, we can find the posterior function ($\mathbf{f}_*$) at some test locations ($\mathbf{X}_*$), given by:

$$p(\mathbf{f}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*) = \mathcal{N}(\mathbf{f}_*|\bar{\mathbf{f}_*}, \mathrm{cov}(\mathbf{f}_*)), \tag{1}$$

where $\bar{\mathbf{f}_*} = m(\mathbf{X}_*) + \mathbf{K}_{\mathbf{X}_*\mathbf{X}}(\mathbf{K_{XX}} + \sigma_n^2\mathbf{I})^{-1}(\mathbf{y} - m(\mathbf{X}))$, and $\mathrm{cov}(\mathbf{f}_*) = \mathbf{K}_{\mathbf{X}_*\mathbf{X}_*} - \mathbf{K}_{\mathbf{X}_*\mathbf{X}}(\mathbf{K_{XX}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{XX}_*}$. To find optimal values for hyperparameters $\boldsymbol{\theta}$, we minimize the negative log marginal likelihood, i.e., $-\log(p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})) = -\log(\mathcal{N}(\mathbf{y}|m(\mathbf{X}), \mathbf{K_{XX}} + \sigma_n^2\mathbf{I})) = \frac{1}{2}(\mathbf{y} - m(\mathbf{X}))^T\mathbf{R}_{\mathbf{XX}}^{-1}(\mathbf{y} - m(\mathbf{X})) + \frac{1}{2}\log|\mathbf{R_{XX}}| + \frac{N}{2}\log(2\pi)$, where $\mathbf{R_{XX}} = \mathbf{K_{XX}} + \sigma_n^2\mathbf{I}$.

2.2. ARD Kernel (Williams & Rasmussen, 2006)

This is a variant of one of the most commonly used kernel function, squared exponential $\exp\left(-\frac{(x-x')^2}{2l^2}\right)$, with the ability to specify different length scales (l) in different dimensions. It is given by:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{2l_d^2}\right), \tag{2}$$

where l_d (scalar constant; thus a stationary kernel) is the characteristic length-scale in dimension d.

2.3. Gibbs Kernel (Gibbs, 1998)

This is a non-stationary kernel; the length scale is allowed to vary with spatial locations. There are many options to construct such kernels, each with problem-dependent pros and cons. We will use

$$k(\mathbf{x},\mathbf{x}') = \prod_{d=1}^{D}\left(\frac{2l_d(\mathbf{x})l_d(\mathbf{x}')}{l_d^2(\mathbf{x})+l_d^2(\mathbf{x}')}\right)^{1/2} \times \exp\left(-\sum_{d=1}^{D}\frac{(x_d-x_d')^2}{l_d^2(\mathbf{x})+l_d^2(\mathbf{x}')}\right), \quad (3)$$

as mentioned in (Williams & Rasmussen, 2006) (originally from (Gibbs, 1998)), where each $l_d(\mathbf{x})$ is an arbitrary positive function of $\mathbf{x}$.

2.4. Additive Gaussian Processes (Duvenaud et al., 2011)

This method relies on the fact that complex kernels can be created by simple operations that combine simpler kernels, especially by addition of legitimate kernels. Using squared exponential as base kernel, $k_d(x_d,x_d') = \exp\left(-\frac{(x_d-x_d')^2}{2l_d^2}\right)$, for each dimension, we can define a n^{th} order additive kernel as:

$$k_{add_n}(\mathbf{x},\mathbf{x}') = \sigma_{f_n}^2 \sum_{1\le i_1<\ldots<i_n\le D}\left[\prod_{d=1}^{n} k_{i_d}(x_{i_d},x_{i_d}')\right], \quad (4)$$

where D is the dimension of the input space, and $\sigma_{f_n}^2$ is the variance assigned to all the n^{th} order interactions. The n^{th} order covariance term is a sum of $\binom{D}{n}$ terms.

2.5. Weighted Mixture of Gaussian Processes

This method is inspired by GMM based Kalman filters; as such, we can think of the prior as a weighted superposition of K number of GPs with different means and covariance kernels. Thus the prior is given by:

$$f(\mathbf{x}) \sim \sum_{i=1}^{K} w_i\mathcal{N}(m_i(\mathbf{x}),k_i(\mathbf{x},\mathbf{x}')), \quad (5)$$

where $m_i(\mathbf{x})$ and $k_i(\mathbf{x},\mathbf{x}')$ are the mean and covariance kernels (which I consider to be ARD) of the i^{th} member GP, and weights sum to 1 ($=\sum_{i=1}^{K} w_i$). We can also rewrite the above prior as a single multivariate Gaussian distribution by appropriately modifying the mean and the covariance kernel, $f(\mathbf{x}) \sim \mathcal{N}(\sum_{i=1}^{K} w_i m_i(\mathbf{x}), \sum_{i=1}^{K} w_i^2 k_i(\mathbf{x},\mathbf{x}'))$. To predict f at X_*, we can again use Bayes's law to write

$$p(\mathbf{f}_*|\mathbf{y},\mathbf{X},\mathbf{X}_*,\boldsymbol{\theta}) = \frac{p(\mathbf{f}_*,\mathbf{y}|\mathbf{X},\mathbf{X}_*,\boldsymbol{\theta})}{P(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})} = \frac{\sum_{i=1}^{K} w_i\mathcal{N}\left(\begin{bmatrix}\mathbf{f}_*\\ \mathbf{y}\end{bmatrix}\middle|\begin{bmatrix}m_i(X_*)\\ m_i(X)\end{bmatrix},\begin{bmatrix}\mathbf{K}_{i_{\mathbf{X}_*,\mathbf{X}_*}} & \mathbf{K}_{i_{\mathbf{X}_*,\mathbf{X}}}\\ \mathbf{K}_{i_{\mathbf{X},\mathbf{X}_*}} & \mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I}\end{bmatrix}\right)}{\sum_{i=1}^{K}\mathcal{N}(\mathbf{y}|m_i(X);\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I})}, \quad (6)$$

where $[\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}]_{nn'} = k_i(\mathbf{x}_n,\mathbf{x}_{n'})$, and $\boldsymbol{\theta}$ contains hyperparameters for all the individual GPs and the weights. Using the matrix inversion lemma for block matrices (see Appendix A.3 of (Williams & Rasmussen, 2006)),

$$p(\mathbf{f}_*|\mathbf{y},\mathbf{X},\mathbf{X}_*,\boldsymbol{\theta}) = \sum_{i=1}^{K} w_i^a\mathcal{N}(\mathbf{f}_*|m_i^a(\mathbf{X}_*);\mathbf{K}_{i_{\mathbf{X}_*\mathbf{X}_*}}^a), \quad (7)$$

where, $w_i^a = \frac{w_i\mathcal{N}(\mathbf{y}|\, m_i(\mathbf{X});\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I})}{\sum_{i=1}^{K} w_i\mathcal{N}(\mathbf{y}|m_i(\mathbf{X});\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I})}$, $m_i^a(\mathbf{X}_*) = m_i(\mathbf{X}_*)+\mathbf{K}_{i_{\mathbf{X}_*,\mathbf{X}}}(\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I})^{-1}(\mathbf{y}-m_i(\mathbf{X}))$, and $\mathbf{K}_{i_{\mathbf{X}_*\mathbf{X}_*}}^a = \mathbf{K}_{i_{\mathbf{X}_*,\mathbf{X}_*}} - \mathbf{K}_{i_{\mathbf{X}_*,\mathbf{X}}}(\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I})^{-1}\mathbf{K}_{i_{\mathbf{X},\mathbf{X}_*}}$. This is basically a weighted sum of posterior for the individual GPs (Eq: 1), and the posterior weight for each member is proportional to its prior weight multiplied with the marginal likelihood. In order to find the optimal hyperparameter values, we again minimize the negative log marginal likelihood, given by, $-\log(p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})) = \sum_{i=1}^{K} w_i\mathcal{N}\left(\mathbf{y}|m_i(\mathbf{X});\mathbf{K}_{i_{\mathbf{X},\mathbf{X}}}+\sigma_n^2\mathbf{I}\right)$.

3. Numerical Experiments

The different kernels mentioned have pros and cons, and it's important to judiciously choose test cases to demonstrate them. I will start with 1D and 2D synthetic data to make comparisons between different kernels, and finish by applying them to real data.

3.1. Synthetic Data

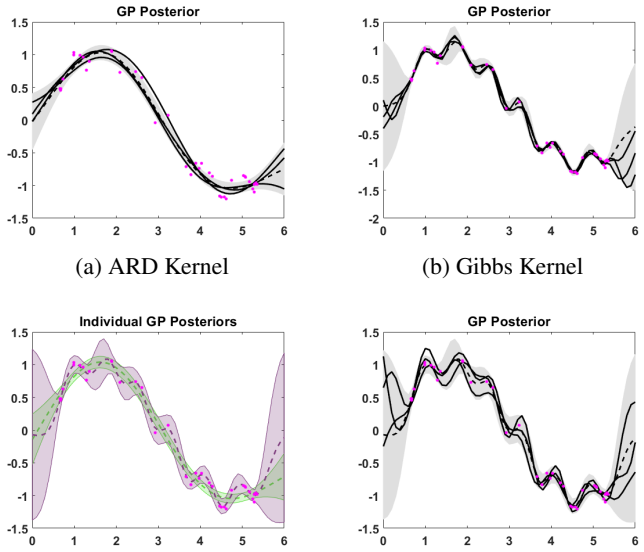
3.1.1. 1D Example

In this example, I first compare the performance of ARD, Gibbs, and weighted mixture of GPs method on synthetic data generated using superposition of two sine waves. ***Data generation:*** Noisy training data is generated at 40 random points according to the function $y = \sin x + 0.2\sin 8x + 0.05\mathcal{N}(0,1)$ for $0.5 \le x \le 5.5$ — the true underlying function is $f(x) = \sin x + 0.2\sin 8x$. ***Training:*** I choose a constant mean function for each candidate method, and a quadratic polynomial parameterized by two parameters for the spatially varying length scale for the Gibbs kernel. For the weighted mixture, I use two mixture components. For the Gibbs kernel, the GPML toolbox (Rasmussen & Nickisch) was used (which requires choice of the valid mean function to act as the length-scale function). Minimization of negative log likelihood in each case in order to find the optimal parameter values used the MINIMIZE.M function of the GPML toolbox, employing a Polack-Ribiere flavor of conjugate gradient method. At max, 200 iterations were performed irrespective of however many parameters were to be optimized. ***Results:*** Posterior GP predictions are provided in Fig. 1. For the ARD kernel, the high-freq. component is being treated as noise, and we basically fit according to the longer length-scale. For the Gibbs kernel, we are able to fit the data very well (which, in my opinion, happens because of a low σ_n value in $\boldsymbol{\theta}$ found by the optimization procedure) and the parameters for length-scale are without much variation in our domain. This behavior commonly leads to overfitting. For the weighted mixture of the two GPs, each of the component discovers one of the sine modes, and have nearly equal posterior weights. Mean Squared Error (MSE; contains both variance and bias) is computed by comparing against the true function values at 61 equally spaced points in

Table 1. Performance of different kernels on 1D data generated using 2 sines.

	ARD	Gibbs	Weighted Mixture
MSE	0.0246	0.0066	0.0136
- Log Marg. Lik.	-10.06	-24.21	-21.73

$0.5 \leq x \leq 5.5$, and provided in Table 1, along with the log marginal likelihood values. By MSE, the Gibbs kernel performs the best, while the higher MSE (nearly midway w.r.t to ARD) for weighted mixtures can be mainly attributed to the presence of high variance in the posterior. A similar trend is seen in the negative log marginal likelihood values, and the values for Gibbs and weighted mixtures are close.



(a) ARD Kernel (b) Gibbs Kernel

(c) Weighted Mixture of GPs (individual components) (d) Weighted Mixture of GPs

Figure 1. Posterior GPs found using different kernels. The magenta dots show observed data, dashed lines the mean, shaded regions denote 2 standard deviations from the mean, and the solid lines are samples from the posterior GP.

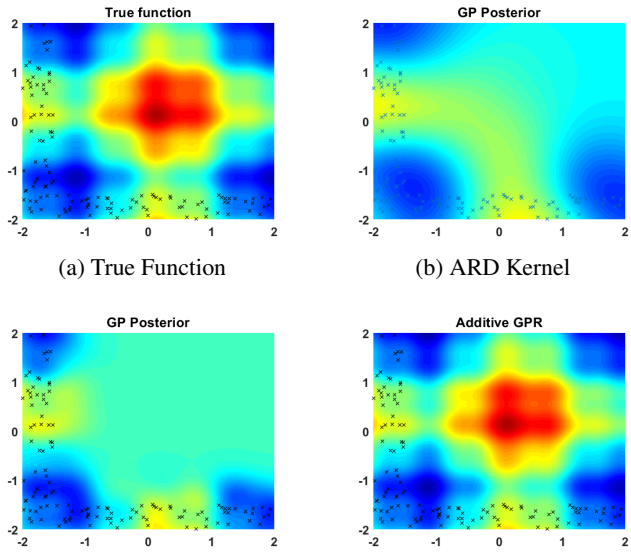
I also compared the performance of these three methods on data generated from a sine with spatially varying frequency, $y = \sin 2\pi 2^{(x-4.5)/2} + 0.05\mathcal{N}(0,1)$, but due to space constraints the results are not provided. As expected, the Gibbs kernel was able to perform much better, and both the components of weighted mixture gave similar fitting as the single ARD kernel, unable to adapt to the varying length-scale.

3.1.2. 2D Example

I use a modified version of the testcase presented in (Duvenaud et al., 2011), replacing Gibbs kernel with the additive GPR. ***Data generation:*** Noisy data is generated at 500 random locations according to $f(\mathbf{x}) = \sum_{i=1}^{2}(\sin(2x_i+1) + 0.2\sin(8x_i+1))$; training data is created by normalizing f with its standard deviation and adding noise $\sim 0.05\mathcal{N}(0,1)$. We keep only the points which lie in a L-shaped area defined by $x_1 \leq -1.5$ or $x_2 \leq -1.5$. ***Training:*** The ARD and weighted mixture of GPs is as in the previous section, again with only 2 mixture components. For the additive GPR, I use the implementation from (Duvenaud et al., 2011), and codes provided at `https://github.com/duvenaud/additive-gps`. Interactions up to 2^{nd} order are considered and L-BFGS with 200 iterations is used for optimization. ***Results:*** Posterior mean for each of the method is provided in Fig. 2, along with the true function. The additive GPR performs much better than ARD kernel or the weighted mixture of GPs, because it found the smaller length scale during optimization, and was able to replicate the pattern in the rest of the domain due to the 2^{nd} order interactions. The weighted mixture method, even with smaller length scales, does wasn't able to find the correct structure in the region with no data. Both mixture components had the same length-scale, probably because the optimizer did not find the correct minima. The MSE shown in table 2 (compared against true function evaluated on a grid with spacing of 0.1×0.1) for the additive GPR is much smaller than the other two (corroborates what we see in Fig. 2), but somehow the negative log marginal likelihood value for weighted mixture is smaller.

Table 2. Performance of different methods on the 2D data.

	ARD	Weigh. Mix.	Add. GPR
MSE	0.92	0.92	0.0072
- Log Marg. Lik.	-15.32	-129.39	-104.98



(a) True Function (b) ARD Kernel

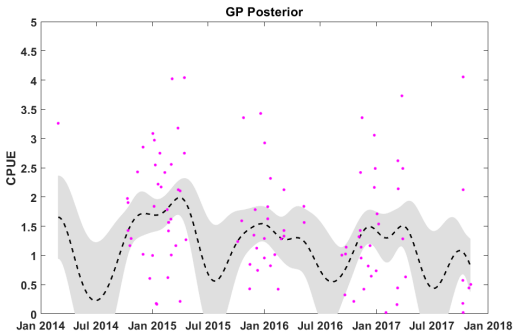
(c) Weighted Mixture of GPs (d) Additive GPR

Figure 2. Posterior mean of GPs found using different methods. The crosses mark the data locations, and all the plots share the same colorbar.

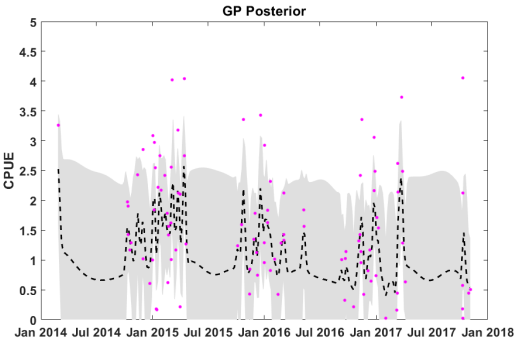
3.2. Real Data

We compare the ARD kernel and weighted mixtures GP on some real data. The first example consists of analysing fish-catch data from the Minicoy island in the Lakshadweep archipelago , collected by the Dakshin foundation as a part of their fisheries monitoring program (Jan 2014 - Jan 2018). The data consists of daily fish-catch, time spent in sea, amount of diesel used, etc. We can compute catch-per-unit-effort (CPUE) $equiv$ (total catch per day)/(total time spent

in the sea). In order to train the GPs, I sample 100 data points randomly and perform the regression. In this example, I do not perform optimization of hyperparameters and initialize them by trial and error. In this data, we expect to have multiple time-scales because fishing is a seasonal activity (with no fishing season every year), and the fishermen tend to rest every few weeks. Thus, using the weighted mixture method, we could use and add two GPs each having the desired time-scales, as shown in Fig. 3, which tends to better capture the weekly variations.



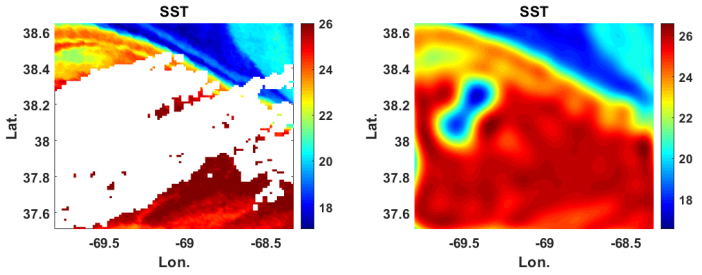
(a) ARD Kernel



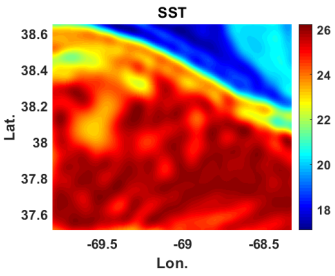
(b) Weighted Mixture of GPs

Figure 3. The magenta dots show observed fish-catch data, dashed lines the mean, and the shaded regions denote 2 standard deviations from the mean for the posterior GPs.

The second application is to sea-surface-temperature (SST) images, which are often marred by cloud cover (Prempraneerach et al., 2017). The aim is to find the temperature with uncertainty estimates in the regions of cloud cover, using the satellite measurements available in the regions of clear sky. I use an SST image from Mar 03, 2018 of the Atlantic ocean for analysis, where SST contains long and short distance correlations due to the presence of the Gulf stream and small eddies. I do not perform hyperparameter optimization, due to CG's tendency to get stuck in local minima. In Fig. 4, note the seemingly unphysical presence of a very cold region in the middle of hot water caused by longer correlations. In the weighted mixture method, we do not see the cold patch, probably because of the presence of shorter scales — the surrounding region is hot.



(a) SST Data (b) ARD Kernel



(c) Weighted Mixture of GPs

Figure 4. (a) The SST image, with the white regions corresponding to the cloud cover. (b) & (c) The posterior means of GPs found using different methods.

4. Computational Cost

For a standard GP, the cost of prediction scales as $\mathcal{O}(N^2)$, while the cost of training scales as $\mathcal{O}(N^3)$. For the Gibbs kernel, the only increase in the computational cost can be attributed to the need for length-scale function evaluations while creating the covariance matrices. For the additive kernel, the cost of creating the covariance matrix is $\mathcal{O}(N^2DR)$, where R is the degree of interaction (max value D). For the weighted mixture method, the prediction and training cost is multiplied by number of components K. Another major factor is the hyperparameter optimization, which quickly becomes costly, especially for the weighted mixture method, with increasing K.

5. Conclusion

In this project, different GP methods were evaluated for their ability to handle multiscale data. We compared the ARD kernel, Gibbs kernel, additive GPR, and a weighted mixture of GPs method. Using 1- and 2-dimensional synthetic data created from superposition of sine functions of different frequencies, we were able to analyse the pros and cons of these methods in different multiscale scenarios. Ocean data often contains multiscale features, and we demonstrated how use of an ARD kernel and the weighted mixture of GPs method can be used to regress fish-catch data from the Lakshadweep islands in India, and to fill gaps in satellite measurements of sea-surface-temperature due to cloud cover. Computational cost comparisons were also made, and hyperparameter optimization proved to be a significant challenge. While this report has demonstrated proof of concept, more rigorous analysis and testing is needed to make these methods an operational tool for ocean applications.

Acknowledgements

I would like to thank the staff of 6.435 (Spring 2019), Prof. Tamara Broderick and TA Brian Trippe for their valuable insight and help, as well as Corbin Foucart for proofreading.

References

Alspach, D. and Sorenson, H. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448, 1972.

Cushman-Roisin, B. and Beckers, J.-M. *Introduction to geophysical fluid dynamics: physical and numerical aspects*, volume 101. Academic press, 2011.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. Additive gaussian processes. In *Advances in neural information processing systems*, pp. 226–234, 2011.

Fox, E. and Dunson, D. B. Multiresolution gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 737–745, 2012.

Gibbs, M. N. *Bayesian Gaussian processes for regression and classification*. PhD thesis, Citeseer, 1998.

Prempraneerach, P., Perdikaris, P., Karniadakis, G., and Chryssostomidis, C. Sea surface temperature estimation from satellite observations and in-situ measurements using multifidelity gaussian process regression. In *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, pp. 28–33. IEEE, 2017.

Rasmussen, C. E. and Ghahramani, Z. Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pp. 881–888, 2002.

Rasmussen, C. E. and Nickisch, H. Gaussian process regression and classification toolbox. URL http://www.gaussianprocess.org/gpml/code.

Sondergaard, T. and Lermusiaux, P. F. Data assimilation with Gaussian mixture models using the dynamically orthogonal field equations. Part II: Applications. *Monthly Weather Review*, 141 (6):1761–1785, 2013a.

Sondergaard, T. and Lermusiaux, P. F. Data assimilation with Gaussian mixture models using the dynamically orthogonal field equations. Part I: Theory and scheme. *Monthly Weather Review*, 141(6):1737–1760, 2013b.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

Zhang, Z., Duraisamy, K., and Gumerov, N. A. Efficient multiscale gaussian process regression using hierarchical clustering. *arXiv preprint arXiv:1511.02258*, 2015.