Unimodality as an extention of Monotonicity in Gaussian Processes

Author EMAILID@AALTO.FI
AALTO UNIVERSITY

Abstract

Dummy abstract!! In probability theory and statistics, a Gaussian process is a stochastic process (a collection of random variables indexed by time or space), such that every finite collection of those random variables has a multivariate normal distribution, i.e. every finite linear combination of them is normally distributed. The distribution of a Gaussian process is the joint distribution of all those (infinitely many) random variables, and as such, it is a distribution over functions with a continuous domain, e.g. time or space. A machine-learning algorithm that involves a Gaussian process uses lazy learning and a measure of the similarity between points (the kernel function) to predict the value for an unseen point from training data. The prediction is not just an estimate for that point, but also has uncertainty informationit is a one-dimensional Gaussian distribution (which is the marginal distribution at that point)[2, 4].

Keywords: Gaussian Processes, Informative Priors, Unimodality

1. Introduction

Gaussian processes are probabilistic models which offer a non parameteric fully bayesian framework for learning a regression task. The prior information is usually encoded within the choice of the mean and covariance functions along with the hyperparameters of these function. The prior choice of monotonicity constraint was shown to be enforcable with the use of psuedo inputs, Gaussian process derivatives and using a sigmoidal link function to enforce the derivatives of a given sign [3]. In this project we try extend the monotonicity constraint to enforce a unimodality constraint.

2. Related Works

2.1 Gaussian Processes

We can model a Gaussian process regression as a stochastic process with input X, evaluating to the underlying latent function f, to which the noise variance is added to form the obseved output Y.

$$\begin{aligned}(\mathbf{Y}|\mathbf{X}) &\sim p(\mathbf{Y}|f)p(f|\mathbf{X}) \\ &\sim \mathcal{N}\big(0, \sigma^2\mathbf{I}\big)\mathcal{N}\big(m(\mathbf{X}), k(\mathbf{X},\mathbf{X})\big) \\ &\sim \mathcal{N}\big(m(\mathbf{X}), k(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}\big)\end{aligned}$$

To make predictions f^* for new input points X^* we have the following joint distribution,

$$\begin{bmatrix}\mathbf{Y}\\ f^*\end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix}K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}^*)\\ K(\mathbf{X}^*,\mathbf{X}) & K(\mathbf{X}^*,\mathbf{X}^*)\end{bmatrix}\right)$$

The conditional distribution of the prediction follows the normal form,

$$f^*|\mathbf{X}^*,\mathbf{X},f \sim \mathcal{N}\Big(K(\mathbf{X}^*,\mathbf{X})\big(K(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I}\big)^{-1}\mathbf{Y}, \\ K(\mathbf{X}^*,\mathbf{X}^*) - K(\mathbf{X}^*,\mathbf{X})\big(K(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I}\big)^{-1}K(\mathbf{X},\mathbf{X}^*)\Big)$$

2.2 Gaussian Process derivatives

Differentiation is a linear operator due to which the derivative of a GP also remains gaussian. The derivative information can be hence be incorporated into the GP model. The RBF covariance function incorporating the derivative information is has the form,

$$Cov\big[f^{(i)}, f^{(j)}\big] = \eta^2 exp\Big(-\frac{1}{2}\sum_{d=1}^{D}\rho_d^{-2}\big(x_d^{(i)}-x_d^{(j)}\big)^2\Big)$$

$$Cov\left[\frac{\partial f^{(i)}}{\partial x_g^{(i)}}, f^{(j)}\right] = \eta^2 exp\Big(-\frac{1}{2}\sum_{d=1}^{D}\rho_d^{-2}\big(x_d^{(i)}-x_d^{(j)}\big)^2\Big)\Big(-\rho_g^{-2}\big(x_g^{(i)}-x_g^{(j)}\big)\Big)$$

$$Cov\left[\frac{\partial f^{(i)}}{\partial x_g^{(i)}}, \frac{\partial f^{(j)}}{\partial x_h^{(j)}}\right] = \eta^2 exp\Big(-\frac{1}{2}\sum_{d=1}^{D}\rho_d^{-2}\big(x_d^{(i)}-x_d^{(j)}\big)^2\Big) \\ \rho_g^{-2}\Big(\delta_{gh} - \rho_h^{-2}\big(x_h^{(i)}-x_h^{(j)}\big)\big(x_g^{(i)}-x_g^{(j)}\big)\Big)$$

2.3 Monotonicity using derivative information

Using the derivative information we can enforce a monotonicity constraint by using sigmoidal likelihood for the derivative observations. A set of M points($\mathbf{X}_\partial$) over the input space are choosen and monotonicity constraint is enforced over those points instead of evaluating the derivative over the whole input space.

$$p\left(\begin{bmatrix}f\\ f_\partial\end{bmatrix}\middle|\begin{bmatrix}\mathbf{Y}\\ \mathbf{Y}_\partial\end{bmatrix}\right) = \frac{1}{C}\,p\left(\begin{bmatrix}f\\ f_\partial\end{bmatrix}\middle|\begin{bmatrix}\mathbf{X}\\ \mathbf{X}_\partial\end{bmatrix}\right)p(\mathbf{Y}|f)p(\mathbf{Y}_\partial|f_\partial) \tag{1}$$

The last probability term acts as the derviative likelihood driving function values without monotonicity to a low probability. The derivative likelihood has the form,

$$p(\mathbf{Y}_\partial|f_\partial) = \prod_{i=1}^{M}\phi\Big(m f_\partial^{(i)}\frac{1}{v}\Big) \tag{2}$$

where M is the number of psuedo derivative points, ϕ is a sigmoidal link function, m is the latent derivative function which gives us the sign of dervivative that we are trying to enforce and the parameter v controls the steepness of the sigmoidal link function.

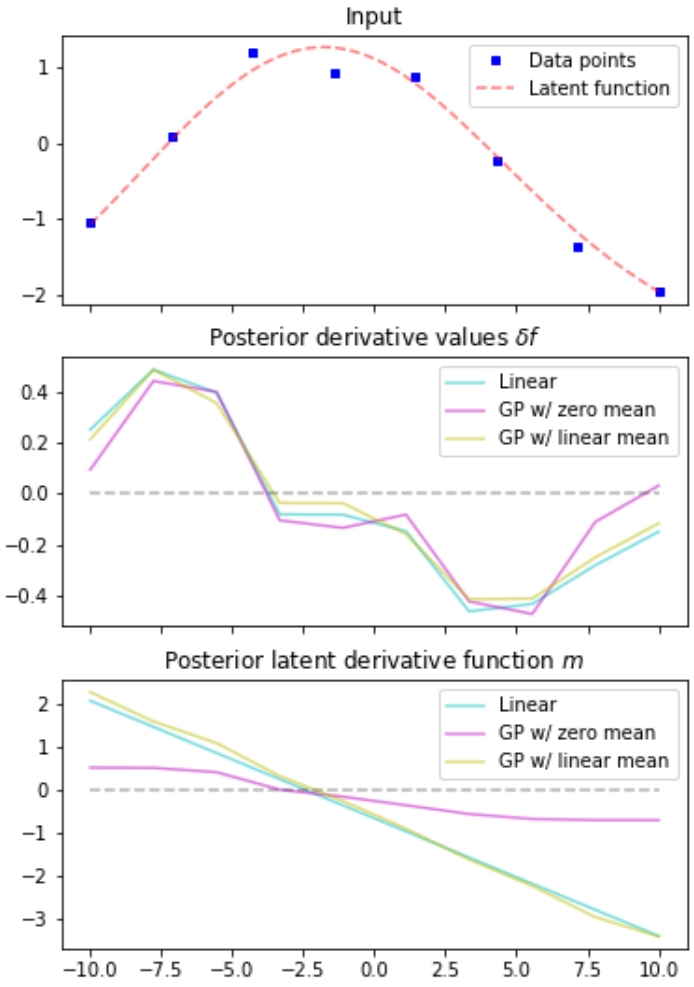

Figure 1: Example of model

3. Unimodality constraint using Monotonicity

The latent derivative function m in equation 2 can be modeled as an input dependant function which can be used to enforce shape constraints. The unimodality information can be modelled by a using parameteric monotonic function to represent the derivative information m. The primary role of the monotonic derivative function model would be to learn the mode of the data accurately, where the sign of the derivative would flip.

We experimented with three different models for the latent derivative function:

1. Linear model of the form $m(x) = ax + b$
2. A zero mean Gaussian process of the form $m(x) = GP(0, k_m(x, x))$
3. A linear mean Gaussian process $m(x) = GP(ax + b, k_m(x, x))$

Figure 1 shows how the unimodality constraint develops under the given model.

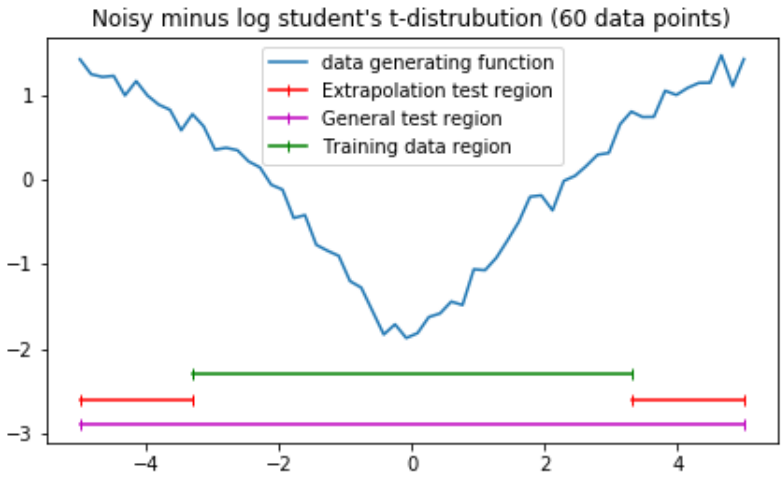

Figure 2: Input to the used to generate the error curve

4. Experiments

To evaluate the advantages of using the above models, the models were trained on a unimodal dataset (Figure 2). The method of inference was using MCMC sampling using the STAN [1].

From the error error curves of the Fig 3 we can see that the unimodality enforced models learn much faster than normal GP regression.

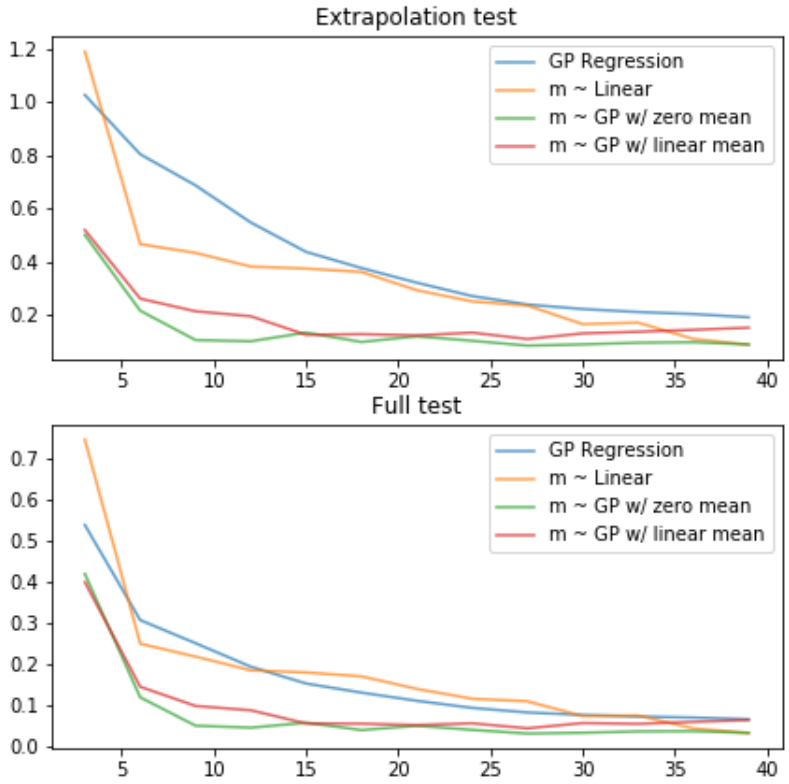

Figure 3: Testing error as a function of number of data points used for training

References

[1] Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32, 2017.

[2] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

[3] Jaakko Riihimki and Aki Vehtari. Gaussian processes with monotonicity information. 9:645–652, 01 2010.

[4] Ercan Solak, Roderick Murray-Smith, W.E. Leithead, Douglas Leith, and C.E. Rasmussen. Derivative observations in gaussian process models of dynamic systems. 16, 02 2003.