Final Report
STAT 689 Spring 2022

Carson James
Department of Statistics
Texas A&M University
College Station, TX 77840
`carsonaj@tamu.edu`

Abstract

This report will discuss an approximation to the posterior mean of a Gaussian process by a neual network that improves over the computational efficiency and true data generating function estimation accuracy of the exact posterior mean.

1 Background

Note 1.0.1. The approximation relies upon tools from the theory of Gaussian processes, Banach spaces, Hilbert spaces, reproducing kernel Hilbert spaces and convex analysis. The relevant results will be mentioned.

Definition 1.0.2. Let T be a set and $c : T^2 \to \mathbb{R}$. Then c is said to be **positive definite** if for each $(x_j)_{j=1}^n \in T^n$, the matrix $c(x, x)$ is positive definite.

Definition 1.0.3. Let T be a set, $(\Omega, \mathcal{F}, P)$ a probability space, $\mu : T \to \mathbb{R}$, $c : T^2 \to \mathbb{R}$ symmetric and positive definite and $f : T \to L^2(\Omega, \mathcal{F}, P)$ (i.e. f is a random function from T to $\mathbb{R}$). Then f is said to be a **Gaussian Process** with mean function μ and covariance function c, denoted $f \sim GP(\mu, c)$, if for each $x = (x_j)_{j=1}^n \in T^n$, $f(x) \sim N_n(\mu(x), c(x, x))$.

Theorem 1.0.4. Let T be a set, $c : T^2 \to \mathbb{R}$ positive definite, $x = (x_j)_{j=1}^n \in T^n$, $y = (y_j)_{j=1}^n \in \mathbb{R}^n$. Suppose we have the following model:

$$\begin{gathered} y_i = f(x_i) + \epsilon_i \\ \epsilon_i \sim N(0, \sigma^2) \\ f \sim GP(0, c) \end{gathered}$$

Then

$$f|x, y \sim GP(\tilde{\mu}, \tilde{c})$$

where

$$\tilde{\mu}(t) = c(t, x)[c(x, x) + \sigma^2 I]^{-1} y$$

and

$$\tilde{c}(s, t) = c(s, t) - c(s, x)[c(x, x) + \sigma^2 I]^{-1} c(x, t)$$

Definition 1.0.5. Frechet Derivative:
Let X, Y be a banach spaces, $A \subset X$ open, $f : A \to Y$ and $x_0 \in A$.

- Then f is said to be **Frechet differentiable at** x_0 if there exists $Df(x_0) \in L(X, Y)$ such that,

$$f(x_0 + h) = f(x_0) + Df(x_0)(h) + o(\|h\|) \qquad \text{as } h \to 0$$

- If f is Frechet differentiable at x_0, we define the **Frechet derivative of f at x_0** to be $Df(x_0)$.
- We say that f is **Frechet differentiable** if for each $x \in A$, f is Frechet differentiable at x.
- If f is Frechet differentiable, we define the **Frechet derivative of** f, denoted $Df : A \to L(X, Y)$, by $x \mapsto D^{(1)}f(x)$.

Theorem 1.0.6. Let X be a Banach space, $f : X \to \mathbb{R}$ and $x_0 \in X$. If f is Frechet differentiable at x_0 and f has a local minimum at x_0, then $Df(x_0) = 0$.

Theorem 1.0.7. Let X be a Banach space and $f : X \to \mathbb{R}$ be continuously second order Frechet differentiable. If for each $x \in X$, $D^2f(x)$ is positive definite, then f is strictly convex.

Definition 1.0.8. Let H be a Hilbert space, $f : H \to \mathbb{R}$ and $x_0 \in H$. Suppose that f is Frechet differentiable at x_0. Then $Df(x_0) \in H^*$. We define the **gradient of f at** x_0, denoted $\nabla f(x_0) \in H$, via the Riesz representation theorem to be the unique element of H satisfying

$$\langle \nabla f(x_0), y\rangle = Df(x_0)(y) \quad \text{for each } y \in H$$

Definition 1.0.9. Let T be a set and $H \subset \mathbb{R}^T$ a hilbert space. For $t \in T$, we define the **evaluation functional at** t, denoted $l_t : H \to \mathbb{R}$, by

$$l_t(f) = f(t)$$

The space H is said to be a **reproducing kernel Hilbert space (RKHS)** if for each $t \in T$, $l_t \in H^*$ (i.e. l_t is bounded).
If H is an RKHS, the Riesz representation theorem implies that for each $t \in T$, there exists $c_t \in H$ such that for each $f \in H$, $\langle c_t, f\rangle = f(t)$.

If H is an RKHS, we define the **reproducing kernel** associated to H, denoted $c_H : T^2 \to \mathbb{R}$, by

$$c_H(s,t) = \langle c_s, c_t\rangle$$

Theorem 1.0.10. Let T be a set and $c : T^2 \to \mathbb{R}$. If c is symmetric and positive definite, then there exists a unique reproducing kernel Hilbert space $H \subset \mathbb{R}^T$ such that $c_H = c$.

Definition 1.0.11. Let T be a set, $c : T^2 \to \mathbb{R}$ a symmetric, postivie definite kernel on T, $H \subset \mathbb{R}^T$ the corresponding RKHS, $x = (x_j)_{j=1}^n \in T^n$, $\lambda > 0$ and $y = (y_j)_{j=1}^n \in \mathbb{R}^n$.
Define $L_{\lambda,y} : H \to \mathbb{R}$ by

$$L_{\lambda,y}(f) = \sum_{j=1}^n (y_j - f(x_j))^2 + \lambda \|f\|_H^2$$

and set

$$\hat{f}_{\lambda,y} = \arg\min_{f\in H} L_{\lambda,y}(f)$$

Theorem 1.0.12. Let T be a set, $c : T^2 \to \mathbb{R}$ a symmetric, postivie definite kernel on T, $H \subset \mathbb{R}^T$ the corresponding RKHS, $x = (x_j)_{j=1}^n \in T^n$, $\lambda > 0$ and $y = (y_j)_{j=1}^n \in \mathbb{R}^n$.
Then $L_{\lambda,y}$ is strictly convex.

Theorem 1.0.13. Let T be a set, $c : T^2 \to \mathbb{R}$ a symmetric, postivie definite kernel on T, $H \subset \mathbb{R}^T$ the corresponding RKHS, $x = (x_j)_{j=1}^n \in T^n$, $\lambda > 0$ and $y = (y_j)_{j=1}^n \in \mathbb{R}^n$.
Then there exist $(\hat{\alpha}_j)_{j=1}^n \subset \mathbb{R}$ such that

$$\hat{f}_{\lambda,y}(t) = \sum_{j=1}^n \hat{\alpha}_j c(t, x_j)$$

Standard multivariable calculus shows that $\hat{\alpha} = (c(x,x) + \lambda I)^{-1}y$ so that

$$\hat{f}_{\lambda,y}(t) = c(t,x)(c(x,x) + \lambda I)^{-1}y$$

Note 1.0.14. The above results tell us that in the context of Gaussian processes,

$$\hat{f}_{\sigma^2,y} = \tilde{\mu}$$

2 Approximation

Lemma 2.0.1. Let T be a set, $c : T^2 \to \mathbb{R}$ a symmetric, postivie definite kernel on T, $H \subset \mathbb{R}^T$ the corresponding RKHS, $x = (x_j)_{j=1}^n \in T^n$, $\lambda > 0$ and $y = (y_j)_{j=1}^n \in \mathbb{R}^n$.
Then for each $f_0 \in H$,

$$\nabla L_{\lambda,y}(f_0) = 2\left[\sum_{j=1}^n (f_0(x_j) - y_j)c_{x_j} + \lambda f_0\right]$$

Approximation 2.0.2. The preceeding lemma implies that in the context of Gaussian processes, we have that

$$\begin{aligned}\nabla L_{\lambda,y}(\tilde{\mu}) &= 2\left[\sum_{j=1}^n (\tilde{\mu}(x_j) - y_j)c_{x_j} + \sigma^2\tilde{\mu}\right]\\ &= 0\end{aligned}$$

Thus, if we approximate $\tilde{\mu}$ by a neural network $g_\theta : T \to \mathbb{R}$, we obtain the restriction that for each $t \in T$,

$$\sum_{j=1}^n (g_\theta(x_j) - y_j)c_{x_j}(t) + \sigma^2 g_\theta(t) = 0$$

So we may make a grid $(t_k)_{k=1}^a \subset T$ and for each $k \in \{1, \dots, a\}$,

$$\sum_{j=1}^n (g_\theta(x_j) - y_j)c_{x_j}(t_k) + \sigma^2 g_\theta(t_k) = 0$$

Now, taking absolute values or squares and summing we obtain the following restrictions:

- $$\sum_{k=1}^a \left|\sum_{j=1}^n (g_\theta(x_j) - y_j)c_{x_j}(t_k) + \sigma^2 g_\theta(t_k)\right| = 0$$
- $$\sum_{k=1}^a \left(\sum_{j=1}^n (g_\theta(x_j) - y_j)c_{x_j}(t_k) + \sigma^2 g_\theta(t_k)\right)^2 = 0$$

Finally, using the triangle inequality and Jensen's inequality we obtain the following loss functions:

- $$l_1(\theta) = \sum_{k=1}^a \left[\sum_{j=1}^n |(g_\theta(x_j) - y_j)|c_{x_j}(t_k) + \sigma^2|g_\theta(t_k)|\right]$$
- $$l_2(\theta) = \sum_{k=1}^a \left[\left(\sum_{j=1}^n (g_\theta(x_j) - y_j)c_{x_j}(t_k)\right)^2 + \sigma^4 g_\theta(t_k)^2\right]$$

Note 2.0.3. Either loss function l_1 or l_2 lets us train g_θ to approximate the exact posterior mean $\tilde{\mu}$. However in many cases the exact posterior mean does not align with a true data generating function f_0 (if there is one) in our regression model. We address this in the following approximation model.

Approximation 2.0.4. To augment the loss function $l \in \{l_1, l_2\}$, we add an additional MSE term:

$$MSE(\theta) = \sum_{j=1}^{n} (g_\theta(x_j) - y_j)^2$$

For $\alpha \in (0, 1)$, we weight the original loss functions and the MSE term in order to obtain the following augmented loss functions:

- $l_1'(\theta) = \alpha l_1(\theta) + (1 - \alpha)\text{MSE}(\theta)$

- $l_2'(\theta) = \alpha l_2(\theta) + (1 - \alpha)\text{MSE}(\theta)$

Note 2.0.5. The motivation for the augmented loss function is to choose $\alpha > 0.5$ so that we have an overall behavior similar to that of the exact posterior mean, but within the data x, y, the MSE term dominates and we have a better fit to the true data generating function.

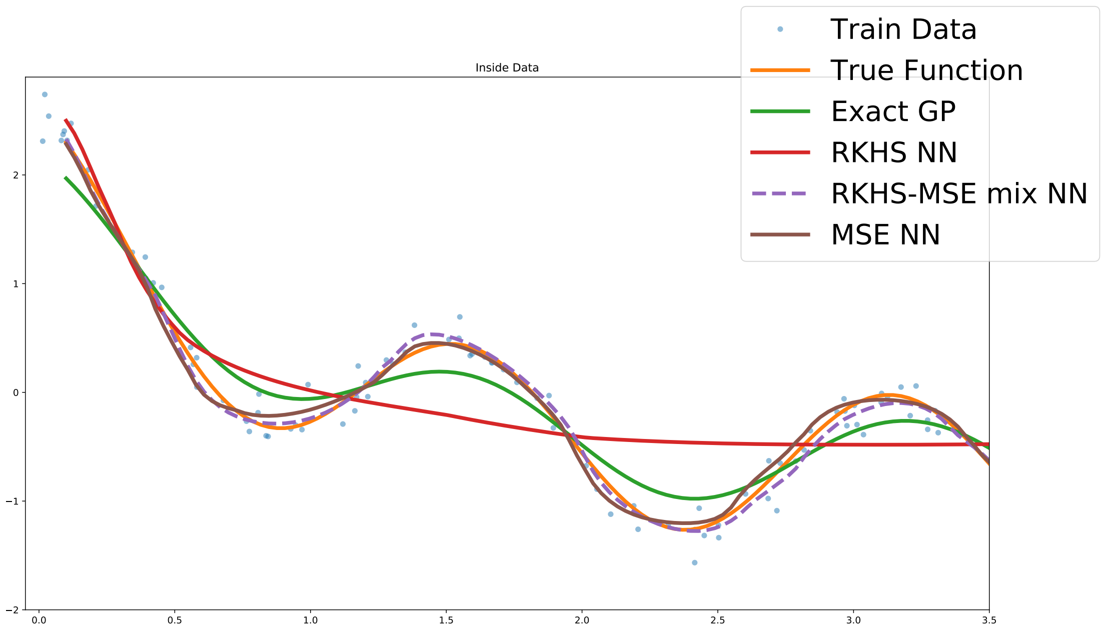
3 Results

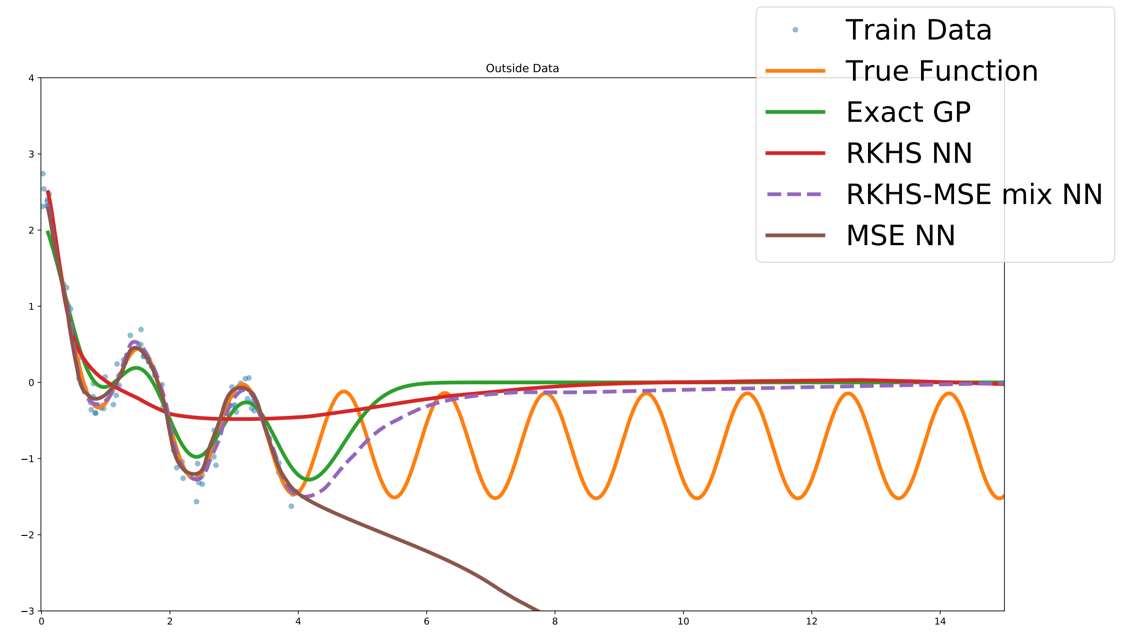
Model 3.0.1. We generated data according to the following model:

$$\begin{aligned} y_i &= f_0(x_i) + \epsilon_i \\ \epsilon_i &\sim N(0, 0.1^2) \\ x_i &\sim \text{U(0,4)} \\ f_0(x) &= 2e^{-x} + 0.5\cos(8x) + 4 \end{aligned}$$

and trained a neural network g_θ on the grid of evenly spaced grid $(t_k)_{k=1}^{40} \subset [0, 40]$.

Note 3.0.2. Below we show the results after training g_θ with the l_2 loss function (RKHS NN), the l_2' loss function with $\alpha = 0.7$ (RKHS-MSE mix NN) and the MSE loss function (MSE NN). We included photos to compare performance inside the data on $[0, 4]$ and outside the data on $[0, 15]$.





Note 3.0.3. There are a few observations to make.

- The RKHS NN does not align well with the exact posterior mean inside the data, but it does revert back to the prior outside the training data like the posterior mean. Neither the RKHS NN nor the exact posterior mean fit the true data generating function well inside the training data.
- The MSE NN fits the true data generating function well inside the training data, but it fits poorly outside the training data
- The RKHS-MSE mix NN fits the true data generating function well inside the training data and reverts back to the prior mean outside the training data.

So the neural network g_θ trained by the augmented loss function $l_2'(\theta)$ yields a better approximation to the true data generating function f_0 than the posterior mean inside the data and retains the nice property of reverting back to the prior mean outside of the data. Another thing to note is that once the network is trained, it is no longer necessary to store the training data and evaluation takes constant time with respect to the size of the training data unlike the posterior mean. So this approximation offers a computational improvement over the posterior mean as well.

References

[1] Reproducing Kernel Hilbert Spaces, Wikipedia

[2] Representer Theorem, Wikipedia

[3] Analysis Notes, My GitHub