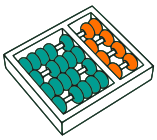


MO435 — Probabilistic Machine Learning
INSTITUTO DE COMPUTAÇÃO — UNICAMP

UNICAMP

Project 2

João Victor da Silva Guerra and Leonardo Alves de Melo *

Abstract

In this project, we implemented a Gaussian Processes algorithm, and evaluated it with samples of the sine, cossine and exponential functions. For each function, we evaluated the size of the training set and test set, varying in 5, 10 and 50 samples. In addition, our implementation was able to evaluate higher dimensional data, which we illustrated with a two-dimensional data. Finally, we evaluated the effects of the squared-exponential kernel parameters and different kernel functions on our predictions.

1 Introduction

A Bayesian Optmization ia a way to find the global maximun or minimun of a given function without calculating the derivative. The Gaussian Processes (GP) is a type of Bayesian Optmization, and performs a Bayesian inference over the functions. In this approach, the prior is defined over functions and, after seen some data, it can be used to find the porterior, which is a distribution over some arbitrary set of points of a function ([1]).

Given a set of points $x_1,...,x_n$, the objective is to calculate the corresponding $f(x_1),...,f(x_n)$. A GP assumes that $p(f(x_1),...,f(x_n))$ is a jointly Gaussian distribution, with some mean $\mu(x)$ and covariance $\Sigma(x)$. To get the covariance values, we have to define a positive definite kernel function $\kappa(x_i,x_j)$ as follows:

$$\Sigma_{ij} = \kappa(x_i,x_j) \tag{1}$$

1.1 Gaussian Processes for regression

In a GP regression, we assume that the prior on the regression function is a GP (Eq. (2)) with a mean function $m(x)$ (Eq. (3)) and a kernel function $\kappa(x,x')$ (Eq. (4)).

$$f(x) \sim GP(m(x),k(x,x')) \tag{2}$$

$$m(x) = \mathbb{E}[f(x)] \tag{3}$$

$$\kappa(x,x') = \mathbb{E}[(f(x)-m(x))(f(x')-m(x'))^T] \tag{4}$$

For a finite set of points, this process becomes a joint Gaussian distribution as follows:

$$p(f|X) = \mathcal{N}(f|\mu,K) \tag{5}$$

where $K_{ij} = \kappa(x_i,x_j)$ and $\mu = (m(x_1),...,m(x_n))$.

Now, suppose we have a training set $(D = (x_i,y_i), i = 1:N))$, where y_i is a noisy observation at x_i, yielded by $y = f(x)+\epsilon$, where $\epsilon \sim \mathcal{N}(0,\sigma_y^2)$. Further, given a test set X_* of size $N_* \times D$, we want predict the function outputs f_*. As we have noisy observations, our model must come "close", but is not necessary to interpolate the data.

The joint distribution of the observed data and the latent noise-free function is given by

$$\begin{pmatrix} y \\ f_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right) \tag{6}$$

where $K_y = \kappa(X,X)+\sigma_y^2 I_N$ is $N \times N$, $K_* = \kappa(X,X_*)$ is $N \times N_*$, and $K_{**} = \kappa(X_*,X_*)$ is $N_* \times N_*$. The second term of the covariance of the noisy observation ($\sigma_y^2 I_N$) is a diagonal because the noise term (σ_y^2) is independently added to each observation.

*117410 and 156188. j117410@dac.unicamp.br and leonardo.alves.melo.1995@gmail.com.

According to standard rules for conditioning Gaussian distributions ([1]), the posterior has the form

$$p(f_*|X_*, X, y) = \mathcal{N}(f_*|\mu_*, \Sigma_*) \tag{7}$$

$$\mu_* = K_*^T K_y^{-1} y \tag{8}$$

$$\Sigma_* = K_{**} - K_*^T K_y^{-1} K_* \tag{9}$$

Here, we implement a Gaussian Process regression and evaluate the effect of the model parameters, such as samples from different function, training and test set sizes, noise's standard deviation, kernel parameters, kernel functions, on the predictions (i.e. posterior).

2 Algorithm

The predictive mean is calculated by the Eq.(10). However, the direct computation of K_y^{-1} is not recommended because of numerical stabilities. Therefore, an alternative approach is to compute the Cholesky decomposition of K_y, which is given by Eq.(11).

$$\overline{f_*} = K_*^T K_y^{-1} y \tag{10}$$

$$K_y = LL^T \tag{11}$$

Then, we compute the predictive mean, variance and the log marginal likelihood, using Cholesky decomposition, as shown in Algorithm 1.

Algorithm 1 Gaussian Processes Regression

1: $L = cholesky(K + \sigma_y^2 I)$;
2: $\alpha = L^T \backslash (L \backslash y)$;
3: $\mathbb{E}[f_*] = K_*^T \alpha$;
4: $v = L \backslash K_*$;
5: $var[f_*] = \kappa(x_*, x_*) - v^T v$;
6: $log(p(y|X)) = -\frac{1}{2} y^T \alpha - \Sigma_i log L_{ii} - \frac{N}{2} log(2\pi)$;

3 Experiments

3.1 Data

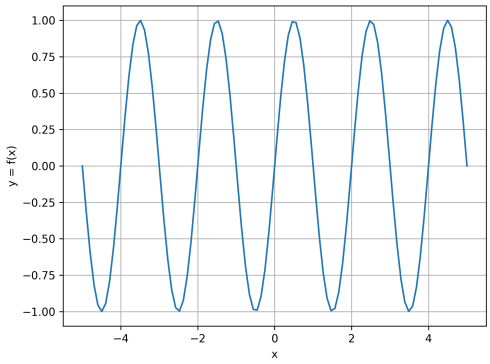
For our experiments, we sampled our training and test sets from different functions. For the functions, we selected sine (f_1, Eq.(12)), cosine (f_2, Eq.(13)), exponential (f_3, Eq.(14)) and a 2D periodic function (f_4, Eq.(15)). All functions are displayed in the range from -5 to 5 on the axes of the independent variables in Fig.1.

$$f_1(x) = sin(\pi x) \tag{12}$$

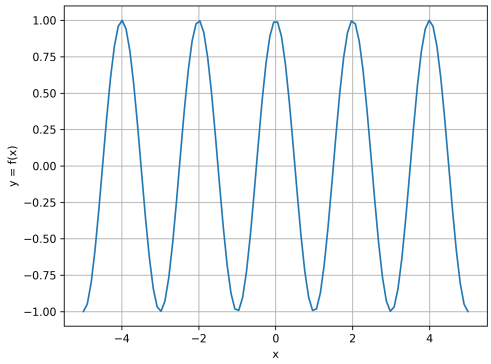
$$f_2(x) = cos(\pi x) \tag{13}$$

$$f_3(x) = e^x \tag{14}$$

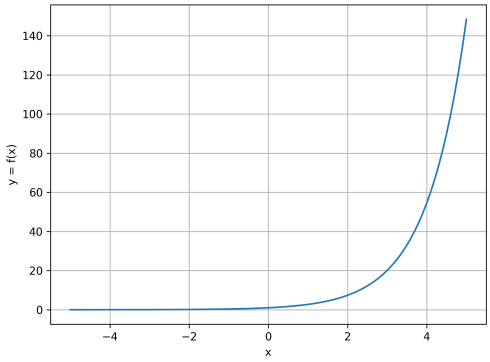
$$f_4(x_1, x_2) = sin(\pi x_1) + cos(\pi x_2) \tag{15}$$



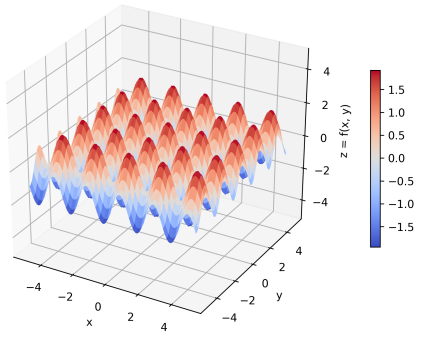
(a) Sine function (f_1)



(b) Cosine function (f_2)



(c) Exponential function(f_3)



(d) two-dimensional periodic function (f_4)

Figure 1. Functions sampled for our datasets.

3.2 Sampling procedure

For each one-dimensional function (f_1, f_2, f_3), we created sample size of 5, 10 and 50 data points, ranging from -5 to 5 in the independent variable. We applied this sampling procedure for training and test sets.

3.3 Kernel parameters

In our model, the default kernel function will be the squared-exponential (SE) kernel. For instance, the one-dimensional case is given by Eq.(16), where $\alpha = (\sigma, \ell)$ is the vector of hyperparameters. In addition, we have the prior of the noise's standard deviation (σ_y^2), which also has effect on our predictions. The default value of σ, ℓ and σ_y^2 are 1, 1 and 0.1, respectively.

$$\kappa_{SE}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) \tag{16}$$

Hence, we will evaluate the effects of different hyperparameters (σ, ℓ and σ_y^2) on the predictions of our regression. We applied the following values of σ: 0.01, 0.1, 0.5, 1, 10 and 100; the following values of ℓ: 0.1, 1, 2, 5, 10 and 100; and the following values of σ_y^2: 0.001, 0.01, 0.1, 1, 2 and 10.

3.4 Kernel functions

As mentioned above, we chose the SE kernel as our default basis function; however, we can choose different kernel functions. Then, we choose Ornstein-Uhlenbeck (OU; Eq.(17)), Rational Quadratic (RQ; Eq.(18)) and Exponential Sine Squared (ESS; Eq.(19)) kernels to evaluate the effect of the kernel function on predictions.

$$\kappa_{OU}(x, x') = \sigma^2 \exp\left(-\frac{|x - x'|}{2\ell^2}\right) \tag{17}$$

$$\kappa_{RQ}(x,x') = \sigma^2 \left(1 + \frac{(x-x')^2}{2\alpha\ell^2}\right)^{-\alpha} \tag{18}$$

$$\kappa_{ESS}(x,x') = \sigma^2 \exp\left(-\frac{2\sin\left(\frac{\pi|x-x'|}{w}\right)^2}{\ell^2}\right) \tag{19}$$

3.5 Higher dimensional data

Until now, our model are handling one-dimensional data, in order to calculate higher dimensional data, we need to extend the SE kernel from Eq.(16) to multidimensional data as follows:

$$\kappa(x_p,x_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(x_p - x_q)^T M (x_p - x_q)\right) \tag{20}$$

where the matrix M can be defined in several ways, but we use an isotropic matrix $M = \ell^{-2} I$.

4 Results

4.1 Effect of Sample Size

Here, we evaluate the different samples sizes for the training and test sets.

4.1.1 Training set

We evaluated effects of the training set size on our predictions, keeping the test set size fixed in 50 samples (Fig 2). In the plot, the blue dashed curve represented our target, the red line is our prediction of the target, and the light red region represents our confidence interval (level of uncertainty about the predicted value of our fitted function). We observed that regions where we had observations has less uncertainty in our prediction, i.e. a smaller confidence interval. On the other hand, regions with no observation has a higher uncertainty, i.e. a wider confidence interval. In addition, the distance bewteen two observations determine the uncertainty in the region between them; hence, smaller the distance, lower are the uncertainty. Further, with the increase of the training set size, our predictions gets relatively close to the target function. Therefore, we observed that 50 samples are a sufficient amount of training points to obtain a reliable curve in the studied range.

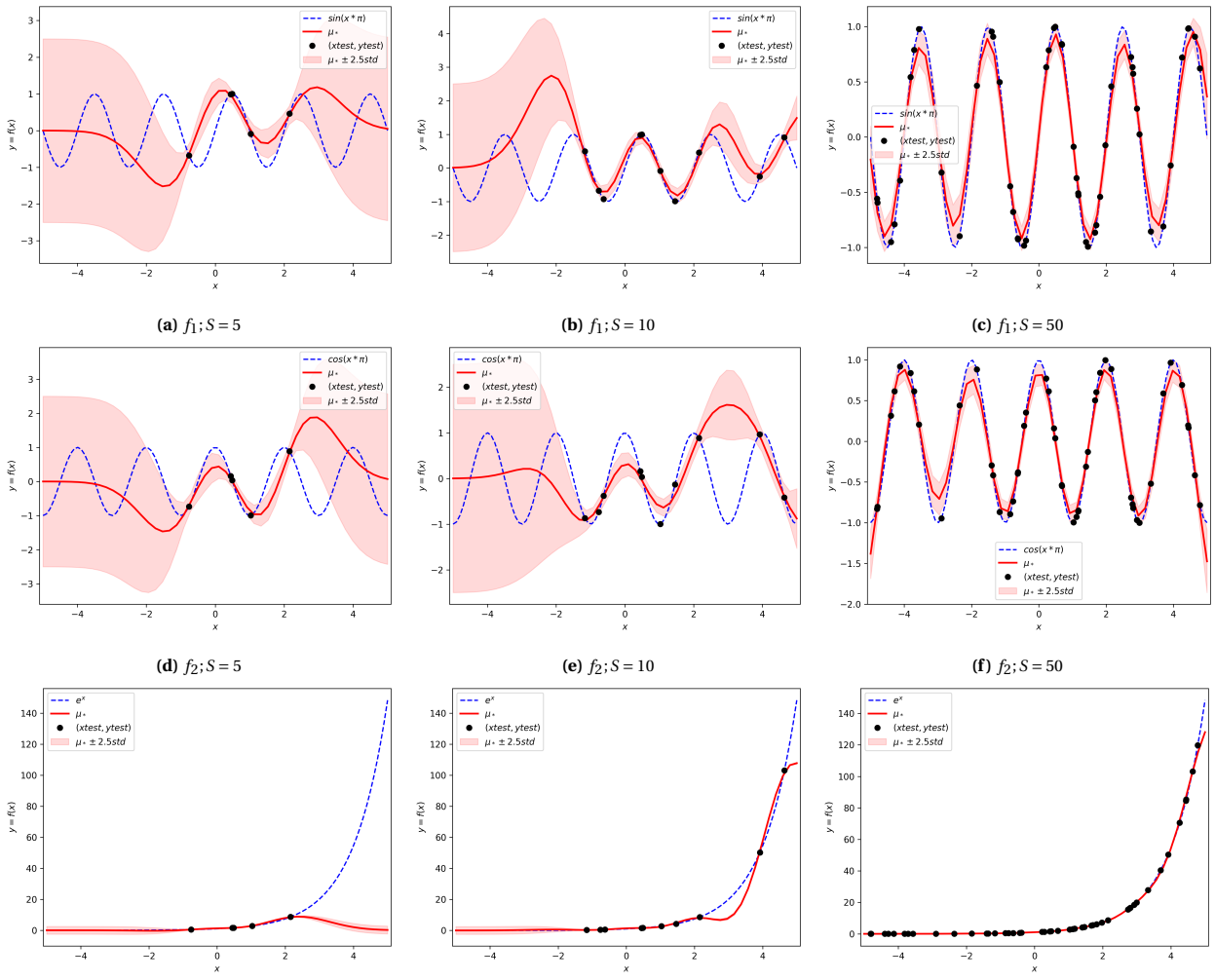
4.1.2 Test set

We evaluated effects of the test set size on our predictions, keeping the training set size fixed in 50 samples (Fig 3). We observed that with more points in the test set, the uncertainty decreases in our prediction, i.e. a smaller confidence interval; however, this effect is lesser when compared to the effect of varying the training set size. In addition, a small number of samples in the test set yields an incorrect prediction of our target, as can be seen for 5 samples, almost being possible to define which points where used in the test set. Then, with the increasing number of samples in the test set, our predictions gets relatively close to the target function. Finally, we suggest that the model appears to use the test samples to interpolate the fitted curve based on the training set information.

Therefore, we observed that 50 samples are a sufficient amount of test points to obtain a reliable curve in the studied range. However, the exponential curve had already achieved a reliable prediction with 10 samples, showing that there are no universal sample size for our model.

4.2 Effect of kernel parameters

Here, we evaluate the different values for the kernel parameters: σ, ℓ and σ_y^2. All experiments in this section used 50 samples in the training and test sets, using the function sine as the target function.

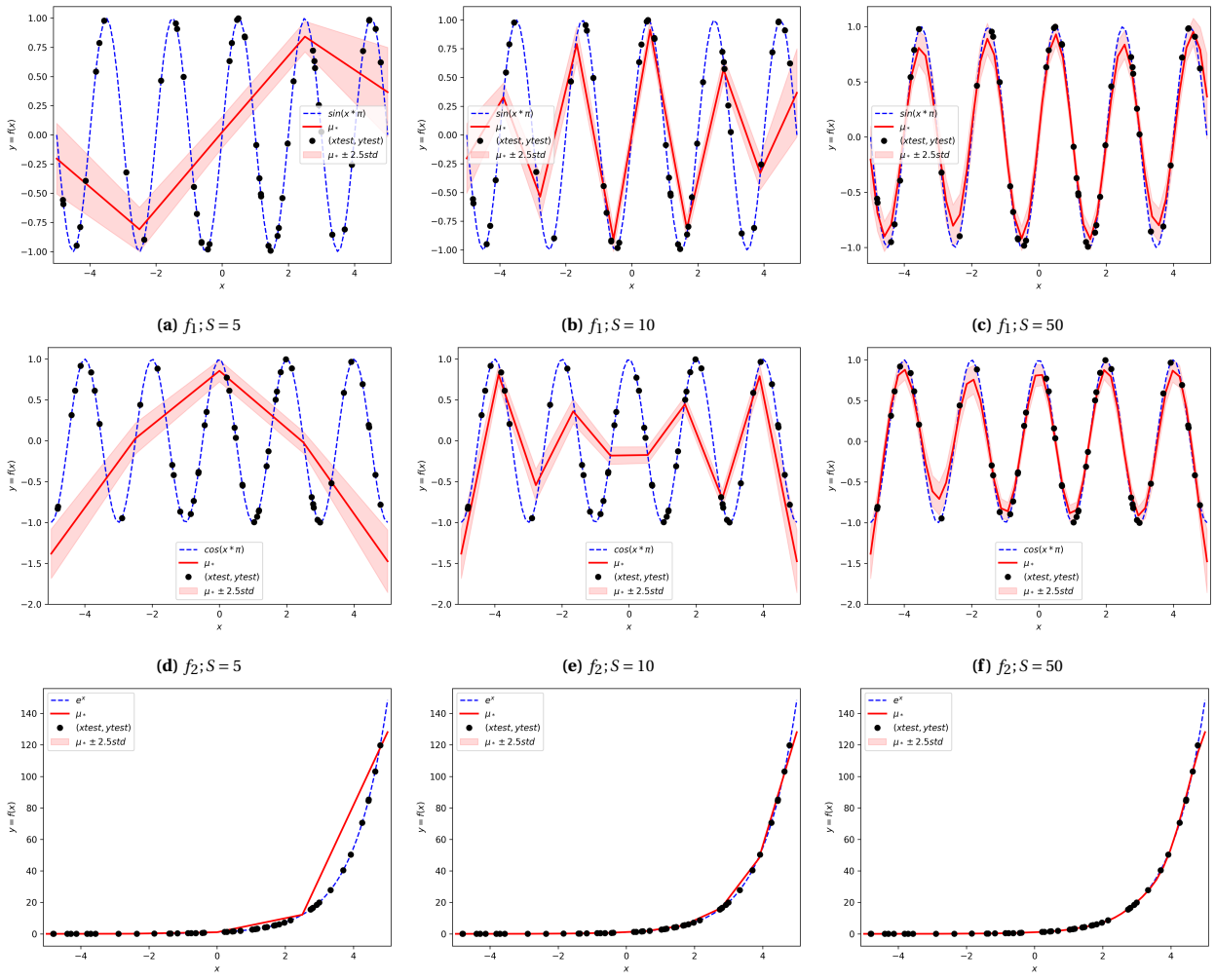


(a) $f_1; S = 5$ **(b)** $f_1; S = 10$ **(c)** $f_1; S = 50$

(d) $f_2; S = 5$ **(e)** $f_2; S = 10$ **(f)** $f_2; S = 50$

(g) $f_3; S = 5$ **(h)** $f_3; S = 10$ **(i)** $f_3; S = 50$

Figure 2. Gaussian Processes with SE kernel with different training set sizes (S). The samples comes from three different functions (sine - f_1, cosine - f_2 and exponential - f_3).



(a) $f_1; S = 5$ **(b)** $f_1; S = 10$ **(c)** $f_1; S = 50$

(d) $f_2; S = 5$ **(e)** $f_2; S = 10$ **(f)** $f_2; S = 50$

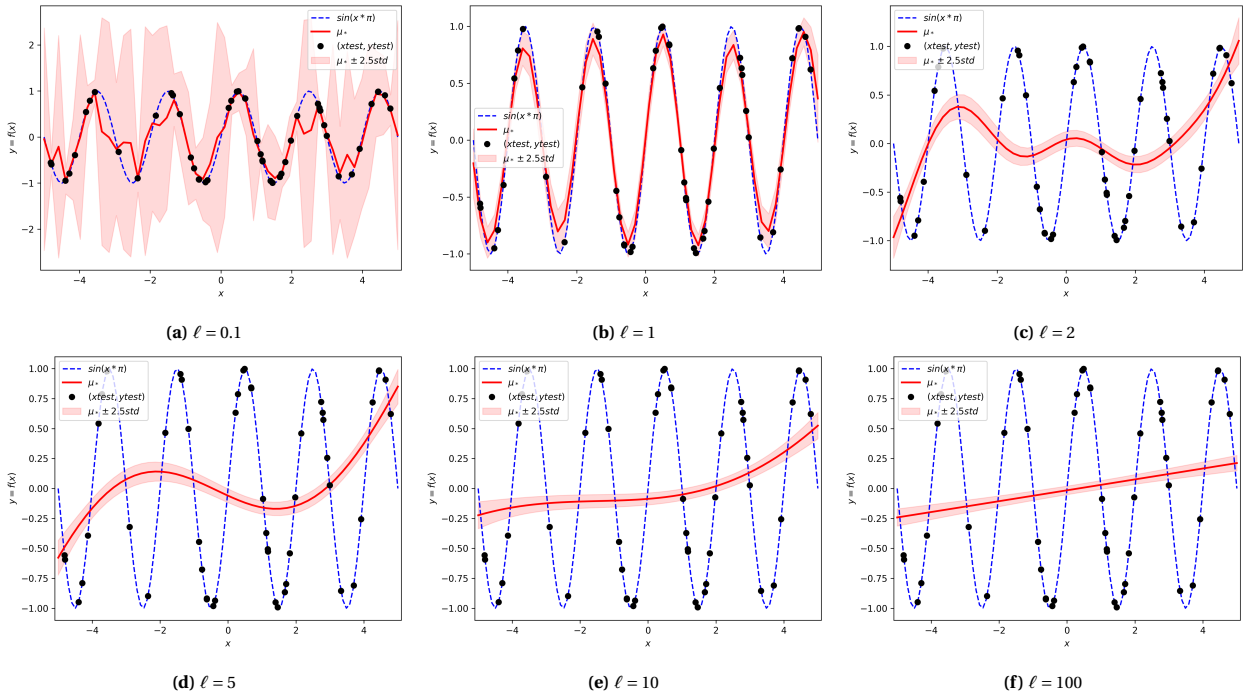
(g) $f_3; S = 5$ **(h)** $f_3; S = 10$ **(i)** $f_3; S = 50$

Figure 3. Gaussian Processes with SE kernel with different test set sizes (S). The samples comes from three different functions (sine - f_1, cosine - f_2 and exponential - f_3).

4.2.1 Parameters ℓ

We evaluated effects of the parameters ℓ of the SE kernel (Eq.(16)) on our predictions, keeping the remaing parameters (σ and σ_y^2) in their default value (Fig. 4).

As known, the parameter ℓ is the horizontal scale over which the target function changes. Based on Fig. 4, we observed that $\ell = 1$ is closer to the optimum value of the parameter. If we decrease its value, the curve becomes more wiggly and the uncertainty also increases (i.e. wider confidence interval), since the effective distance from the training set samples increases more quickly. However, if we increase its value, it becomes smoother until it becomes a straight line.



(a) $\ell = 0.1$ **(b)** $\ell = 1$ **(c)** $\ell = 2$

(d) $\ell = 5$ **(e)** $\ell = 10$ **(f)** $\ell = 100$

Figure 4. Gaussian Processes with SE kernel with different values of ℓ. The remaining parameters of SE kernel are fixed ($\sigma = 1$ and $\sigma_y^2 = 0.01$).

4.2.2 Parameters σ

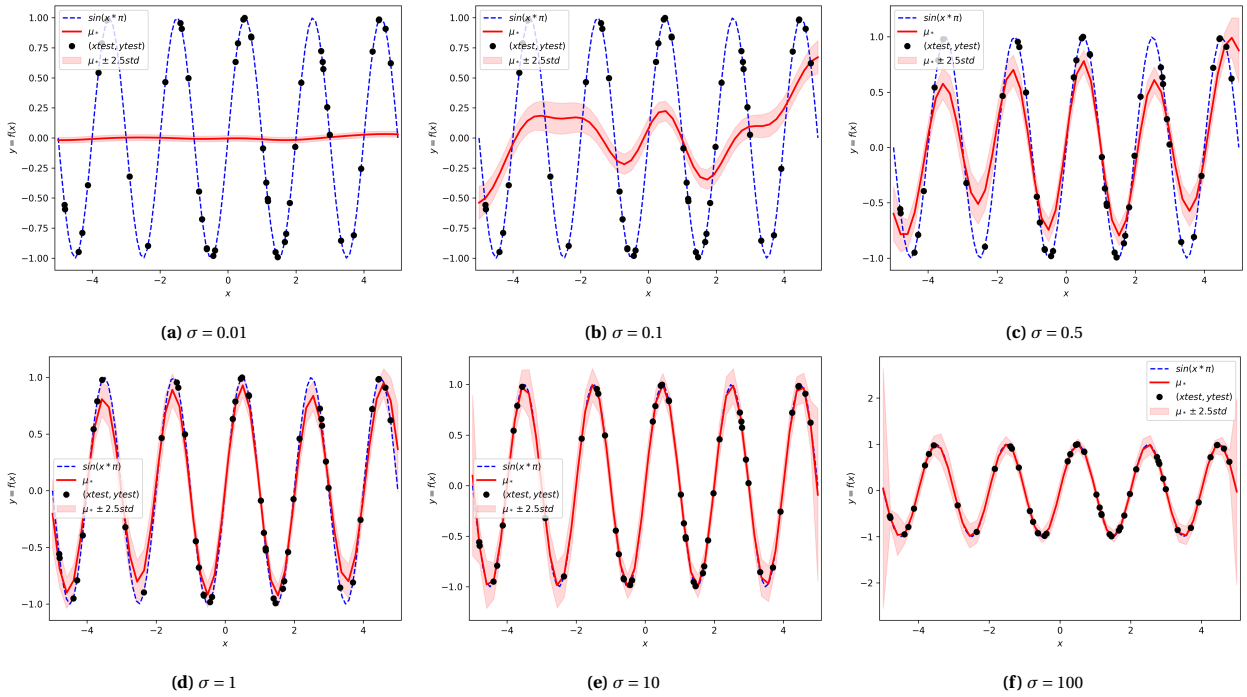
We evaluated effects of the parameters σ of the SE kernel (Eq.(16)) on our predictions, keeping the remaing parameters (ℓ and σ_y^2) in their default value (Fig. 5).

As we know, the parameter σ is the vertical scale of the target function. Based on Fig. 5, we observed that $\sigma = 1$ is closer to the optimum value of the parameter. If we decrease its value, the curve becomes smoother, until it becomes a straight line, and the uncertainty also decreases (i.e. smaller confidence interval). On the other hand, if we increase its value, the regions where there are not training samples had an increase on their uncertainty (i.e. wider confidence interval), this can be seen more clearly in the borders. Even so, there is a smaller increase in regions within the curve.

4.2.3 Parameters σ_y^2

We evaluated effects of the parameters σ_y^2 on our predictions, keeping the remaing parameters (ℓ and σ) in their default value (Fig. 6).

As observed in Fig. 6, when the noise is small, the curve has a good fit to our target function and the uncertainty becomes really small (i.e. small confidence interval). As the noise increase, the curve becomes smoother, until it becomes a straight line, and, at the same time, the uncertainty increases (i.e. wider confidence interval). Therefore, the ideal scenario is to make observations with as little noise as possible for the best results.



(a) $\sigma = 0.01$ **(b)** $\sigma = 0.1$ **(c)** $\sigma = 0.5$

(d) $\sigma = 1$ **(e)** $\sigma = 10$ **(f)** $\sigma = 100$

Figure 5. Gaussian Processes with SE kernel with different values of σ. The remaining parameters of SE kernel are fixed ($\ell = 1$ and $\sigma_y^2 = 0.01$).

4.3 Effect of kernel function

Here, we evaluate effects of different kernel functions on the GP model and on our predictions, without optimizing their hyperparameters that were kept at their default. All experiments in this section used 50 samples in the training and test sets, using the function sine as the target function.

First, we plotted some samples of the priors of each kernel function (SE, OU, RQ and ESS kernels), without noise in the observations, in Fig. 7. When we choose a different kernel function, we are changing the shape of the functions priors, that will be conditioned by our training data with some uncertainty about it. We also plotted some functions sampled from the prior, but with noise in the observations ($\sigma_y^2 = 0.01$), in Fig. 8. Then, the curve becomes more wiggly when the noise is added to the priors. Finally, we note that both SE and RQ kernels are more rounded, and the OU and ESS are less rounded, which is expected for the kernel functions. In addition, we also observe that the ESS kernel is periodic, which is also expected.

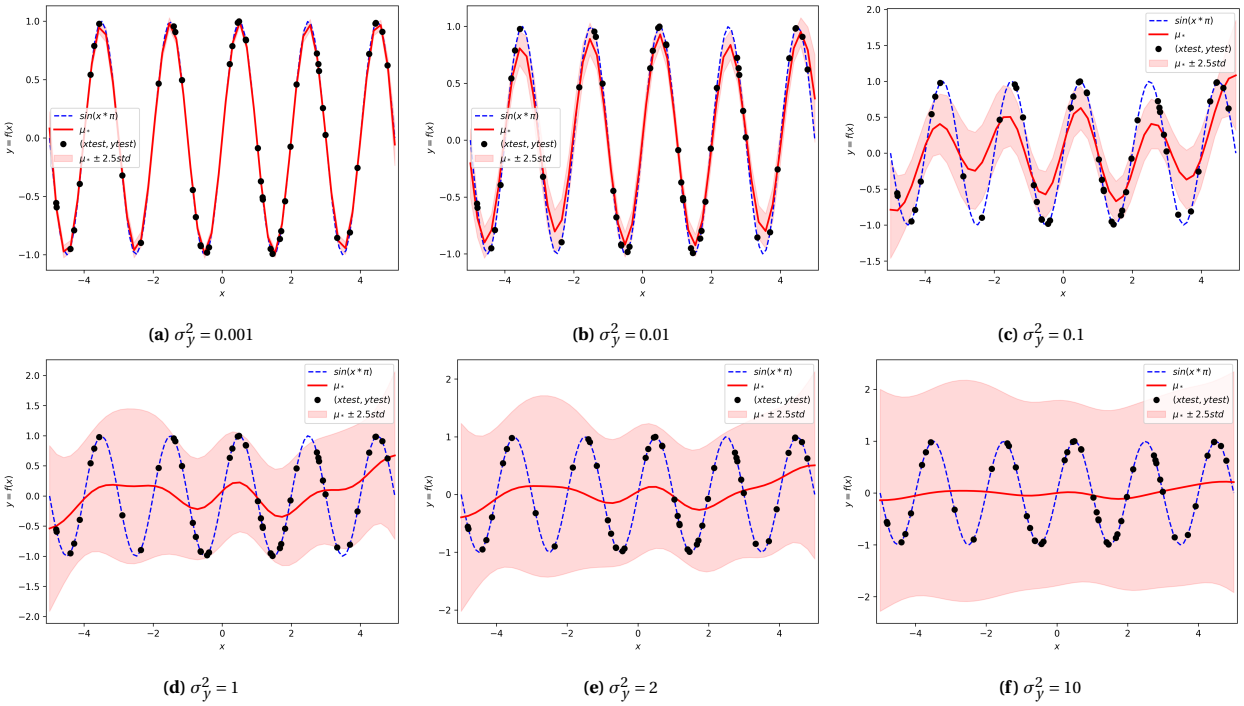
Then, we plotted the GP posteriors with noisy observations ($\sigma_y^2 = 0.01$) in Fig. 9. Based on results, we noted that the RQ kernel (Fig. 9c), followed by the SE kernel (Fig. 9a), presented the best fit to our target function, and the ESS kernel (Fig. 9d) presented the worst fit. However, the hyperparameters were not optimized and therefore, we are not able to define the best kernel function to our target function without further experiments.

Hence, the key ideia of the choice of the kernel function is to determine almost all the generalization properties of the GP model. So, it is an essential step to choose correctly the kernel function to be employed in the problem, as it has a large effect on the results.

4.4 Predictions with higher dimensional data

Previously, we successfully predict functions based on our one-dimensional data. To illustrate our model ability to predict in higher dimensional data, we sampled 100 training points from a 2D periodic function (Eq.(15)), and we used 900 test points to to predict a mean 2D function (Fig. 10).

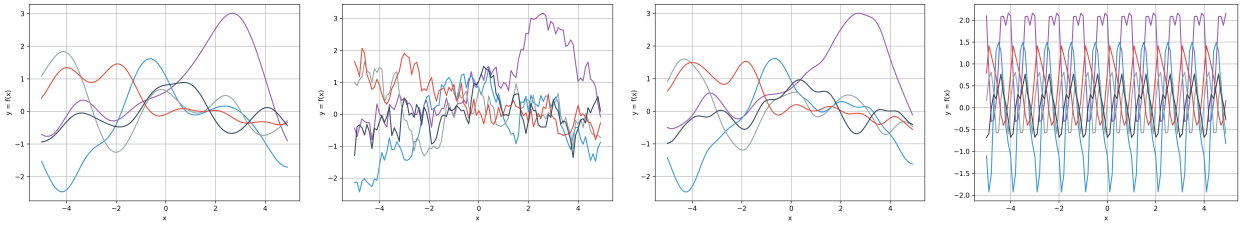
Comparing Fig. 1d and Fig. 10, we notice that our mean function has not a good fit for our target function. It is worth mentioning that we used much more data points in the 2D GP than in the 1D GP, and we had a worse prediction. Then, we suggest that in higher dimensions, the GP is less sensitive to the increase in sample size. Due to limitation on computational resources, we did not perform the prediction with more points in the training and test sets. However, we are able to apply our algorithm in higher dimensional data.



(a) $\sigma_y^2 = 0.001$ **(b)** $\sigma_y^2 = 0.01$ **(c)** $\sigma_y^2 = 0.1$

(d) $\sigma_y^2 = 1$ **(e)** $\sigma_y^2 = 2$ **(f)** $\sigma_y^2 = 10$

Figure 6. Gaussian Processes with SE kernel with different values of σ_y^2. The remaining parameters of SE kernel are fixed ($\ell = 1$ and $\sigma = 1$).



(a) SE kernel **(b)** OU kernel **(c)** RQ kernel **(d)** ESS kernel

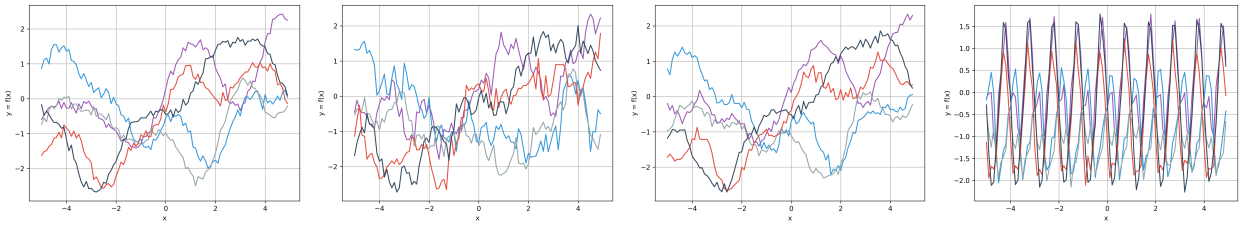
Figure 7. Five function sampled from a GP prior without noise with different kernels.

5 Conclusion

In this project, we sucessfully implemented a Gaussian Process algorithm for regression, and evaluated the effects of the parameters and hyperparameters on the predictions. First, the number of samples on the training and test sets directly affects the quality of the predictions. Next, the hyperparameters (e.g. σ, ℓ and σ_y^2) also affects the quality of the predictions, but each has a different effect on the predicted function. Hence, the hyperparameters should be optimized to achieve the best predictions possible. In addition, different kernel functions define functions with different shapes, that are conditioned by the training data on the GP posterior. Finally, our algorithm can be employed on higher dimensional data.

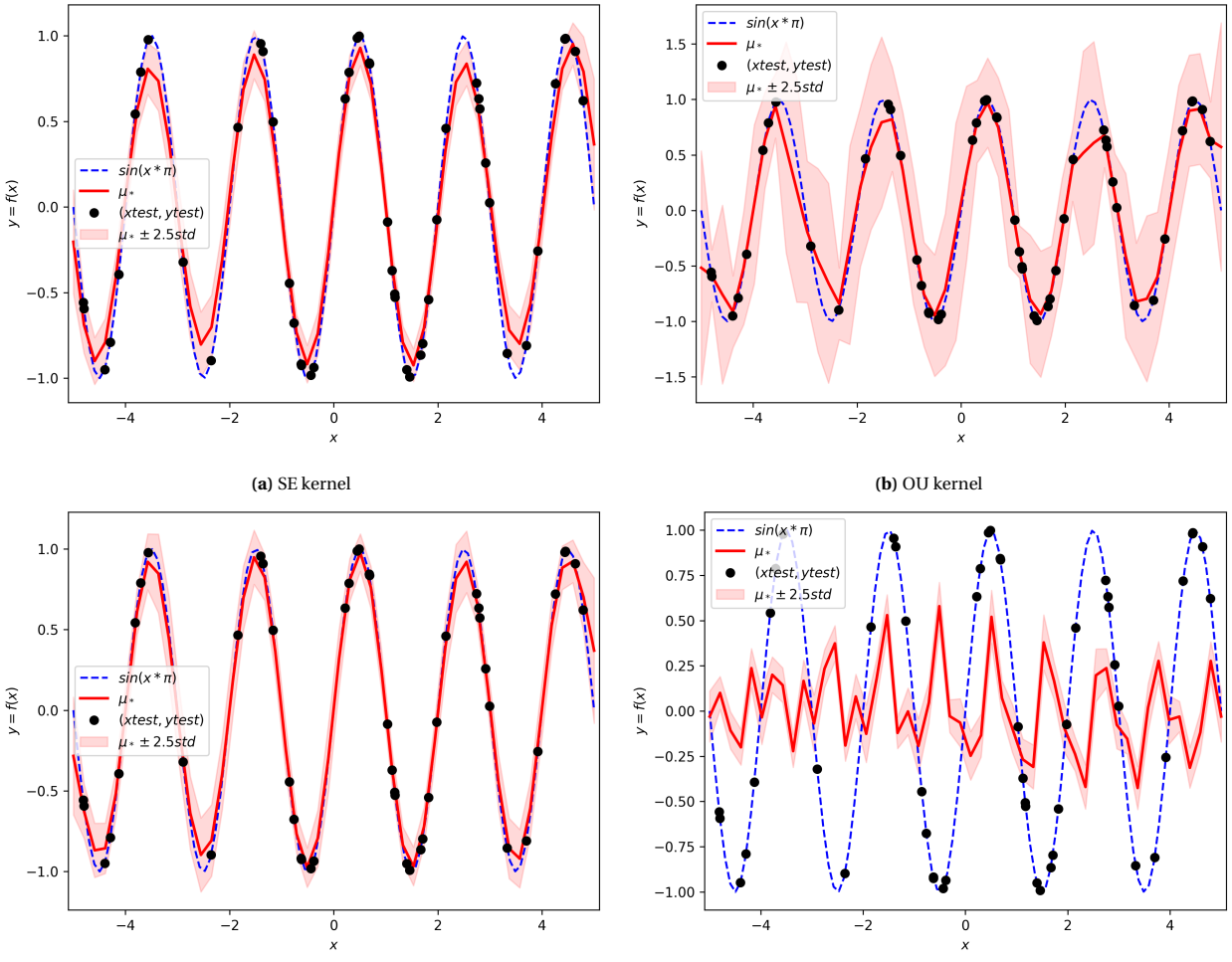
References

[1] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012, pp. 27–33.



(a) SE kernel **(b)** OU kernel **(c)** RQ kernel **(d)** ESS kernel

Figure 8. Five function sampled from a GP prior with noise ($\sigma_y^2 = 0.01$) with different kernels.



(a) SE kernel **(b)** OU kernel

(c) RQ kernel **(d)** ESS kernel

Figure 9. Gaussian Processes with different kernel. The parameters of each kernel are: $\kappa_{SE} = \{\sigma = 1.0, \ell = 1.0\}$, $\kappa_{OU} = \{\sigma = 1.0, \ell = 1.0\}$, $\kappa_{RQ} = \{\sigma = 1.0, \ell = 1.0, \alpha = 1.0\}$ and $\kappa_{ESS} = \{\sigma = 1.0, \ell = 1.0, w = 1.0\}$.

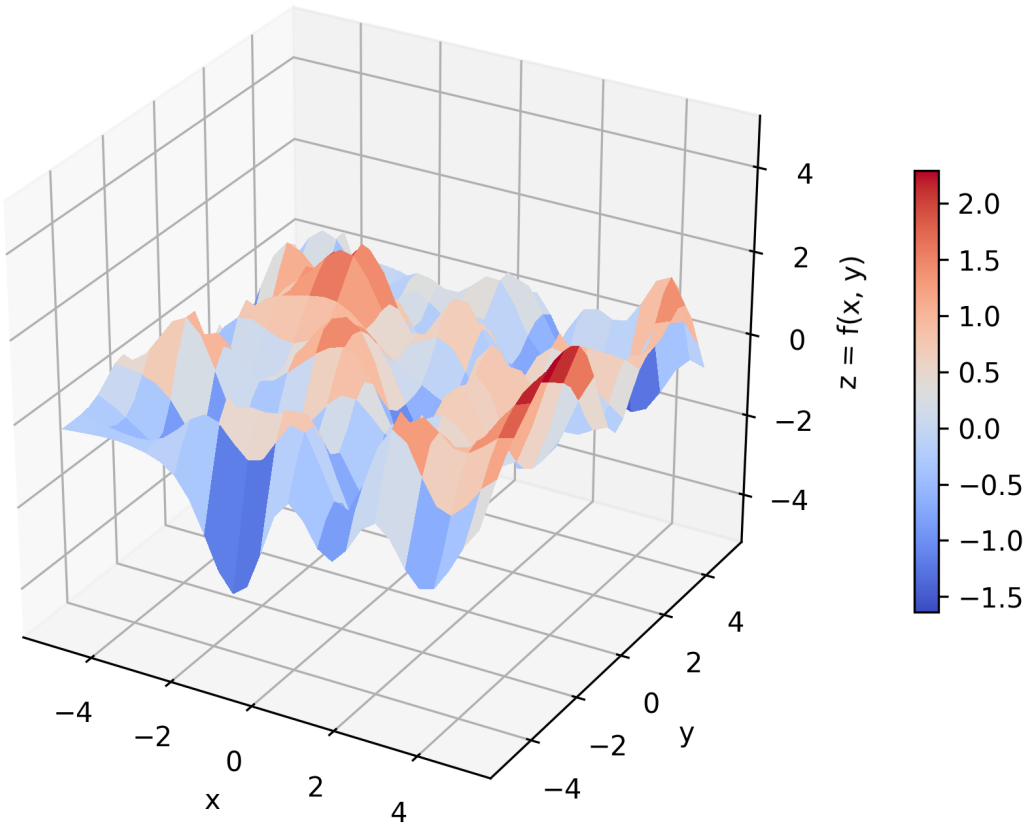


Figure 10. 2D Gaussian Processes with multi-dimensional SE kernel.