Linear and Non-Linear Processes in Machine Learning

Susan Sapkota[1]

[†]Department of Physics, University of New Mexico

***Abstract*—We introduce concept of Gaussian Processes in Linear and Non-linear for prediction, regression and classification. Then we performed experiment on gaussian regression and optimized regression with optimized parameter. We found out with narrow confidence interval the regression is accurate.**

***Index Terms*—Machine Learning, Gaussian Learning, kernal, Marginal Likehood**

I. Introduction

A Gaussian process (GP) is a generalization of gaussian probability distribution over random variables which are scalars or vectors. The random variables and their collection have joint gaussian distribution [2]. Gaussian Process is popular nowadays in machine learning. The Gaussian Process with the use of Bayesian rule provide best guess predictions. In this paper, we compute posterior distribution over model and use it for making prediction on new test point.

II. Theory

A. Bayes Rule

Consider two subset A, B $\in$ universal set (Ω). Bayes rule defines the probability of event A to occur given the probability of the event B given by,

$$P(A|B) = \frac{P(A\cap B)}{P(B)}$$

Also $A\cap B = B\cap A$ then

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where $P(A|B)$ is posterior, P(A) is prior and $P(B|A)$ is likehood. Let us consider D training set of n observation as $D = [(\mathbf{x}_i, y_i) : i \in \mathbb{N}]$. Then we can define linear regression with gaussian noise as $y = \mathbf{x}^T\mathbf{w} + \epsilon$ where $\mathbf{x}$ is input vector, $\mathbf{w}$ is parameter and ϵ is gaussian bias and function is linear model. The bias is defined as $\epsilon \sim N(0, \sigma_n^2)$ with zero mean and standard deviation as σ_n. Then we can define the likehood with probability density of observation with given parameter as,

$$\begin{aligned} p(\mathbf{y}|\mathbf{x},\mathbf{w}) &= \prod_{i=1}^{n} p(\mathbf{y}_i|x_i,\mathbf{w}) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - x_i^T\mathbf{w})^2}{2\sigma_n^2}\right) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}|\mathbf{y} - X^T\mathbf{w}|^2\right) \\ &= N(X^T\mathbf{w}, \sigma_n^2 I) \end{aligned}$$

where X is matrix. Now we can define probability of x_i and y_i with parameter $\mathbf{w}$ is given by,

$$p(\mathbf{w}|y_i,\mathbf{x}_i) = \frac{p(y_i|\mathbf{x}_i,\mathbf{w})p(\mathbf{w})}{p(y_i|\mathbf{x}_i)}$$

where $p(\mathbf{w}|y_i,\mathbf{w})$ is posterior, $p(y_i|\mathbf{x}_i,\mathbf{w})$ is likehood and p($\mathbf{w}$) is prior. Finally,

$$p(\mathbf{w}|y_i,\mathbf{x}_i) \propto p(y_i|\mathbf{x}_i,\mathbf{w})p(\mathbf{w})$$

The above expression is used in the linear regression section B.

B. Linear Regression

Let us consider linear estimator with bias as $y_i = \mathbf{x}_i^T\mathbf{w} + \epsilon_i$ where ϵ is additive white gaussian noise.We can now show the distribution of the noise as

$$p(\epsilon_i) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{\epsilon_i^2}{2\sigma_n^2}\right)$$

Then we have $\epsilon_i = y_i - \mathbf{x}_i^T\mathbf{w}$. we can write above equation as

$$p(y_i[n]|\mathbf{x}_i[n],\mathbf{w}) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{|y_i[n] - \mathbf{x}_i[n]^T\mathbf{w}|^2}{2\sigma_n^2}\right)$$

Assume nature of ϵ, y_i is independent then

$$\begin{aligned} p(\mathbf{y}|\mathbf{X,w}) &= \prod_{n=1}^{N} p(y_i[n]|\mathbf{x}_i[n],\mathbf{w}) \\ &= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{|y[n] - \mathbf{x}[n]^T\mathbf{w}|^2}{2\sigma_n^2}\right) \\ &= \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left(-\frac{|\mathbf{y} - \mathbf{X}^T\mathbf{w}|^2}{2\sigma_n^2}\right) \end{aligned}$$

Here we can assume now w obeys Central Limit Theorem such that $p(\mathbf{w}) \sim N(0, \sum_p)$ where $\sum_p$ is covariance of process. Now we can assume p($\mathbf{w}$) as

$$p(\mathbf{w}) = \frac{1}{(2\pi\Sigma_p)^{(D+1)/2}} \exp\left(-\frac{1}{2}\mathbf{w}^T\Sigma_p^{-1}\mathbf{w}\right) \tag{1}$$

One can write posterior $p(\mathbf{w}|\mathbf{y},\mathbf{X})$ with help of likelihood and prior to maximize the parameter as

$$p(\mathbf{w}|\mathbf{y},\mathbf{X}) \propto \exp\left(-\frac{|\mathbf{y} - \mathbf{X}^T\mathbf{w}|^2}{2\sigma_n^2}\right) \exp\left(-\frac{1}{2}\mathbf{w}^T\Sigma_p^{-1}\mathbf{w}\right)$$

The above expression is gaussian since it is product of gaussian function. we can rearrange the above expression after ignoring $\frac{1}{2}$ as follows:-

$$\frac{|\mathbf{y}-\mathbf{X}^T\mathbf{w}|^2}{\sigma_n^2}+\mathbf{w}^T\Sigma_p^{-1}\mathbf{w}=\sigma_n^{-2}\mathbf{y}^T\mathbf{y}-2\sigma_n^{-2}\mathbf{y}^T\mathbf{X}^T\mathbf{w}+\mathbf{w}^T\mathbf{A}\mathbf{w}$$

where $\mathbf{A}=\sigma_n^{-2}\mathbf{X}^T\mathbf{X}+\Sigma_p^{-1}$. Then we can express the exponent part as

$$-\frac{1}{2}(\mathbf{w}-\overline{\mathbf{w}})^T\mathbf{A}(\mathbf{w}-\overline{\mathbf{w}}) \tag{2}$$

where $\mathbf{A}^{-1}$ is covariance matrix and $\overline{\mathbf{w}}$ is mean. Ignoring $\frac{-1}{2}$ and combining equation 2 and above expression we get,

$$\begin{aligned}\sigma_n^{-2}\mathbf{y}^T\mathbf{y}-2\sigma_n^{-2}\mathbf{y}^T\mathbf{X}^T\mathbf{w}+\mathbf{w}^T\mathbf{A}\mathbf{w}&=(\mathbf{w}-\overline{\mathbf{w}})^T\mathbf{A}(\mathbf{w}-\overline{\mathbf{w}})\\&=\mathbf{w}^T\mathbf{A}\mathbf{w}-2\overline{\mathbf{w}}^T\mathbf{A}\mathbf{w}+\overline{\mathbf{w}}^T\mathbf{A}\overline{\mathbf{w}}\end{aligned}$$

Let $\sigma_n^{-2}\mathbf{y}^T\mathbf{X}^T=\overline{\mathbf{w}}^T\mathbf{A}$ and then we have $\overline{\mathbf{w}}=\sigma_n^{-2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$ satisfying the $\overline{\mathbf{w}}^T\mathbf{A}\overline{\mathbf{w}}=\sigma_n^{-2}\mathbf{y}\ \mathbf{y}$. In short,

$$p(\mathbf{w}|\mathbf{y},\mathbf{X})\propto\exp\left(-\frac{1}{2}(\mathbf{w}-\overline{\mathbf{w}})^T\mathbf{A}(\mathbf{w}-\overline{\mathbf{w}}\right)$$

Then we have $\mathbf{A}=\sigma_n^{-2}\mathbf{X}\mathbf{X}^T$,$\overline{\mathbf{w}}=\sigma_n^{-2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$ and turns out $\overline{\mathbf{w}}=(\mathbf{X}\mathbf{X}^T+\sigma_n^2\Sigma_p^{-1})^{-1}\mathbf{X}\mathbf{y}$. $p(\mathbf{w}|\mathbf{y},\mathbf{X})$ has maximum at $\mathbf{w}=\overline{\mathbf{w}}$. Now consider we have sample $\mathbf{x}^*\notin\mathbf{X}$ and y_i. Then we can compute likelihood of estimator f_* where $f_*=\mathbf{w}^T\mathbf{x}^*$ as $p(f_*|\mathbf{x},\mathbf{w})$ using Total Probability Theorem as

$$p(f_*|\mathbf{x}^*,\mathbf{y},\mathbf{X})=\int_{\mathbf{w}}p(f_*|\mathbf{x}^*,\mathbf{w})p(\mathbf{w}|\mathbf{y},\mathbf{X})dw$$

After solving integral, we get

$$p(\overline{f}_*|\mathbf{x}^*,\mathbf{y},\mathbf{X})\sim N(\overline{\mathbf{w}}^T\mathbf{x}^*,\mathbf{x}^{*T}\mathbf{A}^{-1}\mathbf{x}^*) \tag{3}$$

The advantage of Gaussian Process over both MMSE and Ridge regression is distribution of the prediction. One can check accurateness of prediction by checking variance $\sigma_{f_*}^2=\mathbf{x}^{*T}\mathbf{A}^{-1}\mathbf{x}^*$ of output.

C. Non linear Regression with Kernal Trick

We can extend Gaussian Process as non linear. Let us consider a nonlinear transformation into a Hilbert space $\psi(\mathbf{x})$ then inverse of covariance matrix $\mathbf{A}$ with linear parameter takes form of $\mathbf{A}=\sigma_n^{-2}\Phi\Phi^T+\Sigma_p^{-1}$ and $\overline{\mathbf{w}}=\left(\Phi\Phi^T\sigma_n^2\Sigma_p^{-1}\right)^{-1}\Phi\mathbf{y}$. $\Phi\Phi^T$ is infinite dimension matrix. we can apply Representer Theorem in transformation $\psi(\mathbf{x}_i)\rightarrow\Sigma_p^{1/2}\psi(\mathbf{x}_i)$ given by,

$$\overline{f}_*=\psi(\mathbf{x}^*)^T\mathbf{w}=\psi(\mathbf{x}^*)^T\Sigma_p\Phi_\alpha \tag{4}$$

We can find the expression for the dual parameter after plugging $\overline{w}$ in $\mathbf{w}$ with $\Sigma_p=\Sigma_p^{1/2}\Sigma_p^{1/2}$ where Σ_p is positive definite matrix. The dual parameter is

$$\alpha=\left(\Phi^T\Sigma_p\Phi+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y}=\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y} \tag{5}$$

Then we have $\mathbf{K}=\Phi^T\Sigma_p\Phi$ and its component $k_{i,j}(\mathbf{x}_i,\mathbf{z}_i)=\psi(\mathbf{x}_i)^T\Sigma_p\psi(\mathbf{z}_i)=\psi(\mathbf{x}_i)^T\Sigma_p^{1/2}\Sigma_p^{1/2}\psi(\mathbf{z}_i)$. The choice of kernal matrix decides covariance matrix. We can say kernal matrix is covariance matrix.

$$\begin{aligned}\overline{f}_*&=\psi(\mathbf{x}^*)^T\Sigma_p\Phi\alpha\\&=\mathbf{k}(\mathbf{x}^*)^T\alpha\\&=\mathbf{k}(\mathbf{x}^*)^T\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\end{aligned}$$

where $\mathbf{k}(\mathbf{x}^*)=<k(\mathbf{x}^*,\mathbf{x}[1],....k(\mathbf{x}^*,\mathbf{x}[N])>^T$ and covariance matrix $\mathbf{A}^{-1}=\Sigma_p-\Sigma_p\Phi(\mathbf{K}+\sigma_n^2\mathbf{I})^{-1}\Phi^T\Sigma_p$ from inversion matrix lemma.

$$\begin{aligned}\sigma_{f_*}&=\psi(\mathbf{x}^*)^T\mathbf{A}^{-1}\psi(\mathbf{x}^*)\\&=\psi(\mathbf{x}^*)^T\Sigma_p\psi(\mathbf{x}^*)-\psi(\mathbf{x}^*)^T\Sigma_p\Phi\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\Phi^T\Sigma_p\psi(\mathbf{x}^*)\\&=k(\mathbf{x}^*,\mathbf{x}^*)-\mathbf{k}(\mathbf{x}^*)^T\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}(\mathbf{x}^*)\end{aligned}$$

Now we can show kernal Matrix $\mathbf{K}$ is the covariance matrix of the Gaussian Process. we have variance $\sigma=E[x^2]-(E[x])^2$ where E is the expectation. we assume the process have zero mean and covariance matrix of estimation is $\mathbf{C}_{ff}=E(\mathbf{f}.\mathbf{f}^T)$. Then we have entries of the matrix as

$$\begin{aligned}(\mathbf{C}_{ff})_{i,j}=E[f(\mathbf{x}_i).f(\mathbf{x}_j)]&=E[\psi(\mathbf{x}_i)^T\mathbf{w}\mathbf{w}^T\psi(\mathbf{x}_j)]\\&=\psi(\mathbf{x}_i)^TE[\mathbf{w}\mathbf{w}^T]\psi(\mathbf{x}_j)=\psi(\mathbf{x}_i)^T\Sigma_p\psi(\mathbf{x}_j)\\&=k(\mathbf{x}_i,\mathbf{x}_j)\end{aligned}$$

Here $\mathbf{C}_{ff}=E(\mathbf{f}.\mathbf{f}^T)=\mathbf{K}$ which is equivalent to kernel matrix.

III. Experiments and Results

A. GPML toolbox

We used GPML toolbox in matlab from [1] to perform Gaussian Process analysis. *gp* function is used for training and testing data which does posterior inference, learns hyperparameters, and evaluate prediction from marginal likelihood. The function takes hyperparameters, inference method, mean function, covariance function, likelihood, training inputs, training targets, test cases for the input. The function has two modes as training and predication. If we do not declares the modes then function compute negative log marginal likelihood and its partial derivatives with parameter is evaluated. we initialized hyperparameters for respective likelihood, mean and covariance function. One can get optimal hyperparameter struction using minimize function.

B. Linear Gaussian for prediction

We used $w[n]$ with 100 samples $(1\leq n\leq 100)$ of Gaussian noise with mean as zero and variance as 1 from normal random number generator. Our input is $x[n]$ where n=1,2,,,,100. We trained our Gaussian Process model using above input and label just using 20 samples. After training our model, we used 100 samples of data for the predictive probabilities. Finally, we choose hyperparameters with mean fucntion, covariance function and likelihood function to obtain below prediction.

We used Auto Regressive Moving Average(ARMA) process to generate targets labels with moving average coefficients $b=$

$[0.0048; 0.0193; 0.0289; 0.0193; 0.0048]^T$ and auto regression coefficient $a = [1; -2.3695; 2.3140; -1.0547; 0.1874]^T$. Now we construct a linear Gaussian process model shown in figure 1 and 2.

C. *Linear ARMA with AR(1) noise*

We added AR(1) process noise to the above target labels with coefficient $a_n = 0.2$ to the output. Now we have, $y[n] = f(n) + g[n]$ where $g[n] = w_g[n] + 0.2g[n-1]$ where $w_g[n]$ is gaussian white noise with mean zero and variance $\sigma = 0.1$. Then we can see the model matches with process $f[n]$. In other words, we have $f[n] = a^T f[n-1] + b^T w[n1] = w^T x[n]$ where $f[n-1]$ and $w[n-1]$ are vector with sample n-1 to n-5 terms of input and output process. Then we can evaluate theoretical value of the output covariance as

$$\begin{aligned} \mathbf{K} &= E[y^T[i]y[j]] \\ &= E[(f[i] + g[i])^T f[j] + g[j]] \\ &= E[f^T[i]f[j] + f^T[i]g[j] + g^T[i]f[j] + g^T[i]g[j]] \end{aligned}$$

We see that the signal and noise are independent.So, our above expression becomes,

$$\mathbf{K} = E[f^t[i]f[j]] + E[g^t[i]g[j]] \tag{6}$$

We have all the equipment to make the model with coefficient of the ARMA model. so,we constructed linear gaussian process regression based on this model including AR(1) which is given in figure 3.

D. *Non-Linear Gaussian Process*

We used non-linear model with mean function hyperparameter and covariance function to construct the model similar to above. We made model with AR(1) process with noise and another with just white noise shown in coresponding figure 4 and 5.

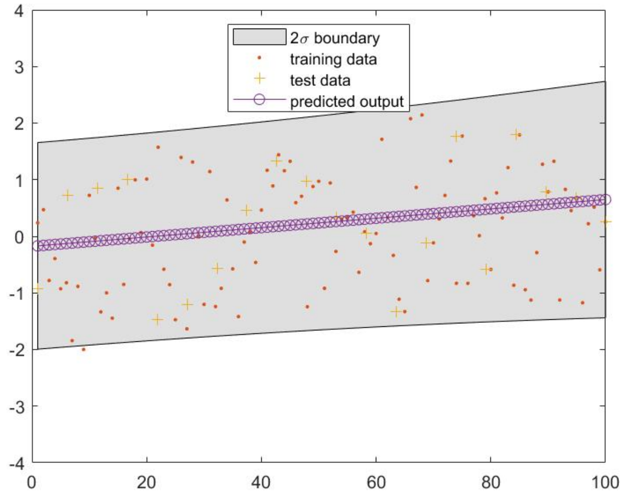

Fig. 1. Linear Gaussian Process Prediction

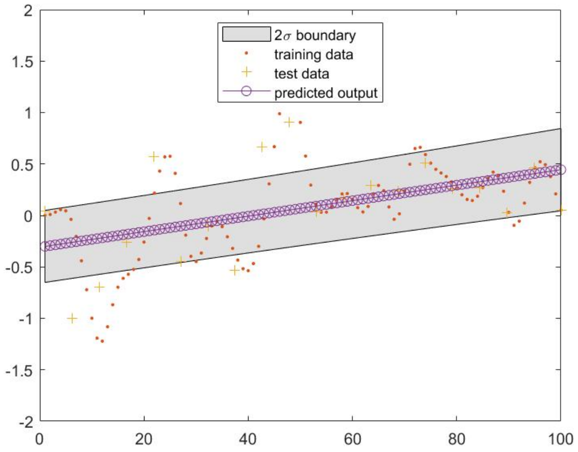

Fig. 2. Linear Gaussian Process Prediction with filter

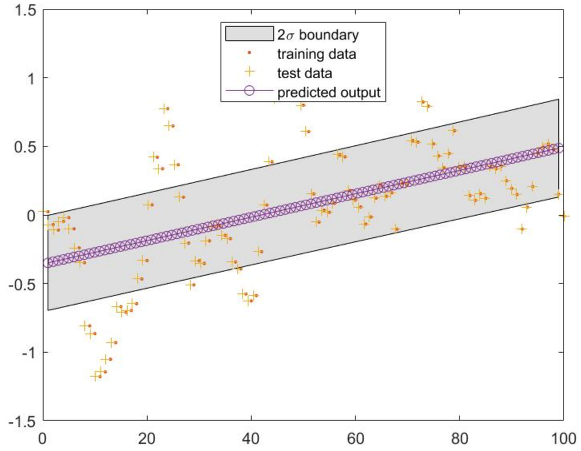

Fig. 3. Linear Gaussian Process Prediction with ARMA and filter

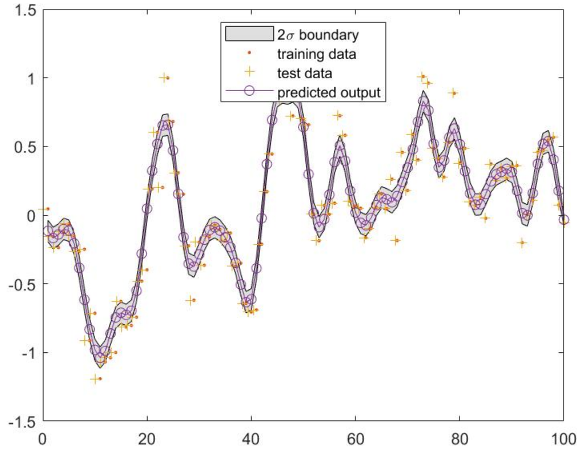

Fig. 4. Non-Linear Gaussian Process with white noise and ARMA

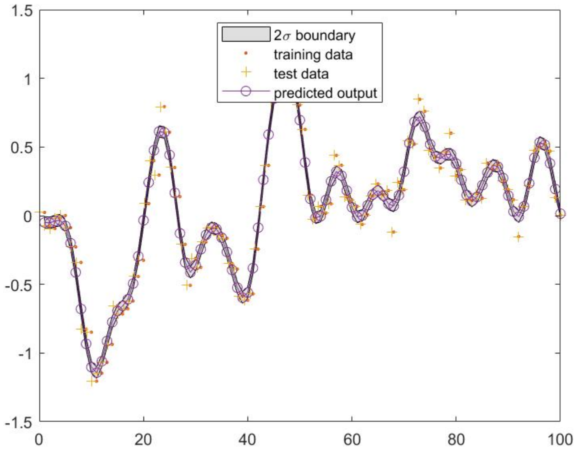

Fig. 5. Non-Linear Gaussian Process with white noise

IV. Discussion

Figure 1 and 2 show the Linear Gaussian prediction without filter and with filter. Similarly, Figure 3 show the linear Gaussian prediction with ARMA noise and filter. Some of the points are poorly classified and in figure 3 some of the points are outside of the boundary. The linear classifier is poor in that case with low confidence level. Figure 4 and 5 shows the case for non-linear gaussian prediction with ARMA noise and white noise and with just white noise. we can clearly see that the confidence level for the prediction is higher as test data is closed with the training data. Without ARMA noise, we see the test data is much more closer to training data in figure-5.

V. Conclusion

In this paper, we developed theoretical framework of linear and non-linear gaussian procession in short. We performed experiment with linear and non-linear gaussian process with optimized the hyper-parameters of the covariance function and used it for regression. we saw that the non linear regression is more accurate than linear regression.

References

[1] C. E. Rasmussen and H. Nickisch, “Gaussian processes for machine learning (gpml) toolbox,” Journal of machine learning research, vol. 11, no. Nov, pp. 3011–3015, 2010..

[2] Wiener, Norbert, and Interpolation Extrapolation. ”Smoothing of stationary time series.” (1949)