On Gaussian Processes for Regression

Jeffrey Alido
Department of Electrical and Computer Engineering
Boston University
jalido@bu.edu

Shashank Manjunath
Department of Electrical and Computer Engineering
Boston University
manjuns@bu.edu

April 27, 2021

Introduction

- Gaussian processes are a class of machine learning models that allow us to easily incorporate prior observations into our data.
- Example: predicting temperatures throughout a room.
  - Suppose you are trying to determine the temperature at a certain point in a room, $\boldsymbol{x}_{n+1}$
  - You know the temperatures at points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
  - If $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}_{n+1}\}$ are close, the temperatures at these points will be highly correlated
  - If they are far apart, the temperatures will be less correlated.
  - We can model the n known points as a multivariate Gaussian distribution with the covariance of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ dependent on the physical distance between the two points, then use our distribution to predict the temperature at $\boldsymbol{x}_{n+1}$

Introduction

- In a GP, we assume any new points we observe follow the same multivariate normal observed in our training data
- Some history of GPs [1]
  - Blight and Ott first introducted GPs as priors over functions in 1975[2]
  - Gaussian Process models were first recognized as the limit of a Bayesian neural network by Mackay (1992) and Neal (1996)[2]
- GPs are non-parametric models (unlike models such as Neural Networks)[3]
- GPs allows us to quantify uncertainty in our predictions
- GPs are not advantageous in that they scale poorly to large datasets

Multivariate Gaussian Distributions

- A set of univariate Gaussian random variables may be characterized jointly as a multivariate Gaussian distribution, with joint probability distribution fully characterized by a mean vector and a covariance matrix:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

  - $\boldsymbol{\mu} \in \mathbb{R}^n$ indicates the mean vector
  - $\Sigma \in \mathbb{R}^{n \times n}$ indicates the covariance matrix whose entries describe the covariance between each pair of random variables

Gaussian Processes

- A Gaussian process $f(\boldsymbol{x})$ is defined as a a random process where each set of random variable in the random process is has a multivariate Gaussian distribution.
- In mathematical notation, $f(\boldsymbol{x})$ is fully characterized by a mean function $m(\boldsymbol{x})$ and covariance function, $K(\boldsymbol{x}, \boldsymbol{x'})$:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x'}))$$

$$K(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))]$$

- The mean function $m(\boldsymbol{x})$ is typically defined as zero
- The covariance is chosen based on some prior belief about the dataset
- The covariance function is analagous to a kernel function $\kappa(\cdot, \cdot)$, where each entry of the covariance matrix is the kernel function calculated between the corresponding points

Gaussian Processes for Regression

- While we have defined a Gaussian process, we now describe how to fit a Gaussian process (predictive distribution) given a training set (prior distribution) and test points
- Suppose we observe training data $\boldsymbol{x}$, test data $\boldsymbol{x}'$, and choose kernel κ. Then the mean and covariance functions are given by

$$m(\boldsymbol{x}) = \kappa(\boldsymbol{x}, \boldsymbol{x}')^\top \left(\kappa(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 I\right)^{-1} \boldsymbol{x}$$

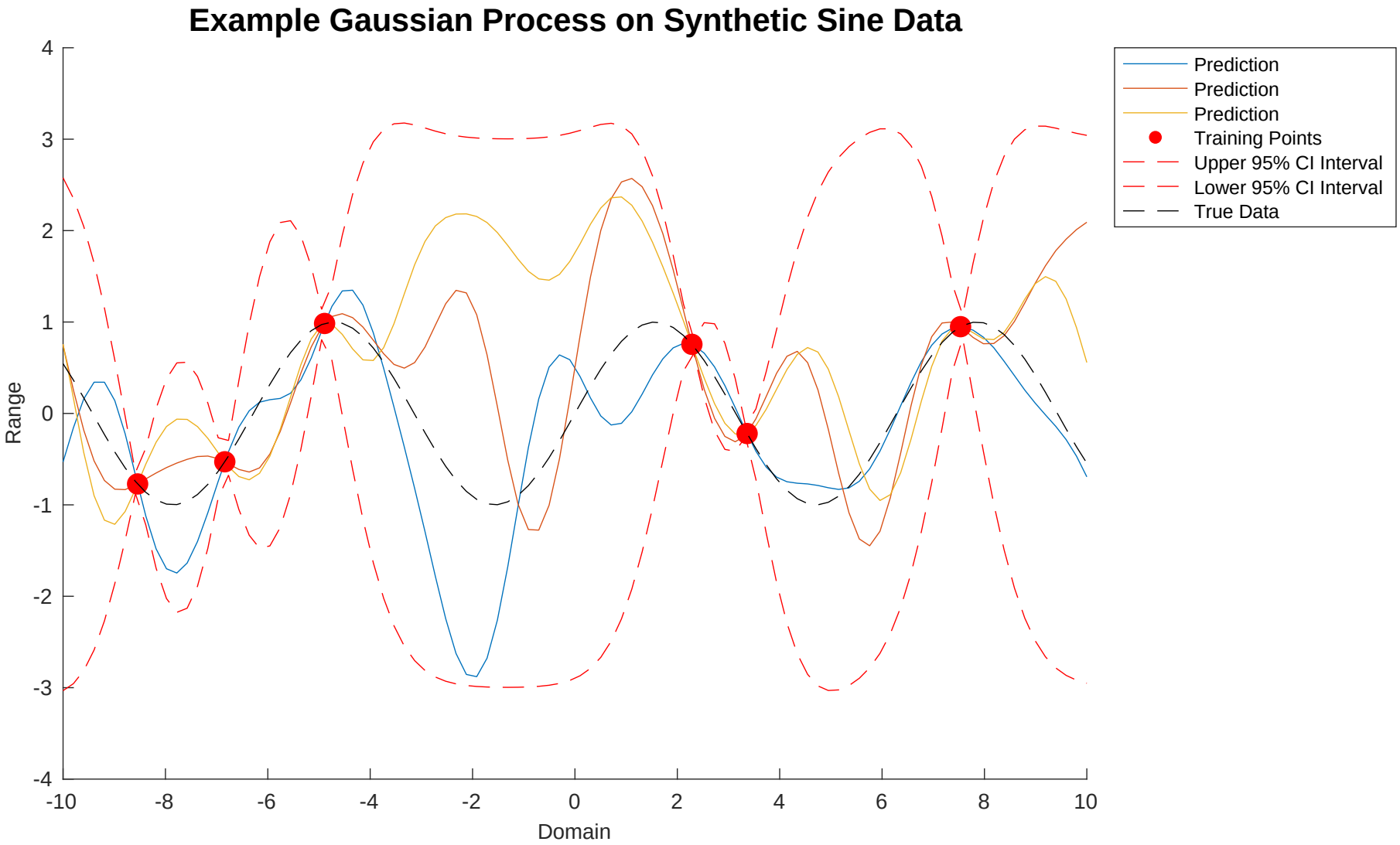
$$K(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}', \boldsymbol{x}') - \kappa(\boldsymbol{x}, \boldsymbol{x}')^\top \left(\kappa(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 I\right)^{-1} \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

Those interested in a full derivation of the results are encouraged to consult section 2 of [1].

Kernels

- Gaussian processes use kernels to help project input features into alternate feature spaces[4]
    - We test out 6 different kernels, and use 5-fold cross validation grid search to find the best hyperparameters
- We can additionally combine simple kernels to create more complicated kernels
    - We test out the Local Periodic Kernel, which is a combination of the Squared Exponential and Periodic kernels
- Kernel choice is up to you - you know the problem best!

A Simple Demonstration



The Boston Housing Dataset

- Originally published in 1978[5]
- 506 data points, 13 features, 1 label (median value of a house in a Boston suburb, in $1000s)
- Well-suited to Gaussian processes due to small size
- Features detailed in Table 1

Table 1: Table of Boston Housing Dataset feature names and features

Feature Name	Feature Description
CRIM	Per capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centres
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of Black people by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

Implementation Details

- We implemented grid search for our hyperparameters
- We normalize our features, which leads to improved algorithm performance, using the following formula:

$$X_{\text{feat}} = \frac{X_{\text{feat}} - \mu(X_{\text{feat}})}{\sigma(X_{\text{feat}})}$$

- We also normalize our label, then convert back to given units (value in \$1000s) after fitting the GP.
- We use the RMSE metric:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and N is the number of samples

Results

- All values presented are RMSE values
- We present Regression SVM (trained with cross-validation hyperparameter search) test error values and Linear Regression test error values for comparison
- Linear Regression: **4.751**

Table 2: Table of Boston Housing Dataset feature names and features

	Gaussian Process Regressor	**SVM Regressor**	Linear Regression
Linear Kernel	**4.751**	4.935	**4.751**
Square Exp Kernel	**3.586**	9.014	4.751
Rational Quadratic Kernel	**3.560**	9.013	4.751
Periodic Kernel	8.944	9.017	**4.751**
Local Periodic Kernel	5.065	9.014	**4.751**
Polynomial Kernel	**4.213**	80.389	4.751

References

- C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. **Adaptive computation and machine learning, Cambridge, Mass: MIT Press, 2006.** OCLC: ocm61285753.
- M. Kuss, *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*. **PhD Thesis, Biologische Kybernetik, Mar. 2006.** Backup Publisher: Technische Universität Darmstadt, Darmstadt, Germany.
- C. M. Bishop, *Pattern Recognition and Machine Learning*. **Springer, 2006.**
- D. Duvenaud, *Automatic Model Construction with Gaussian Processes*. **PhD Thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.**
- D. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of Environmental Economics and Management*, vol. 5, no. 1, pp. 81–102, 1978.

Code for MATLAB implementation, LaTeX for presentation and paper can be found on Github at https://github.com/shashankmanjunath/GaussianProcessRegression