Gaussian Processes for Statistical Learning

Connor McColgan

November 24, 2020

Abstract

Recap of first semester progress and a plan of direction for the second semester.

1 Introduction and Overview

Many regression problems require assessment of unknown functions that map inputs to outputs. In many cases, the shape of the underlying function is unknown; the function is hard to evaluate analytically; or complicated information acquisition. Gaussian processes offer a way of performing inference on the underlying function both passively as well as actively. Gaussian process regression is a non-parametric Bayesian approach. Informally, one could think of the Gaussian process as defining a distribution over functions, and inference taking place directly in the space of functions, the function-space view. Explicitly, if we pick any two or more points in the function, observations of these outputs follow a joint multivariate Gaussian distribution.

1.1 Gaussian process definition

Formally, we can define a Gaussian process in the function-space view as the collection of random variables, any finite of which have a joint Gaussian distribution. Exploring this definition, we see that the GP can be completely specified by its covariance and mean functions. We define the mean function $m(\boldsymbol{x})$ and the covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$ of a real process $f(\boldsymbol{x})$ by:

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})], \tag{1}$$

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x'}) - m(\boldsymbol{x'}))], \tag{2}$$

and so we can express the Gaussian process as such

$$f(\boldsymbol{x}) \sim \mathcal{N}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})). \tag{3}$$

The definition gives rise to the marginalisation property if the covariance function specifies entries in the covariance matrix. This property means that examination of larger set of variables does not change the distribution of the smaller set.

1.2 Covariances

Note, that Gaussian processes are often defined with mean function of zero. In this formulation the process is completely defined by its second order statistic, the covariance function. The covariance function can be thought of as encoding the assumptions about the function we wish to learn. The covariance matrix K is constructed from the covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$:

$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4}$$

We are free to choose the covariance function, so long as the covariance matrix produced are symmetric and positive semidefinite.

1.2.1 The Squared Exponential (SE) covariance function

One of the most common choices for covariance function is the SE kernel. It provides very smooth sample functions, which are infinitely differentiable. It is given in the form

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left\{ -\frac{1}{2l^2} (\|\boldsymbol{x} - \boldsymbol{x}'\|)^2 \right\}. \tag{5}$$

The SE covariance is stationary and more specifically isotropic.

The two free parameters, to be known as hyperparameters, are the signal variance σ_f^2 and lengthscale l. The signal variance controls the typical amplitude of sample functions and the lengthscale controls the typical lengthscale of variation. We call them hyperparameters to emphasise that they are parameters of a non-parametric model.

In this instance, function variables close in the input space are highly correlated, whereas function variables relatively far apart when compared with l are uncorrolated.

1.3 Other examples

The Matérn covariance function is given by

$$k(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu r}}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu r}}{l} \right) \tag{6}$$

where $r = \|\boldsymbol{x} - \boldsymbol{x}^{'}\|$, K_ν is the modified Bessel function, $\Gamma(\cdot)$ is the gamma function and σ, l and ν are the hyperparameters.

Again this covariance function is isotropic and if we let $\nu \to \infty$ then we regain the squared exponential covariance. For finite ν the sampled functions are signficantly rougher than those from the squared exponential.

Another interesting covariance function, especially for use in finance, would be the Brownian motion kernel.

$$k(\boldsymbol{x}, \boldsymbol{x}^{'}) = \sigma^2 \min(\boldsymbol{x}, \boldsymbol{x}^{'}) \tag{7}$$

which is non-stationary.

The final covariance function to be introduced is the anisotropic, stationary version of the squared exponential covariance function. The change made is to allow an independent lengthscale hyperparameter l_d for each input dimension:

$$k(\boldsymbol{x}, \boldsymbol{x}^{'}) = \sigma^2 \exp\left[-\frac{1}{2} \sum_{d=1}^{D} \left(\frac{x_d - x_d^{'}}{l_d} \right)^2 \right]. \tag{8}$$

This can be said to implement automatic relevance determination. If a particular input dimension d has no relevance to the regression, then the associated lengthscale will increase to essentially filter out that feature.

It is possible to combine covariance functions by summing, taking the product or convolutions to obtain more flexible and complex processes. This has been done using the problem of CO_2 emissions from Moa Launa.

1.4 Gaussian Process Regression

Consider the problem of regression (in one dimension for simplicity) in which we wish to make predictions with noisy observations. To do this we use the covariance function to express our prior belief about the underlying function we are modelling.

We start with a zero mean Gaussian process prior with covariance matrix K on the function variables,

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, K). \tag{9}$$

We assume that observed data y is an observation of the function corrupted by Gaussian white noise

$$y = f + \varepsilon, \; \mathbb{E}[\varepsilon(\boldsymbol{x}), \varepsilon(\boldsymbol{x}')] = \sigma_n^2 \delta_{\boldsymbol{x}\boldsymbol{x}'} \tag{10}$$

where δ is a Kronecker delta and σ_n^2 is the variance of the noise. So the noise model, or likelihood can be expressed as

$$\boldsymbol{y}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma_n^2 \boldsymbol{I})$$

where $\boldsymbol{I}$ is the identity matrix. By the marginalisation property, its possible to integrate over the function variables $\boldsymbol{f}$ to give the marginal likelihood,

$$\mathbb{P}(\boldsymbol{y}) = \int d\boldsymbol{f}\mathbb{P}(\boldsymbol{y}|\boldsymbol{f})\mathbb{P}(\boldsymbol{f}),$$

APPENDIX?

$$\implies \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, K + \sigma_n^2 \boldsymbol{I})$$

We have training set comprised of input, output pairs $X = \{(x_i, y_i)|i = 1, \dots, n\}$ and test data (i.e. the predictions to be made) $X_* = \{x_i|i = 1, \dots, n_*\}$. We introduce the shorthand notation $K = K(X, X)$, $K_* = K(X, X_*)$ and $K_{**} = K(X_*, X*)$. K_* denotes the $n \times n_*$ matrix of the covariances, generated using the selected covariance function, evaluated at all pairs of the training inputs in X and the test inputs in $X*$. We can now write the the joint distribution of the observed target values $\boldsymbol{y}$ and the function values at the test location $\boldsymbol{f}_*$ under the prior as

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} K + \sigma_n^2 I & K_* \\ K_*^T & K_{**} \end{bmatrix}\right) \tag{11}$$

Conditioning this joint Gaussian prior distribution on the observations gives the predictive distribution

$$\begin{aligned} \boldsymbol{f}_*|X_*, X, \boldsymbol{f} &\sim \mathcal{N}(\bar{\boldsymbol{f}}_*, cov(\boldsymbol{f}_*)), \text{ where} && (12)\\ \bar{\boldsymbol{f}}_* &= \mathbb{E}[\boldsymbol{f}_*|X_*, X, \boldsymbol{f}] && (13)\\ &= K_*^T[K + \sigma_n^2 I]^{-1}\boldsymbol{y}, && (14)\\ cov(\boldsymbol{f}_*) &= K_{**} - K_*^T[K + \sigma_n^2 I]^{-1}K_*. && (15) \end{aligned}$$

The variance is expressed as the difference between two terms; the first representing the prior covariance and the second representing the information the observations give about the function. It is, also, important to note that these equations give predictive crrelations between any pair of test inputs. It can be thought of as a Gaussian Process in its own right with non-zero mean function and covariance function in form above.

Often we use only the marginal variances as a measure of predictive uncertainty. In this case we can consider the single general test input $\boldsymbol{x}_*$, at which the predictive mean and variance are given by:

$$\begin{aligned} \mu_* &= K_*^T[K + \sigma_n^2 I]^{-1}\boldsymbol{y}, \\ \sigma_*^2 &= K_{**} - K_*^T[K + \sigma_n^2 I]^{-1}K_* + \sigma_n^2. \end{aligned}$$

In terms of computational cost, computing the inverse $K + \sigma_n^2 \boldsymbol{I}$ costs $\mathcal{O}(N^3)$. The mean prediction is a weighted sum of N basis functions, $\mu_* = K_*^T \boldsymbol{\alpha}$, where $\alpha = [K + \sigma_n^2 \boldsymbol{I}]^{-1}$. So by precomputing $\boldsymbol{\alpha}$, the mean prediction per test case costs $\mathcal{O}(N)$. In terms of the predictive variance, no such precalcualtion can be made and costs $\mathcal{O}(N^2)$ per test case. The most stable and efficient method of implementation of these equations is using Cholesky decomposition of the covariance matrix, which is shown in my "GPfromScratch" notebook.

2 Hyperparameter learning

Although cross-validation can be used, for Gaussian processes we can select the hyperparameters directly from the training data. This is as the Gaussian process defines a full probabilistic model. Ideally, we would place a prior and compute a Bayesian posterior $\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{y})$ on the hyperparameters. However, this is not analytically tractable in general. So we turn our attention to the marginal likelihood as an appropiate cost function. We try to minimise the negative log marginal likelihood $\mathcal{L}$ with respect to the hyperparameters of the covariance $\boldsymbol{\theta}$ as such

$$\begin{aligned} \mathcal{L} &= -\log \mathbb{P}(\boldsymbol{\theta}|\boldsymbol{y}), \\ &= \frac{1}{2}\log\det \boldsymbol{C}(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{y}^T \boldsymbol{C}^{-1}(\boldsymbol{\theta})\boldsymbol{y} + \frac{N}{2}\log(2\pi) \end{aligned}$$

where $\boldsymbol{C} = K +_n^2 \boldsymbol{I}$.

We must be careful not to overfit the model when using the training data to optimise the parameters. Firstly, note that the covariance functions shown so far only have a limited number of hyperparameters and so overfitting tends not to be a problem. Secondly, we are not optimising the function variables themselves, rather integrating over their uncertainty. The Gaussian process hyperparameter optimisation takes place at a higher heirarchical level and can be referred to as type II maximum likelihoood.

The minimisation of the negative log marginal likelihood is a non-convex optimisation task. In the GPfromScratch we use a Scipy Optimiser which uses a quasi-Newton method called BFGS to computer the optimal hyperparameters. The cost of computing the log marginal likelihood and its gradients is dominated by the inversion of the covariance matrix $\boldsymbol{C}$. Local minima can be also be a problem, particularly in cases of small amount of data. It is always good practice to make several optimisations from random starting points and investigating the different minima.

Need more time for a write up for MonteCarlo methods which would also be included here. A very basic coded example can be found at the end of "GPFlowTest" notebook.

3 The Future

Over christmas break, I hope to clarify the notation used so far; write a little about Monte Carlo methods; and try experimenting with Stan, although currently I am happy with using GPFlow and Python.

A financial application of the Gaussian process would be volatility forecasting. In order for this to be computationally viable some use of sparse methods would be necessary. An investigation into the State-Space Gaussian Process could also be of use in this financial setting.