Gaussian Processes

Lesson No. 09

Thiago Gomes Marçal Pereira *

1 Definition

When working with supervised learning a good approach is to infer different distributions over functions given a set of data, and then, make new predictions over given input data. Gaussian Processes (GP) are a way of doing that.
With GP, our intention is to define priors over functions, and having some data, find a way to convert that to posteriors.
To work with Gaussian Processes, we must be given a set of "pairs" (x_i, y_i), where:

$$y_i = f(x_i)$$

GP also assumes that the distribution of $f(x)$ is Gaussian with mean $\boldsymbol{\mu}(x)$ and covariance given by a kernel function $\kappa(x_i, x_j)$. This is because, we expect the output of a function to be similar, if the input parameters x_i and x_j are also similar.

2 Regression

The prior is define by

$$f(x) \sim GP(m(x), \kappa(x, x')) \tag{1}$$

where $m(x)$ is the mean and $\kappa(x, x')$ is the kernel.

And $\kappa()$ is defined as a Gaussian over a finite set of points.

$$p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{K}) \tag{2}$$

where $\boldsymbol{K}_{x,j}$ is the set of covariances calculated over x_i and x_j, and $\boldsymbol{\mu}$ is the set of means for each x.
For simplicity, the means can be defined as $m(x) = 0$, since GP are flexible to model means arbitrarily.

2.1 Noise-free Predictions

Considering a training set $\mathcal{D} = \{(x_i, f_i), i = 1 : N\}$ being a noise-free observation, we want to predict our outputs $\boldsymbol{f}_*$, over a test set $\boldsymbol{X}_*$.
As previously stated, we expect our outputs to be similar based on similar inputs. So we can define our GP to have the following distribution form

$$\begin{pmatrix} f \\ f_* \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} K & K_* \\ K_*^T & K_{**} \end{pmatrix}\right) \tag{3}$$

And so, our posterior can be calculated as

$$p(f_*|X_*, X, f) = \mathcal{N}(f_*|\mu_*, \sum{}_*) \tag{4}$$

with,

$$\boldsymbol{\mu}_* = \boldsymbol{\mu}(\boldsymbol{X}_*) + \boldsymbol{K}_*^{\boldsymbol{T}} \boldsymbol{K}^{-1}(\boldsymbol{f} - \boldsymbol{\mu}(\boldsymbol{K})) \tag{5}$$

*RA: 189691 - t189691@g.unicamp.br

and

$$\sum\nolimits_* = \boldsymbol{K}_{**} - \boldsymbol{K}_*^{\boldsymbol{T}} \boldsymbol{K}^{-\mathbf{1}} \boldsymbol{K}_* \tag{6}$$

We can also parametrize our kernel function by doing

$$K(x, x') = \sigma_f^2 exp(-\frac{1}{2\ell^2}(x - x')^2) \tag{7}$$

2.2 Predictions over Noisy Observations

When we have noisy observations, we can consider $y = f(x) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$.
In this case, the model won't interpolate the training data, but should come close to the observed data during training, and so, our covariance will also consider the noise.

$$cov[y_p, y_q] = \kappa(x_p, x_q) + \sigma_y^2 \delta_{pq} \tag{8}$$

and over our data,

$$cov[y|X] = K + \delta_y^2 I_N \stackrel{\Delta}{=} K_y \tag{9}$$

Considering our test inputs to be noise-free and assuming mean is zero, we have out distribution as

$$\begin{pmatrix} y \\ f_X \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right) \tag{10}$$

And for a single test input, we can simplify our posterior as

$$p(f_*|x_*, X, y) = \mathcal{N}(f_*|k_*^T K_y^{-1} y, k_{**} - k_*^T K_y^{-1} k_*) \tag{11}$$

with k_* being the set of covariances between each training input and our test input data, and k_{**} being the covariance between our test data and itself $\kappa(x_*, x_*)$

2.3 Kernel Parameters Estimation

It would be possible to make a search of values, but this would be very time expensive. So, a good approach, is to maximize the marginal likelihood, using Bayesian approach.

$$p(y|X) = \int p(y|f, X) p(f|X) df \tag{12}$$

To calculate the marginal likelihood, we take the logarithm

$$log p(y|X) = log \mathcal{N}(y|0, K_y) = -\frac{1}{2} y K_y^{-1} y - \frac{1}{2} log|K_y| - \frac{N}{2} log(2\pi) \tag{13}$$

where the first term is our data fit, the second term is the model complexity and the third term is a constant and can be discarded.

To maximize the marginal likelihood, we then denote the kernel parameters by θ, so

$$\frac{\partial}{\partial \theta_j} log p(y|X) = \frac{1}{2} y^T K_y^{-1} \frac{\partial K_y}{\partial \theta_j} K_y^{-1} y - \frac{1}{2} tr\left(K_y^{-1} \frac{\partial K_y}{\partial \theta_j}\right) \tag{14}$$

so

$$= \frac{1}{2} tr\left(\left(\alpha \alpha^T - K_y^{-1}\right) \frac{\partial K_y}{\partial \theta_j}\right) \tag{15}$$

where $\alpha = K_y^{-1} y$
Given this, we might estimate the kernel parameters using gradient-based optimizer.

3 Gaussian Processes and Gaussian Linear Models

Extending GPs to GLM classification cases, could lead to different approaches. The simplest and fastest would be Gaussian approximation.

3.1 Binary Classification

We define the model as $p(y|x_i) = \sigma(y_i f(x_i))$ and assume $y_i \in -1, +1$ and $\sigma(z)$ can assume different values depending on the approach:
$\sigma(z) = sigm(z)$ for logistic regression
$\sigma(z) = \Phi(z)$ for probit regression
for GP regression we assume $f \sim GP(0, \kappa)$

To compute the posterior, we define the log of the unnormalized model and at convergence our posterior is approximate as

$$p(f|X, y) \approx \mathcal{N}(f, (K^{-1} + W)^{-1}) \tag{16}$$

and to optimize the kernel parameters, we need the marginal likelihood as

$$log p(y|X) \approx log p(y|\hat{f}) - \frac{1}{2}\hat{f}^T K^{-1}\hat{f} - \frac{1}{2} log|K| - \frac{1}{2} log|K^{-1} + W| \tag{17}$$

where $W \triangleq -\nabla\nabla log p(y|f)$ is a diagonal matrix because the data is conditional on f.

3.2 Multi-class Classification

Each class has one latent function and is priori-independent and so, can use different kernels. As for the binary classification, a Gaussian approach for the posterior is used, but using the multinomial probit.

4 Conclusion

Using Gaussian Processes provide a good way of estimating outputs based on different possible functions. Its capability of using different kernels also provide a gain in the results, because it also has uncertainty information. It also benefits from properties of normal distributions, and make estimations fast, and with errors easily measured.