Emulators: How they work

Tom McClintock

June 26, 2017

Note: all of this comes from David Barber's book **Bayesian Reasoning and Machine Learning**.

1 Multivariate Gaussian

A multivariate Gaussian distribution is given by

$$p(\mathbf{x}|\mu,\Sigma) = \mathcal{N}(\mathbf{x}|\mu,\Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{1}$$

where μ is the mean vector of the distribution, Σ is the covariance matrix. Note that the inverse covariance Σ^{-1} is called the precision matrix.

2 Partitioned Gaussian

A feature of multivariate Gaussians for our purposes is the idea of a *partitioned* multivariate Gaussian. Consider $\mathcal{N}(\mathbf{z}|\mu,\Sigma)$ defined jointly over two vectors $\mathbf{x}$ and $\mathbf{y}$ of potentially different dimensions,

$$\mathbf{z} = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \tag{2}$$

with corresponding mean and partitioned covariance

$$\mu = \begin{pmatrix}\mu_x\\ \mu_y\end{pmatrix} \tag{3}$$

$$\Sigma = \begin{pmatrix}\Sigma_{xx} & \Sigma_{xy}\\ \Sigma_{yx} & \Sigma_{yy}\end{pmatrix} \tag{4}$$

where $\Sigma_{xy} = \sigma_{yx}$. The marginal distribution is given by

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu_x,\ \Sigma_{xx}) \tag{5}$$

but the *conditional* distribution is given by

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(\mathbf{y}-\mu_y),\ \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}). \tag{6}$$

In other words, we have separated two parts of a multivarate Gaussian, and written the conditional probability of one vector given the other. Note that if there is no correlation $\Sigma_{xy} = 0$ then 6 reduces to 5.

3 Gaussian Process

A Gaussian process is a process by which we assume that given some set of observations $\mathbf{y}(\mathbf{x})$, a future observation $y^*(x^*)$ is drawn from a multivariate Gaussian, given by 6. (**Note**: Gaussian processes always assume that $\bar{\mathbf{y}} = 0$, or that μ_y has already been subtracted off of $\mathbf{y}$ and can be added on later.)

Further, since every y is a function of some location in the domain x, we assume that we have some function (the *kernel* function) that describes the covariance between observations made at different locations in the domain $k(x_1, x_2)$. Therefore, any element of a covariance matrix constructed between observations is given by

$$[K]_{1,2} = k(x_1, x_2). \tag{7}$$

This means that the covariance matrix of the observations $\mathbf{y}$ is

$$[K_{x,x}]_{i,j} = k(x_i, x_j), \quad i, j = 1, ..., N \tag{8}$$

where N are the number of observations. Additionally, the covariance between the prediction y^* and the observations $\mathbf{y}$ is a vector

$$[K_{x,x^*}]_i = k(x_i, x^*), \quad i = 1, ..., N. \tag{9}$$

Since $\mathbf{y}$ and y^* form a multivariate Gaussian, we can immediately write down a prediction and uncertainty on that prediction from 6

$$p(y^*|\mathbf{y}) = \mathcal{N}(y^*|K_{x^*x}K_{xx}^{-1}\mathbf{y},\ K_{x^*x^*} - K_{x^*x}K_{xx}^{-1}K_{xx^*}) \tag{10}$$

where $K_{xx^*} = K_{x^*x}^T$. If the observations $\mathbf{y}$ have variances assiciated with them, σ^2 then we make the transformation $K_{xx} \to K_{xx} + \mathbf{I}\sigma^2$.

4 Kernel Functions

The kernel function (aka covariance function) isn't specified apriori, and in general could take almost any form. In Barber's book he explains in detail how different functions are appropriate for different purposes. In general, we will use the squared exponential kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = k(|\mathbf{x}_1 - \mathbf{x}_2|) = k_0 \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_1 - \mathbf{x}_2)^2}{L}\right). \tag{11}$$

where k_0 is the covariance amplitude, and L is known as the "kernel length". The domain $\mathbf{x}$ can be multi-dimensional, with each dimension having slightly different meanings (e.g. Ω_m, w, σ_8 in cosmology), so L is actually an array containing a kernel length for each dimension. These parameters k_0 and L are known as *hyperparameters* in the literature.

5 Hyperparameters

Values for the hyperparameters aren't random, but are informed by the observations. In general, L corresponds to the "feature size", or the approximate size in the domain of features in your observations, while k_0 is approximately the size of the error bars of your data.

The best way to find an optimal choice of hyperparameters is to write a likelihood of your observations which you can then maximize. This likelihood is

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2}\mathbf{y}^T K_{xx}^{-1}\mathbf{y} - \frac{1}{2}\log\det(2\pi K_{xx}). \tag{12}$$

This also lets you obtain full posteriors for your hyperparameters, which can be propogated forward into uncertainties into y^* if you want to be thorough.

6 Complexity and Limitations

Training a Gaussian process is $O(N^3)$ in time because it requires matrix inversion (faster if you have smart inverters), and $O(N^2)$ in space because it requires the matrix to be stored completely. Therefore, as your training data becomes large the algorithm becomes unweildy. Note: in very specific instances this can be mitigated, see Foreman-Mackay et al. 2017 (https://arxiv.org/abs/1703.09710).

If an optimizer is used to find the hyperparameters, then the resulting Gaussian process will be sensitive to the optimizer used. The best advice I can give is to try a few methods and see what works best.