$\mathcal{GP}$ Write Up 3

August 24, 2018

Covers inference with imaginary points on the grid

1. Inference with imaginary points on the grid

M number of observations on the grid

W number of imaginary observations that we introduce to complete the grid

L number of test points to extrapolate

$\mathbf{y}_w = \mathcal{N}(0, \sigma_w^2 I_w)$ the distribution over the imaginary grid observations. We let the variance be very high.

$N = M + W$ the total number of observations in the training set

$\mathbf{y}_N = [\mathbf{y}_M; \mathbf{y}_W]^T$ all observations in the training set

$k(\mathbf{x}_i, \mathbf{x}_j|\theta)$ covariance function with θ as hyper parameters

$K(X, X|\theta) = K_N$ the covariance matrix for $k(x_i, x_j|\theta)$ covariance function for N examples with θ as hyper-parameters

The covariance matrix in terms of the inputs

$$K_N + D_N = \begin{bmatrix} K_M + \sigma_M^2 I_M & K_{MW} \\ K_{MW}^T & K_{WW} + \sigma_W^2 I_W \end{bmatrix} \tag{1}$$

The predictive mean(posterior) for the L test points with N training points is given by:

$$\mu_L = K_{LN}(K_N + D_N)^{-1}\mathbf{y}_N = K_{LN}\begin{bmatrix} K_M + \sigma_M^2 I_M & K_{MW} \\ K_{MW}^T & K_{WW} + \sigma_W^2 I_W \end{bmatrix}^{-1}\mathbf{y}_N \tag{1}$$

Let

$$(K_N + D_N)^{-1}\mathbf{y}_N = \mathbf{v}_N \tag{2}$$

The algorithm for training/fit consists of two steps:

1. **Conjugate gradient step**: Solve equation (2) using conjugate gradient to obtain $\mathbf{v}_N$ with θ hyper-parameters fixed.
2. **Optimization step**: Update the hyper-parameters θ with $\mathbf{v}_N$ obtained above.

$$\theta \leftarrow argmin_\theta ||\mathbf{y}_N - (K_N + D_N)\mathbf{v}_N||_2^2 \tag{2}$$

With the updated θ solve for one last time with the new θ:

$$\mathbf{v}_N = (K_N + D_N)\mathbf{y}_N \tag{3}$$

for $\mathbf{v}_N$ and obtain:

$$\mu_L = K_{LN}\mathbf{v}_N \tag{3}$$

To establish the correctness of the above approach we have to compare (1) with (3). Equation (1) can be solved using the traditional Cholesky decomposition and we can quantify the error as, $||\mu_L^{(1)} - \mu_L^{(3)}||_2^2$.

2. Tensor Product Kernels

The conjugate gradient method requires a matrix multiplication Kv on every iteration. This matrix multiplication has $O(N^2)$ time complexity. We can reduce this to $O(N)$ time complexity with $O(N)$ storage by considering only tensor product kernels.