$\mathcal{GP}$ Write Up 7

October 15, 2018

Github: https://github.com/AnuroopKuppam/conv-gps

Conv-gp implementation: https://colab.research.google.com/drive/1P7IBJI01EuehxWorcvcmDtqbn3RI87D_

Experimental results

1. Results comparison Naive vs Convolutions with Conjugate gradient loss

All the experiments are performed using the squared exponential kernel:

$$k(x,y) = \sigma_f^2 exp(\frac{(x-y)^T(x-y)}{2l^2}) \tag{1}$$

In the naive way we maximize the marginal log likelihood wrt to σ_f and l:

$$logp(y|X) = -0.5y^T(K+\sigma_n^2 I)y - 0.5log|K+\sigma_n^2 I| - 0.5nlog(2\pi) \tag{2}$$

For the training data we used three different smooth functions on a grid of size 50×50, with 5% of the observations missing and are filled with $\mathcal{N}(0, 10^{100})$.

The convolution implementation uses the conjugate gradient loss and the network is run for 10k iterations, with σ_f, l set to 0.1 and 10 respectively.

We compare the performance between the methods by taking the predicted mean for the naive and convolution methods and obtaining the squared l2 norm between the predicted mean and the training data without the imaginary observations:

$$L(y\_pred, y\_train) = ((y\_pred - y\_train) * mask)^2 \tag{3}$$

For the functions described below:

$$X \in [-25, 25], Y \in [-25, 25] \tag{4}$$

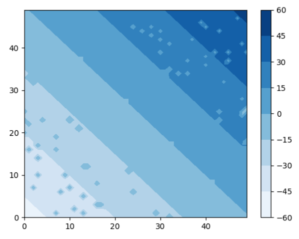
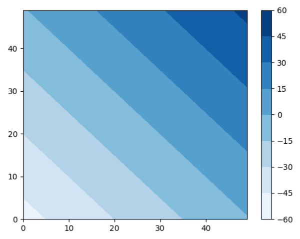
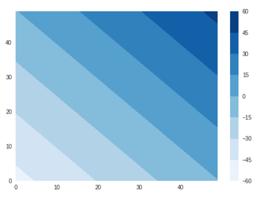
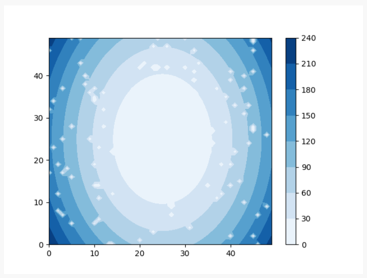
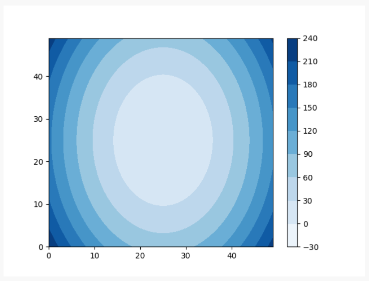
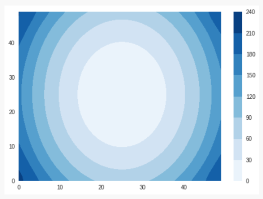
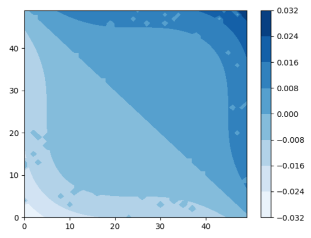
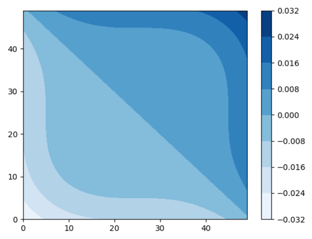
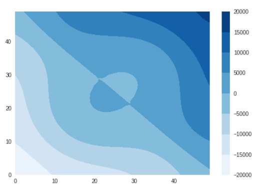
Function	**Train**	**Naive Predict**	**Conv Predict**
$X+Y$			
X^2+Y^2			
X^3+Y^3			

Table 1: Predictions for naive gp and conjugate gradient methods against the training data with missing observations. Each of the above images are contour plots.

Function	**Naive , σ_f, l**	**Naive loss**	**Conv Loss**
$X+Y$	5.6,10.7	1.8×10^{-5}	464.6
X^2+Y^2	5.9,9.8	0.0014	4k
X^3+Y^3	4.7,10.9	9.6×10^{-11}	10^{10}

Table 2: Hyper parameters learned in naive gp, loss incurred according to equation (3) for naive and conjugate gradient implementation.