GP - Rasmussen & Williams - Ch. 2: Regression

Outline

GP prior

$$k(x,y) = \exp(-\tfrac{1}{2}|x-y|^2) \quad (1)$$

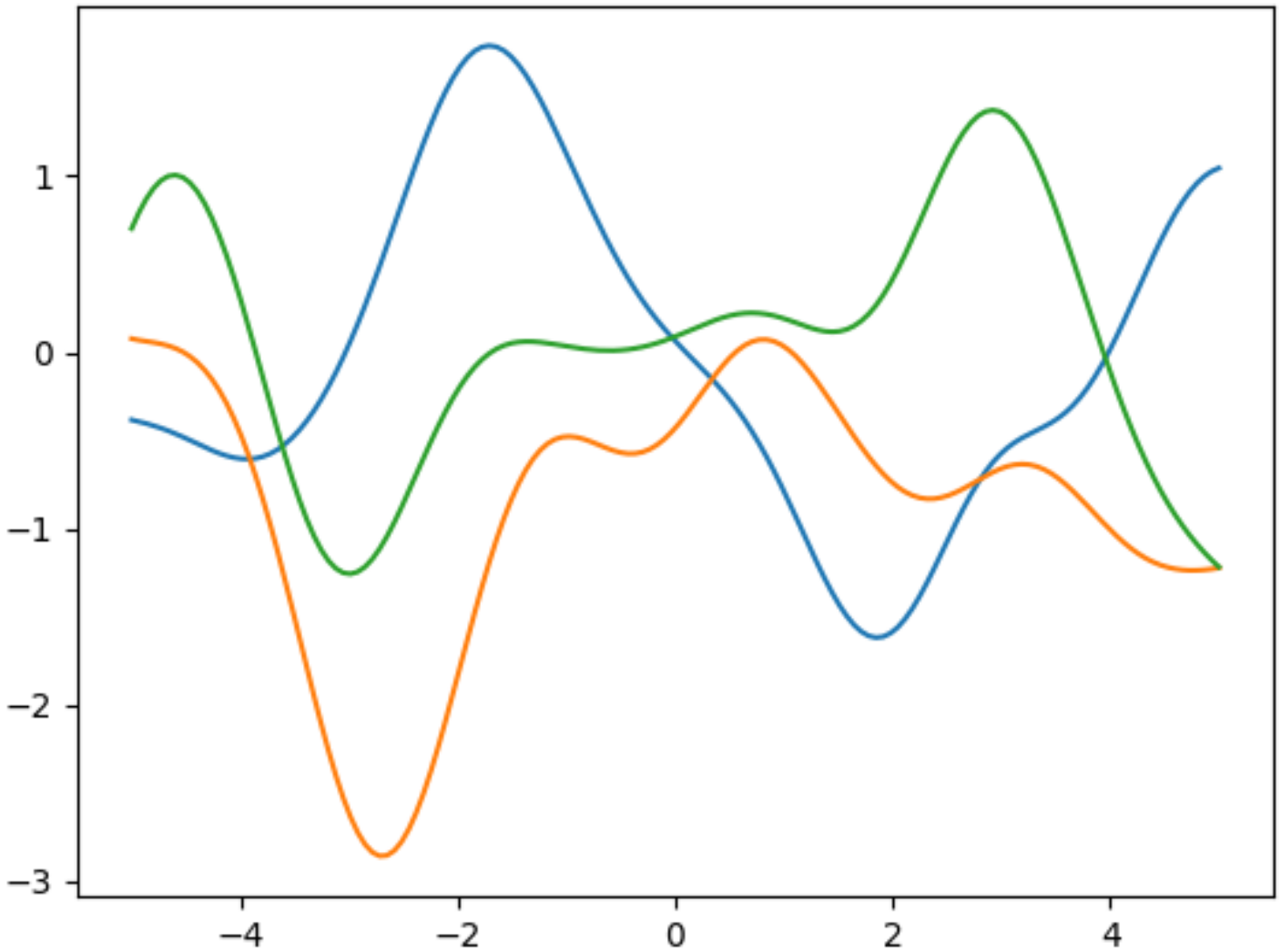
$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, K(\mathbf{x},\mathbf{x})) \quad (2)$$

Python code

```
import numpy as np

def rbf(length_scale):
    def k(x,y):
        if len(x.shape)==1:
            d = 1
        else:
            d  = x.shape[1]
        lx = x.shape[0]
        ly = y.shape[0]
        dists =  np.sum(((x.T.reshape([d,lx,1]) -  y.T.reshape([d,1,ly]))/length_scale)**2,0)
        return np.exp(-.5 * dists)
    return k

def genSamplesSimple(x, k):
    n = x.shape[0]
    return np.random.multivariate_normal(np.zeros(n), k(x,x) + np.eye(n)*1e-8)

# def gesnSamplePostSimple(y,x,k,X):

# def gesnSamplePostSimple(y,x,k,X):
```

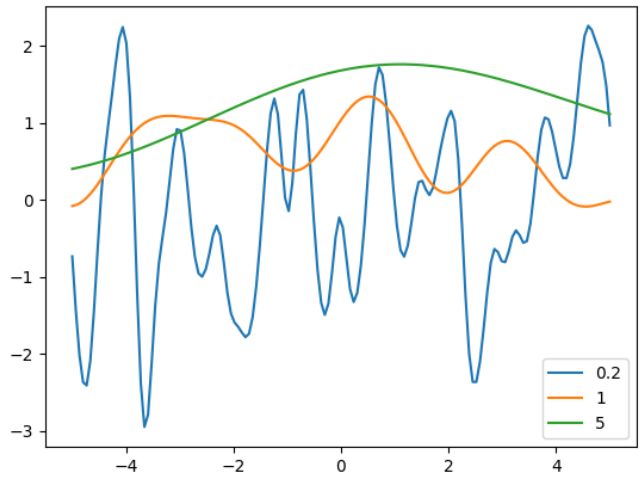
```
import matplotlib.pyplot as plt
plt.figure()
x = np.linspace(-5,5,150)
k = rbf(1)
for i in range(3): plt.plot(x, genSamplesSimple(x,k));
```

Random functions in 1D



Different length scales

```
scales = [0.2, 1, 5]
for i in scales:
  plt.plot(x, genSamplesSimple(x,rbf(i)))
plt.legend(scales);
```



Two dimensions

```
x = np.linspace(5,-5,50)
xx, yy = np.meshgrid(x, x)
xy = np.concatenate(
       [xx.reshape([1,-1]),yy.reshape([1,-1])]).T
z = genSamplesSimple(xy, rbf(2)).reshape([50,50])
```

```
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot_surface(xx, yy, z, rstride=8,
                cstride=8, alpha=0.3)
cset = ax.contour(xx, yy, z, zdir='z',
                  offset=-2.5, cmap=cm.coolwarm)
cset = ax.contour(xx, yy, z, zdir='x',
                  offset=-5, cmap=cm.coolwarm)
cset = ax.contour(xx, yy, z, zdir='y',
                  offset=5, cmap=cm.coolwarm)

ax.set_xlabel('X')
ax.set_xlim(-5, 5)
ax.set_ylabel('Y')
ax.set_ylim(-5, 5)
ax.set_zlabel('Z')
ax.set_zlim(-2.5, 2.5)
```