TU München
Fakultät für Informatik
PD Dr. Rudolph Triebel
John Chiotellis, Max Denninger

Machine Learning for Computer Vision
Winter term 2017

December 12, 2017
Topic: Kernels and Gaussian Processes

Exercise 1: Constructing kernels

Let k_1 and k_2 be kernels, $f : \mathbb{R}^n \to \mathbb{R}$ an arbitrary function . Show that we can construct new kernels via

a) $k(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2)$

b) $k(x_1, x_2) = k_1(x_1, x_2)k_2(x_1, x_2)$

c) $k(x_1, x_2) = f(x_1)k_1(x_1, x_2)f(x_2)$

d) $k(x_1, x_2) = \exp(k_1(x_1, x_2))$

e) $k(x_1, x_2) = x_1^T A x_2$, where A symmetric, positive semi-definite $n \times n$ matrix

Exercise 2: Gaussian Process Regression

a) Implement a simple Gaussian Process Regressor. As training data, you can use the provided code snippet to generate ten points along a sinus curve. Use a fixed *length scale* parameter l of 3.0, with a σ_f of 1.0 and σ_n of 0.5.

```
import numpy as np
sigma_noise = 0.5
x_min, x_max = -5, 5
X_train = np.linspace(x_min, x_max, num=10)
# Simulate sinusoid with some gaussian noise
Y_train = [10*np.sin(x) + (np.random.rand() - 0.5) * sigma_noise for x in
    X_train]
```

We suggestion you use a kernel function like this:

```
# Kernel function
def rbf_kernel(x, y, l=1.0, sigma_f=1.0, sigma_n=0.5):
    return sigma_f**2 * np.exp(-(x - y)**2 / (2*l**2)) + sigma_n**2*(x==y
    )
```

b) Now test different *length scale* parameters and plot the results and compare them to each other, what do you observe.

c) Do the same for the σ_f parameter, use a *length scale* of 0.5. How does it influence the result?

Exercise 3: Laplace Approximation

In Gaussian Process classification, we cannot integrate exactly over the parameters $\mathbf{w}$.

a) Why is this the case? Why is this a problem?
Name 3 approaches one can use to tackle this problem?