Principles of Machine Learning: Exercise 3

Alina Pollehn (3197257), Julian Litz (3362592), Manuel Hinz (3334548)
Felix Göhde (3336445), Felix Lehmann (3177181), Caspar Wiswesser (3221493)
Adrian Köring (3347785), Greta Günther (3326765), Linus Mallwitz (3327653)
Niklas Mueller-Goldingen (3363219), Jennifer Kroppen (2783393)

04.12.2023

Implementation of exercise 3.1

outer difference:

```
def diffMatrix(u,v):
    return np.subtract.outer(u,v)
```

outer product:

```
def prodMatrix(u,v):
    return np.multiply.outer(u,v)
```

for general operators:

```
def outer_operator(f,u,v):
    return f(np.expand_dims(u,axis=1), np.expand_dims(v,axis=0))
```

Implementation of exercise 3.2.1

Linear kernel matrix $K(u, v|\alpha) \in \mathbb{R}^{n_u \times n_v}$

$$[K]_{ij} = \alpha u_i v_j$$

```
def linearKernelMatrix(u,v,alpha):
    return alpha*prodMatrix(u,v)
```

Implementation of exercise 3.2.2

gaussian kernel matrix $K(u, v|\alpha, \sigma) \in \mathbb{R}^{n_u \times n_v}$

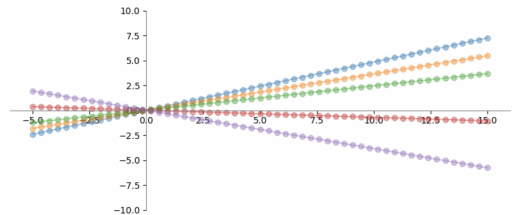
$$[K]_{ij} = \alpha \exp\left(-\frac{(u_i - v_j)^2}{2\sigma^2}\right)$$

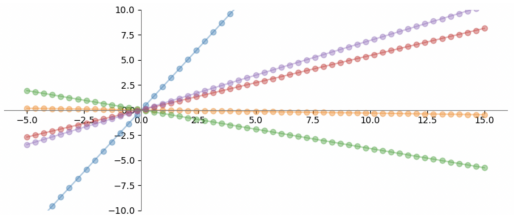
```
def gaussKernelMatrix(u,v,alpha,sigma):
    return alpha*(np.exp(-diffMatrix(u,v)**2/(2*sigma**2)))
```

Sampling from a linear kernel matrix

Sampling 5 vectors twice yields

```
y=multivariate_normal(vec0 , linearKernelMatrix(vecX,vecX, 1))
```



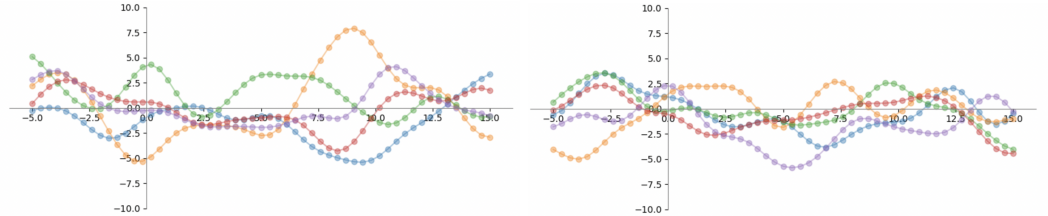


Keep in mind, that these results are random and will look different each time. Outliers, as seen in the second image, are quite common.

Sampling from a gaussian kernel matrix

Sampling 5 vectors twice yields

```
y=multivariate_normal(vec0 , gaussKernelMatrix(vecX,vecX, alpha=6, sigma=1.5))
```



Keep in mind, that these results are random and will look different each time