TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Datenverarbeitung
PD Dr. Martin Kleinsteuber

Information Retrieval in High Dimensional Data

Assignment #3, 05.07.2017

Due date: 17.07.2017, 10:30 A.M.

Please hand in your solutions via Moodle. You can add your conclusions for the PYTHON Task as comments in the PYTHON files. For the other exercises deliver a PDF file either created using Latex or as a scan of your handwritten solution.

Solutions can be handed in by groups of up to **four** people. Please state the names of your group members at a prominent place in your submission. (For example, at the beginning of your provided PYTHON code or in a separate text file.)

Kernel PCA (kPCA)

Task 1.1: [5 points] Download the file `task3_1_kpca_demo.py` from the web page. Implement KPCA using a Gaussian kernel function (1) by filling in the missing lines.
Vary `alpha` and `sigma` and observe the generated plots. In each of the two plots all generated data points are plotted. The color in the first plot indicates the value of the respective point when projected onto the first PC. The color in the second plot indicates the value of the respective point when projected onto the second PC.

For appropriate choices of `alpha` and `sigma` you see a horizontal separation in the first component and a vertical separation in the second component. However, if `alpha` becomes too large, the second component is no longer vertical. Test this behavior for `alpha` between 1 and 12 and determine the `sigma` that provides vertical separation in the second component for each `alpha`. Provide a 2-dimensional plot with the axis `alpha` and `sigma` to illustrate this behavior.

Task 1.2: [5 points] Download the Python script `task3_1_toy_data.py` for generating a toy example. The produced data set contains two groups. Implement kPCA and determine the kernel function that allows you to linearly separate the data using the first principal component. Illustrate this by plotting the reduced data.

Kernel SVM

Task 2: [10 points] The file `task3_2.py` that is included in the zip file on Moodle contains the

frame for a kernel SVM method. Your task is to fill in the gaps in the provided file such that a kernel SVM using the Gaussian kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\right), \tag{1}$$

is trained.

The last part of the script shows you the classification regions. Provide the plots for $\sigma = \{1, 0.5, 0.1, 0.01\}$.

Note that you need to install the `cvxopt` package. If you use Anaconda, I recommend installing it via the command
`conda install -c omnia cvxopt=1.1.8`