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

Information Retrieval in High Dimensional Data

Assignment #3, 09.01.2018

Due date: 23.01.2018, 6 P.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. Alternatively, you can hand in an IPython/Jupyter notebook. Solutions can be handed in by groups of **four or five** people. Please state the group number and 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

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

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. (Note that horizontal separation means, that you can separate the different colors in the plot with vertical lines and vice versa for vertical separation.) 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.

Fisher LDA

Task 2: [10 points] Refer to Lab Course 7. In 1.e, we implemented LDA by normalizing the between-scatter matrix $\mathbf{S}_b$ with the within-scatter matrix $\mathbf{S}_w$. Another way to approach LDA is by finding the projection space spanned by the columns of

$$\hat{\mathbf{U}} = \underset{\mathbf{U} \text{ s.t } \mathbf{U}^\top \mathbf{U} = \mathbf{I}_k}{\arg\max} \operatorname{tr}(\mathbf{U}^\top \mathbf{S}_b \mathbf{U}) - \beta \operatorname{tr}(\mathbf{U}^\top \mathbf{S}_w \mathbf{U}),$$

and projecting the data onto that space. Here, β is a real positive tuning parameter to be chosen by hand.

- Explain, how you can solve this problem by EVD.
- Provide PYTHON code that creates the plots in accordance with the lab course by performing the dimensionality reduction with the described approach. In particular, choose $k = 2$ for the subspace dimension.
- Choose 1,2 and 3 as the digit classes and 1 to 1000 as the training samples from each class.
- Create four plots for $\beta = 2$, $\beta = 4$, $\beta = 6$ and $\beta = 8$, respectively. Give an interpretation of the results.