Exercise Session 3: Unsupervised Learning and Large Scale Problems

Dries Hendrikx* Konstantinos Kontras† Francesco Tonin‡

Yingyi Chen§ Johan Suykens

Academic Year 2020-2021

This document describes a step-by-step guide for the application of support vector machine methods. The goal of the exercise sessions is to provide you with experience and teach you how to tackle future application problems. The MATLAB scripts and toolboxes, the course documents, the referred papers and all the datasets used in the exercise sessions are available for academic purposes on `https://toledo.kuleuven.be`.

- The exam consists of an oral discussion on the basis of the reports written for the 3 exercise sessions. It is important to show that you have understanding about the problem and that you can work in a constructive manner towards getting good solutions to given problems.
- The reports have to be written *individually*.
- The reports contain the solutions to the exercises *and* the homework problems. All the questions that need to be answered in the report are indicated by the grey box, which reads 'Questions'. We do not expect minimal answers to the questions but rather 'comprehensive' answers.
- A good report finds a good balance between *visuals* on the one hand and to-the-point *explanations* on the other hand. Use figures and tables to explain things, rather than only long and elaborate sentences. Make sure to include a few key equations in the textual part.
- Both the text and figures should not be too small. They have to be readable.
- Write a *separate* report for every exercise session. At the end of the semester, you have to submit a pdf document of **30 pages maximum** consisting of 3 separate reports in single column (max 10 pages for each session).

*dries.hendrikx@esat.kuleuven.be
†konstantinos.kontras@esat.kuleuven.be
‡francesco.tonin@esat.kuleuven.be
§yingyi.chen@esat.kuleuven.be

1 Exercises

1.1 Kernel principal component analysis

Kernel principal component analysis (KPCA) corresponds to linear PCA in a kernel-induced feature space, which is non-linearly related to the original input space. Thus, nonlinearities can be included via the kernel function and the corresponding problem keeps the form of an eigenvalue problem. Kernel PCA has numerous applications: it can be used for feature extraction, denoising, dimensionality reduction and density estimation, among others. In this section, we will explore the use of kernel PCA for denoising.

Download the files `kpca_script.m` and `pca.m` from Toledo. In order to illustrate the power of kernel PCA for denoising, this exercise focusses on an artificial example: a dataset which resembles the yin yang pattern. In order to get insight in the number of principal components, the choice of the kernel (and influence of kernel parameters) and the regularization hyperparameter, execute the script:

```
>> kpca_script
```

In order to provide uniformity across results, choose the number of data points equal to 400 and the data point dispersion equal to 0.3. Answer the following questions.

Questions

- Describe how you can do denoising using PCA. Describe what happens with the denoising if you increase the number of principal components.
- Compare linear PCA with kernel PCA. What are the main differences? How many principal components can you obtain?
- For the dataset at hand, propose a technique to tune the number of components, the hyperparameter and the kernel parameters.

1.2 Spectral clustering

Spectral clustering techniques make use of the eigenvectors of a Laplacian matrix derived from the data to create groups of data points that are similar. In this context, the kernel function acts as a similarity measure between two data points. The Laplacian matrix is consequently obtained by rescaling the kernel matrix. Note that these techniques can be interpreted as a form of kernel PCA.

Download the files `sclustering_script.m` and `two3drings.mat` from Toledo. Try the script:

```
>> sclustering_script
```

Questions

- Explain briefly how spectral clustering works.
- What are the differences between spectral clustering and classification?
- Edit the script and try different values of `sig2` (e.g., 0.001, 0.005, 0.01, 0.02). What is the influence of the `sig2` parameter on the clustering results?

1.3 Fixed-size LS-SVM

Based on the Nyström approximation, an approximation to the feature map can be obtained. This map can consequently be used to construct parametric models in the primal representation of the LS-SVM model. The approximation of the feature space is based on a fixed subset of data-points. One way to select this fixed-size set is to optimize the entropy criterion (`kentropy`) of the subset.

In some cases, we are interested in a sparser solution that we can attain using a predefined number of representative points. We can achieve this by applying ℓ_0-type of a penalty in an iterative fashion to an initial fixed-size LS-SVM solution.

For this exercise, we need the files `fixedSizeScipts.zip` (which contains two scripts) and `fixed-size` (which can be found in `SVM course scripts`). Download these files and add them to your directory.

Questions

- In which setting would one be interested in solving a model in the primal? In which cases is a solution in the dual more advantageous?
- What is the effect of the chosen kernel parameter `sig2` on the resulting fixed-size subset of data points (see `fixedsize_script1.m`)? Can you intuitively describe to what subset the algorithm converges?
- Run `fslssvm_script.m`. Compare the results of fixed-size LS-SVM to ℓ_0-approximation in terms of test errors, number of support vectors and computational time.

2 Homework problems

2.1 Kernel principal component analysis

Download `digitsdn.m`. In this homework problem, we do image denoising using kernel PCA. The dataset used in this exercise consists of images of handwritten numerals (0 to 9), extracted from a collection of Dutch utility maps. Approximately 20 patterns per class (digit 9 has 18 images, total of 198 patterns) have been digitized in binary images. Execute the script:

```
>> digitsdn
```

As a rule of thumb, `sig2` is calculated as the mean of the variances of each dimension times the dimension (number of features) of the training data.

Questions

- Illustrate the difference between linear and kernel PCA by giving an example of digit denoising for `noisefactor` = 1.0. Give your comments on the results (based on visual inspection).
- What happens when the `sig2` parameter is much bigger than the suggested estimate? What if the parameter value is much smaller? In order to investigate this, change the `sigmafactor` parameter for equispaced values in logarithmic scale.

- Investigate the reconstruction error on training (`Xtest`) and validation sets (`Xtest1` and `Xtest2`), as a function of the kernel PCA denoising parameters. Select parameter values such that the error on the validation sets is minimal. Can you observe any improvements in denoising using these optimized parameter settings?

2.2 Fixed-size LS-SVM

In the second homework problem, we investigate the use of fixed-size LS-SVM for two additional datasets: the Shuttle dataset (classification) and the California housing dataset (regression).

2.2.1 Shuttle (statlog)

Adjust `fslssvm_script.m` and proceed with classification on the Shuttle dataset. Additional information on the Shuttle dataset can be found at `https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)`.

Questions

- Explore and visualize (part of) the dataset. How many datapoints? How many and meaning of attributes? How many classes? What is to be expected about classification performance?
- Visualize and explain the obtained results.

2.2.2 California

Adjust `fslssvm_script.m` and proceed with regression on the California dataset. Additional information on the California dataset can be found at `http://www.dcc.fc.up.pt/%7Eltorgo/Regression/cal_housing.html`.

Questions

- Explore and visualize (part of) the dataset. How many datapoints? How many and meaning of attributes?
- Visualize and explain the obtained results.