Support Vector Machines: Methods and Applications
Exercise Session III

Carlos M. Alaíz   Emanuele Frandi   Johan A.K. Suykens
{cmalaiz,emanuele.frandi,johan.suykens}@esat.kuleuven.be

Introduction

The Matlab scripts and toolboxes, the related documents, referred papers and data-sets are available for academic purposes on Toledo: http://toledo.kuleuven.be/.

3 Exercise Session 3: Unsupervised Learning

For this session, the LS-SVMlab toolbox is used.

3.1 Kernel Principal Component Analysis

Kernel PCA 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 (like linear PCA)*. Kernel PCA can be used for feature extraction, denoising, dimensionality reduction and density estimation. In this Section we will use kernel PCA mainly for denoising.

Download the files digitsdn.m, kpca_script.m and pca.m available at: Toledo website → SVM Exercises Course → Assignments → Session 3. Download these files and put them inside the LS-SVMlab toolbox directory (LSSVMlab). Try the script:

```
>> kpca_script
```

We focus for a moment on this toy dataset in order to get insight in the number of components, the choice of the kernel and the kernel hyper-parameter. Can you describe what's happening with the denoising if you increase the number of principal components? What is the difference with linear PCA? How many principal components can you obtain with kernel PCA? and with linear PCA? Can you think of a technique to tune the number of components and the kernel hyper-parameter?

3.2 Handwritten Digit Denoising

This data-set consists of features of handwritten numerals ('0'-'9') extracted from a collection of Dutch utility maps. Approximately 20 patterns per class (for a total of 198 patterns) have been digitized in binary images.

Try the sample script on Toledo and explain what you observe:

```
>> digitsdn
```

More the number of principal components in kpca, more noise gets incorporated in the reconstructed signal. In other words we model the noise / overfit the signal.

By using a non-linear function k instead of the standard dot product, we implicitly perform PCA in a (possibly) high dimentional space which is non-linearly related to input space.

The number of components that can be obtained with kpca depends on the kernel we choose. It is possible to have infinite number of components as with RBF. With linear PCA the maximum number of components can be equal to dimension of input (feature) space.

3.3 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 then obtained by re-scaling the kernel matrix. These techniques can be interpreted as a form of kernel PCA.
Download the files `sclustering_script.m` and `two3drings.mat` available at: `Toledo website` → `SVM Exercises Course` → `Assignments` → `Session 3` → `Sample script and data for spectral clustering`.
Try the script:

```
>> sclustering_script
```

What is the difference with classification? Edit the script and try different values of `sig2` (*e.g.* 0.001, 0.005, 0.01, 0.2). What is the influence of the `sig2` hyperparameter on the clustering results?

Lower value of sigma produces better results. better classification.

3.4 Fixed-size LS-SVM

For this subsection we will need the scripts available from:

- `Toledo website` → `SVM Exercises Course` → `Assignments` → `Exercise 3` → `Fixed-size LS-SVM scripts`.
- `Toledo website` → `SVM Exercises Course` → `Course Documents` → `SVM course scripts`.

Based on the Nyström approximation, an approximation to the feature map is obtained. This mapping can be used to construct *parametric* models in the primal.

1. 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. A simple example illustrates the corresponding procedure (see sample script `fixedsize_script1.m` on the course website on Toledo). What is the influence of the chosen `sig2`?
2. Can you intuitively describe to what subset the algorithm converges? Given this optimized subset, the feature space mapping can be reconstructed:

   ```
   >> features = AFEm(subset,'RBF_kernel',sig2,X);
   ```

   (see sample script `fixedsize_script2.m` on the course website on Toledo).
3. In same cases we are interested in a sparser solution that we can attain using a predefined number of representative points[1]. We can achieve this by applying ℓ_0-type of a penalty in an iterative fashion to an initial Fixed-size LS-SVM solution. Run `fslssvm_script.m` and obtain the results. Compare the results of Fixed-size LS-SVM to ℓ_0-approximation in terms of the test errors, number of Support Vectors, etc.

3.5 Homework Problems

We would like you now to apply the procedures discussed above to the Handwritten Digits, Shuttle (statlog) and California Data-Sets. Answer the following set of questions. **Please, justify all of your answers as thoroughly as possible**. Keep in mind that one of the skills being evaluated during the final examination is also your ability to creatively and constructively address problems with which you may not be entirely familiar with the help of the tools learned during the course.

[1] Applying entropy criteria.

3.5.1 Handwritten Digit Denoising

Consider Subsection 3.2. Usually, a rule of thumb for `sig2` is calculated as the mean of the variances of each dimension times the dimension (number of features) of the training data.

- What happens when `sig2` hyperparameter is much bigger than the suggested estimate?
- Edit the `digitsdn` script and change the `sigmafactor` parameter for equispaced values in logarithmic scale and give your comments on the results.
- Illustrate the difference between linear and kernel PCA. Give two examples of digit denoising for a `noisefactor` of 1.0. Can you reconstruct using these methods the original digits of `Xtest2`? Check the reconstruction error on training and validation sets. Select `sig2` such that the error on the validation set is minimal. Can you observe any improvements?

3.5.2 Shuttle (statlog)

Please adjust `fslssvm_script.m` and proceed with classification on the Shuttle dataset. Explain and visualize the obtained results. Additional information can be found at `http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/shuttle/shuttle.doc`.

3.5.3 California

Please adjust `fslssvm_script.m` and proceed with regression on the California dataset. Explain and visualize the obtained results. Additional information can be found at `http://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html`.