Exercise Session 1: Classification

Alex Lambert* Sonny Achten† Konstantinos Kontras‡
Francesco Tonin§ Yingyi Chen¶ Johan Suykens

Academic Year 2022-2023

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).

*alex.lambert@esat.kuleuven.be
†sonny.achten@esat.kuleuven.be
‡konstantinos.kontras@esat.kuleuven.be
§francesco.tonin@esat.kuleuven.be
¶yingyi.chen@esat.kuleuven.be

1 Exercises

1.1 A simple example: two Gaussians

To get an intuitive idea what classification is about, the exercises start with a simple toy example. An artificial dataset is constructed from two classes of Gaussians with the same covariance matrices:

```
>> X1 = randn(50,2) + 1;
>> X2 = randn(51,2) - 1;
```

Note that ';' avoids the content of the variables to be written out on the screen. The corresponding class labels are defined:

```
>> Y1 = ones(50,1);
>> Y2 = -ones(51,1);
```

To see the content of the variables, type:

```
>> X1
```

Leaving out the ';' at the end of the statement allows for the variable to written out on the screen. Alternatively, you can type:

```
>> disp(X1);
```

The dataset can be visualized using the following series of commands:

```
>> figure;
>> hold on;
>> plot(X1(:,1), X1(:,2), 'ro');
>> plot(X2(:,1), X2(:,2), 'bo');
>> hold off;
```

where `'ro'` and `'bo'` define respectively the labels of the data of the positive class to be red circles and the labels of the data of the negative class to be blue circles. The `hold on` command allows to display multiple items within one figure.

Questions

- Obtain a line to classify the data by using what you know about the distributions of the data. In which sense is it optimal?

1.2 Support vector machine classifier

Go to `https://cs.stanford.edu/people/karpathy/svmjs/demo/`. On this website, you find an online application which lets you experiment with support vector machine (SVM) classification. Create a dataset by adding more data points to the existing dataset (unfortunately, you can not remove these 'starting points'). Make sure to have at least 10 data points for each class. Try out both the linear and the RBF kernel for classification.

- For each kernel, answer the following questions:
  - What do you observe when you add more data points to the dataset - both on the right and on the wrong side of the hyperplane. How does it affect the classification hyperplane?

  - Try out different values of the regularization hyperparameter `C` and the kernel parameter `sigma`. What is the role of the parameters? How do these parameters affect the classification outcome?

- Compare classification using the linear kernel with classification using the RBF kernel. Which performs better? Why?

Questions

- What is a support vector? When does a particular datapoint become a support vector? When does the importance of the support vector change? Illustrate visually. Note that a support vector is indicated by a large circle with bold-lined circumference and that its importance is proportional to the size in the online application.
- What is the role of parameters `C` and `sigma`? What happens to the classification boundary if you change these parameters. Illustrate visually.
- What happens to the classification boundary when `sigma` is taken very large? Why?

1.3 Least-squares support vector machine classifier

We proceed with the least squares based variant of the support vector machine (LS-SVM), using the LS-SVMlab toolbox. More information on the use of the toolbox can be found in the introduction guide on Toledo. Before we start experimenting with the toolbox ourselves, let's first have a look at a demo by typing

```
>> democlass
```

In this exercise, we investigate the `iris` dataset (available as `iris.mat` on Toledo). The data points are two-dimensional and indicate the length and the width of the sepals of the iris flower. Since the data points are two-dimensional, we can visualize our resulting classifier. In addition, the classification problem is binary: two different types of iris plants. We start by loading the dataset:

```
>> load iris.mat
```

Four variables appear in the workspace: `Xtrain`, `Ytrain`, `Xtest` and `Ytest`. `X` variables are used to indicate the data points, while `Y` variables denote class labels.

1.3.1 Influence of hyperparameters and kernel parameters

The hyperparameter (regularization constant γ) and kernel parameters highly influence the classification model.

Questions

- Try out a polynomial kernel with `degree` $= 1, 2, 3, \ldots$ and `t` $= 1$ (fix `gam` $= 1$). Assess the performance on the test set. What happens when you change the degree of the polynomial kernel?
- Let's now focus on the RBF kernel with squared kernel bandwidth σ^2.

  - Try out a good range of different `sig2` values as kernel parameters (fix `gam` = 1). Assess the performance on the test set. What is a good range for `sig2`?
  - Fix a reasonable choice for the `sig2` parameter and compare the performance using a range of `gam`. What is a good range for `gam`?

- Compare your results with `SampleScript_iris.m`, which is available on Toledo.

1.3.2 Tuning parameters using validation

The intuition developed in the previous section is now used to motivate automated tuning algorithms. In general, automatic tuning algorithms further split up the training data into a *training* and a *validation* part. This can be done in multiple ways:

- **Random split**. One way of splitting up the training dataset into a training and a validation part is by randomly taking some data points for training. The remaining data points are consequently used for validation. This can be implemented as:

```
>> perf = rsplitvalidate({Xtrain, Ytrain, 'c', gam, sig2,
    'RBF_kernel'}, 0.80, 'misclass');
```

  where the parameter value `0.80` indicates that 80 percent of the training data is used for *training* (thus, 20 percent of the training data is used for *validation*).

- ***k*-fold crossvalidation**. Another validation method is k-fold crossvalidation. In general, k is often taken to be equal to 10. This can be implemented as:

```
>> perf = crossvalidate({Xtrain, Ytrain, 'c', gam, sig2,
    'RBF_kernel'}, 10, 'misclass');
```

  where the parameter value `10` indicates that 10 folds are used in the crossvalidation procedure.

- **Leave-one-out validation**. Leave-one-out validation is a special case of k-fold crossvalidation, where k is taken equal as the number of data points. This can be implemented as:

```
>> perf = leaveoneout({Xtrain, Ytrain, 'c', gam, sig2,
    'RBF_kernel'}, 'misclass');
```

Questions

- Compute the performance for a range of `gam` and `sig2` values (e.g., $\gamma, \sigma^2 = 10^{-3}$, $\ldots$, 10^3). Use the random split method, 10-fold crossvalidation and leave-one-out validation. Visualize the results of each method: do you observe differences? Interpret the results: which values of `gam` and `sig2` would you choose?
- Why should one prefer crossvalidation over simple validation (random split)? How to choose the value of k in k-fold crossvalidation?

1.3.3 Automatic parameter tuning

The tuning procedure is fully automized in the LS-SVMlab toolbox. Type:

```
>> [gam,sig2,cost] = tunelssvm({Xtrain, Ytrain, 'c', [], [],
    'RBF_kernel'}, 'algorithm', 'crossvalidatelssvm',{10, 'misclass'});
```

where `'algorithm'` can be chosen as `'simplex'` (Nelder-Mead method) or `'gridsearch'` (brute force gridsearch).

Questions

- Try out the different `'algorithm'`. What differences do you observe? Why do the obtained hyperparameters differ a lot in different runs? What about the `cost`? Computational speed? Explain the results.

1.3.4 Using ROC curves

In practice, an alternative way to judge a classifier is by using the Receiver Operating Characteristic (ROC) curve of a binary classifier (see `>> help roc`). The main performance measure derived from such a curve is the area under the curve: the better the classifier, the higher the area under the curve. A ROC plot can be generated as:

```
>> % Train the classification model.
>> [alpha, b] = trainlssvm({Xtrain, Ytrain, 'c', gam, sig2,
    'RBF_kernel'});

>> % Classification of the test data.
>> [Yest, Ylatent] = simlssvm({Xtrain, Ytrain, 'c', gam, sig2,
    'RBF_kernel'}, {alpha, b}, Xtest);

>> % Generating the ROC curve.
>> roc(Ylatent, Ytest);
```

Questions

- In practice, we compute the ROC curve on the test set, rather than on the training set. Why?
- Generate the ROC curve for the `iris.mat` dataset (use tuned `gam` and `sig2` values). Interpret the result.

1.3.5 Bayesian framework

Using the Bayesian framework, it is possible to get probability estimates. Use a tuned pair of parameters (`gam`, `sig2`). Now, the probabilities that a certain data point belongs to the positive class given the model is computed by:

```
>> bay_modoutClass({Xtrain, Ytrain, 'c', gam, sig2}, 'figure');
```

Questions

- How do you interpret the colors of the plot? Hint: activate the colorbar of the figure.

- Change the values of `gam` and `sig2`. Visualize and discuss the influence of the parameters on the figure.

2 Homework problems

Illustrate your skills on support vector machine learning of synthetic and real-life datasets:

1. the Ripley dataset (`ripley.mat`),
2. the Wisconsin Breast Cancer dataset (`breast.mat`) and
3. the Diabetes dataset (`diabetes.mat`).

All datasets are available on Toledo. More information regarding these datasets can be found on the UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/index.php`. For each of these datasets, answer the following questions related to LS-SVM classification.

Questions

- Visualize the data. Inspect the data structure: what seems to be important properties of the data? Which classification model do you think you need, based on the complexity of the data?
- Try out different models (linear, polynomial, RBF kernel) with tuned hyperparameter and kernel parameters. Compute the ROC curves. Which model performs best? Which model would you choose?
- Are you satisfied with the performance of your model? Would you advise another methodology?