COMP90051 Statistical Machine Learning

Semester 2, 2015

Workshop Week 8: Support Vector Machines

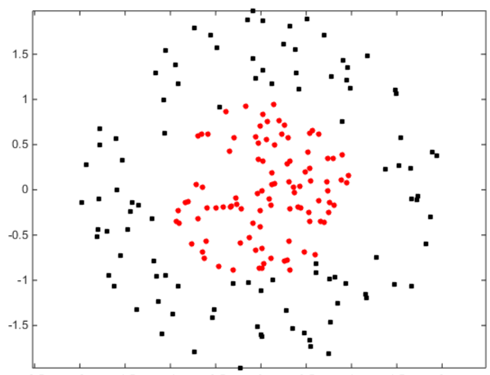
Introduction

In this workshop we will get practical experience in classifying data that is not linearly separable using support vector machines (SVM). Furthermore, we will practice in tuning SVM parameters using cross-validation. You can use any programming language, but MATLAB, R or Python are recommended, because for these languages we provide links to corresponding SVM tutorials (see Appendix below). The hints below are written in white font. Use text selection in order to be able to read them.

Step 1

Generate a two-dimensional dataset with two classes that looks something like shown in the figure. (Hint:

.)



Step 2

Learn how to train an SVM classifier and make predictions in your language of choice. If you are using MATLAB, R or Python you can refer to the corresponding SVM tutorial (see Appendix below). If you are using a different language, you will need to find a tutorial, and also make sure SVM implementation is readily available. Whichever tutorial you end up following, for the purposes of this workshop, you don't need to read the whole tutorial (but it can be beneficial for you to read all of it after the class).

In this step, the aim is to identify commands that correspond to the following pseudo-code:

```
trained_classifier = fit_svm(
      train_data, labels, type_of_kernel, regularisation_param_c)

predicted_class = predict(new_data, trained_classifier)
```

Clearly, in different programming languages this command will have different signature. Also, there will be many additional parameters that you will be able to provide. Those parameters can be left to their default values.

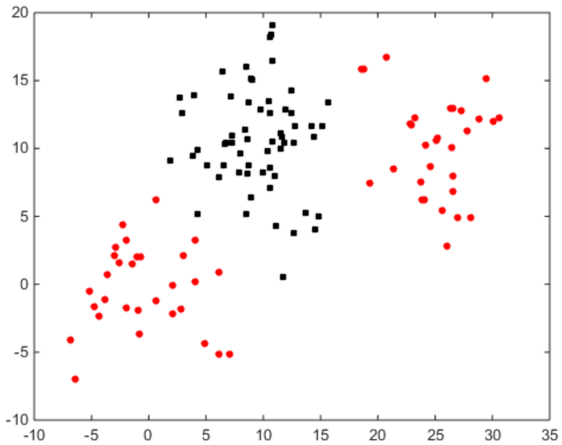
Step 3

Train an SVM classifier on data from Step 1. Use a linear kernel and parameter C set to 1. Plot the decision boundary. Now repeat the fit, and use a polynomial or RBF kernel. Plot the decision boundary for the new fit. What is the difference between the boundaries?

Step 4

Generate another dataset that looks like this. (Hint:

.)



Step 5

User RBF kernel to train an SVM classifier on the data from Step 4 using a range of values for regularisation parameter C (e.g., varying C from 0.01 to 10). Plot fitted decision boundary obtained for each value C. How does regularisation affect the decision boundary?

Step 6 (optional)

Use cross-validation to choose the best value for regularisation parameter C. Plot fitted decision boundary. How does this boundary compare with boundaries found during Step 5?

Step 7 (optional)

Repeat Step 5 using different kernel functions.

Appendix

For MATLAB, refer to “Train SVM Classifiers Using a Custom Kernel” section in the following tutorial:
http://au.mathworks.com/help/stats/support-vector-machines-svm.html

For R, refer to Section 9.6.1 in the following book:
http://www-bcf.usc.edu/~gareth/ISL/ISLR%20First%20Printing.pdf

For Python, refer to “Classification” section in the following tutorial:
http://scikit-learn.org/stable/modules/svm.html