7COM1033 Semester B 2020/21
Jan T Kim, j.t.kim@herts.ac.uk 12 April 2021

Neural Networks & Machine Learning: Worksheet 5

Aim

In this worksheet you will explore Support Vector Machines (SVMs). You will

- use LIBSVM through the R package e1071,
- download and install the LIBSVM package,
- explore SVM kernels and hyperparameters, and their effects on the separation structures in 2D input data spaces,
- gain experience in running the LIBSVM command line programs and identifying command line parameters based on mathematical or other formal specifications,
- develop skills in using the command line to reproducibly carry out LIBSVM analyses.

This worksheet is not marked and no submission is required. You are most welcome to post any questions, comments or results on Canvas.

Tasks

1. **Download and Install LIBSVM.** Visit the LIBSVM homepage at

   http://www.csie.ntu.edu.tw/~cjlin/libsvm/ .

   Read the introduction and get an overview of the other sections of the page. Then download the LIBSVM zip file in the "Download LIBSVM" section (the file is called libsvm-3.24.zip at the time of writing this sheet). As an alternative you can also download the file from the Canvas site. Unzip the archive and review its contents.

   Installation and use of LIBSVM depends on your operating system platform. The archive contains some executables (*.exe files) for Windows in the windows subdirectory. On other platforms you'll need to build the executables yourself. Alternatively, you may be able to find pre-built binaries for your platform, e.g. such packages are available for most major Linux distributions.

   The LIBSVM system is comprised of a set of programs that are designed to be run from the command line, which, depending on your operating system, may be known as "Command Prompt", "shell", or (somewhat imprecisely) as the "terminal" or "console".

   After installing LIBSVM, typing the command svm-train should result in displaying a usage message summarising various command line options provided by this program. Depending on your operating system and personal configuration, you may have to include a relative path in your command (e.g. ./svm-train is a quite typical example).

   The svm_toy.jar application, which you'll use in task 3 is written in Java, so you need a Java environment to run it. Suitable Java systems, along with installation instructions for all major operating systems, are available for download from https://adoptopenjdk.net/ or https://jdk.java.net/.

   Please use Canvas discussions for this worksheet for questions and for sharing useful information around building and installing LIBSVM, running LIBSVM programs, installing Java, and other technical matters.

2. **The LIBSVM Command Line Interface.** This task is designed to prepare you by focusing on determining command line elements based on your understanding of SVMs as covered in the lectures. You are not required to run any of the LIBSVM programs for this task.

   Read the documentation in the files README and study the "svm-train Usage" section in detail.

Develop a LIBSVM command line options that specify the polynomial kernel

$$k(\boldsymbol{x},\boldsymbol{y}) = (1 + \langle \boldsymbol{x},\boldsymbol{y} \rangle)^2.$$

Start by finding the polynomial kernel type and notice its kernel equation. Then match the symbolic names `gamma`, `degree` and `coef0` to the kernel equation above. This kernel is used in the XOR example in the lecture slides. Note down these command line options (ideally using a text editor on your computer).

3. **Explore SVMs Using the "SVM toy" Program.** The LIBSVM suite includes the `svm_toy.jar` program for exploring SVMs by interactively "playing" with data in $\mathbb{R}^2$ and up to three classes. You can build this file from the Java sources in the LIBSVM zip archive, or alternatively download it from Canvas.

   You can run this program using a command line like

   ```
   java -jar svm_toy.jar
   ```

   or, depending on your operating system, by double-clicking the file in a file browser. The `svm_toy` application is also embedded in the LIBSVM home page, and you can use that for this task if you have trouble running it on your system. The application presents you with a graphical user interface (GUI). The main portion of the GUI shows the 2-dimensional data space in which you can place data points by clicking. The "Change" button changes the class label of the points you place, indicated by colour. Use this facility to place points that are not linearly separable as shown in the XOR example in the lecture slides.

   Replace the default command line options with those you previously determined and click the "Run" button. This should result in the data space getting coloured according to the class predicted for the respective coordinates in the data space.

   Do you get a separation that corresponds to XOR, i.e. with one colour in the lower left and upper right area, and the other colour in the upper left and lower right area? If not, consider which effect you would expect the C regularisation parameter to have on the outcome. Would setting a larger C increase or decrease the chance of getting the desired separation?

   After deciding your approach, consult the `README` documentation and set the C parameter accordingly. If your approach is to increase it, start with a moderate value and double that repeatedly until you achieve the desired effect. Take care to leave the parameters to specify the quadratic polynomial kernel unchanged.

   Once you have obtained an XOR separation, try variations of placing the four points. You can use the "Clear" button to erase the points you have placed and start over. Which effect does placing the points closer together have on the C setting required to achieve XOR separation? Can you explain your finding based on the principles of SVMs?

4. **Using "SVM toy" With Data from Files.** For this exercise you need the files `data-1.txt`, `data-2.txt` and `data-3.txt` which are available on Canvas. You also do need to run the `svm_toy` application on your system, as the version embedded in the LIBSVM home page does not support loading data from local files.

   For each of the tasks below, keep notes of command line parameter settings that give good or interesting results.

   (a) Load `data-1.txt` by clicking the "Load" button in the window displayed by the "SVM toy" application. Notice that the data points are easily linearly separable. Set the command line parameters to specify a linear kernel with $C = 100$, and train a SVM by clicking the "Run" button.

   (b) Load `data-2.txt` and notice that there is one green item close to the turquoise ones in the upper left part of the data space. Find a parameter setting that results in this point to be classified correctly, and one in which this point is not classified correctly but there is a wide margin around the border separating the areas of classification. You should use the simplest possible kernel to achieve this.

   (c) Move on to the `data-3.txt` dataset and notice that this is not linearly separable. Try to find a SVM that correctly separates the data and, at the same time, requires a small number of support vectors.

5. **Run SVMs from the Command Line.**

(a) Find the command line options you developed with the "SVM Toy" application in your notes, and enter the command svm-train, followed by these options, the file name (i.e. data-1.txt), and a model file name, e.g. data-1-model.txt. You should see the message "optimization finished", followed by a few line of statistics about the optimisation run. After svm-train completes, run the dir command to verify that your model file has been created.

LIBSVM model files are plain text files, so you can inspect them with a text editor. You are also encouraged to inspect the data files in the same way.

Use the svm-train program to create models based on data-2.txt and data-3.txt, using the command line parameters you developed using the "SVM Toy" program.

(b) Review the usage instructions for the svm-predict program to predict outputs from the data-1.txt. Notice the message message indicating accuracy.

6. **Using LIBSVM in R.** The R package e1071 provides an R interface to LIBSVM. Find this package on the Comprehensive R Archive Network (CRAN) and install it in your R system and review the documentation of the svm function for constructing SVM classifiers, and the predict.svm function for predicting class labels.

The files data-1.csv, data-2.csv and data-3.csv contain the same data as the corresponding *.txt files, in a CSV format that is more convenient for use with R. The class labels are in the column named cls. Use these files to replicate your results with the LIBSVM executables (svm-train, svm-predict).

The models created by the svm function provide convenient access to various properties, including tables containing the support vectors. Use these facilities to find out how your choices of kernels and other hyperparameters affect the number of support vectors.

There are also some further CSV files for you to include in your work. In addition to using these to construct SVMs, you should also get an impression of their contents e.g. by checking out some basic plots.

7. **Further LIBSVM exploration.** Check out "A Practical Guide to Support Vector Classification" by the LIBSVM authors Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, which is available at

https://www.csie.ntu.edu.tw/~cjlin/papers/guide/

You will also find data files at this site. The exercises on this sheet should help to get you started working with these, to apply the methods discussed in the guide, and to perhaps reproduce some of the analyses shown in the guide as well.