Human-Oriented Robotics
Prof. Dr. Kai Arras, Social Robotics Lab
Lab instructors: Timm Linder, Luigi Palmieri, Billy Okal

Winter term 2014/2015
University of Freiburg
Department of Computer Science

Submission: Send your solution via email to `palmieri@informatik.uni-freiburg.de` until December 2, 2014 with subject "`[exercises] Sheet 5`". All files (Matlab scripts, exported figures, hand-written notes in pdf/jpg format) should be compressed into a single zip file named `lastname_sheet5.zip`.

Exercise 5: Support Vector Machines

For this exercise, you will need to download a dataset files and starter code from the course website. This exercise continues on supervised learning with focus on using SVMs.

Exercise 5.1: Classification with SVMs We will use the Matlab built-in implementation of SVMs (`svmtrain, svnclassify`). This requires the statistics toolbox which is included in the 'Typical' install. For each of the provided datasets (`simple.txt` and `complex.txt`), perform the following:

a) Load the dataset into Matlab using appropriate commands. Each dataset contains an $N \times 3$ matrix of space-separated numbers where each row corresponds to a training pair $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^2$ are samples of data and $y_i \in \{+1, -1\}$ are the labels of the two classes.

b) Plot the dataset using the `scatter` command to visually inspect how the samples can be separated. Annotate the plot accordingly (add axes labels, title, etc.) and save the result as a `JPEG` image.

c) Divide the dataset into a training set and a test set. The training set should contain 2/3 of the data, and the test set 1/3. Randomly draw samples from the original data set for this purpose. **Hint:** use `randperm`.

d) Train a SVM model using the `classifysvm` function provided with the datasets. This function uses Matlab's `svmtraim, svmclassify` functions. You are encouraged to look at their respective documentation to be familiar with their options. The function returns the Lagrange multipliers, support vectors and bias in a struct as well as different classification measures. Implement the dual version of SVM inference given by Equation 1 for new data points $\mathbf{x}'$ on a grid over the range $x_1 = [0..1]$ and $x_2 = [0..1]$.

$$y' = \sum_{i=1}^{N} \lambda_i y_i k(\mathbf{x}_i, \mathbf{x}') + b \qquad (1)$$

e) Store all corresponding y'-values and plot the contours of this function for values $y = -3, y = 1, y = 0, y = -1, y = 3$ on top of the data points using the command `contour`. Use `clabel` to label the contours and save the result as a `JPEG` image. What is the meaning of those contours?

f) Using different kernels (linear, rbf, polynomial) repeat steps (d) and (e). You can also use the `surf` or `surfc` for 3D surface plots.

g) Create a table comparing the different kernels on the following measures (See Exercise 4.3 for the definitions); Discuss the performance of the various kernels.

- **false positive** (fp)
- **true positive** (tp)
- **false negative** (fn)
- **true negative** (tn)

h) Vary the values of stiffness parameter C, and the kernel parameters σ and p and repeat steps (d), (e) and (f). Discuss the resulting decision boundaries, number of support vectors and performance measures as a function of different parameter values. Find one example of severe overfitting, discuss it and save the corresponding image in `JPEG` format.