Machine Learning 2
Data Science
Winter Semester 2018/19 **Prof. Tim Downie**

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN University of Applied Sciences

Workshop 4
Support vector machines

In this workshop you will need the packages `e1071`, `ISLR`, `rpart`, `ROCR` and `MASS`.
Note that the code for all the labs in James are available to download from:
`http://www-bcf.usc.edu/~gareth/ISL/code.html`.

Exercise 1 Introduction to support vector machines

If you have not yet worked through Exercise 3 in Workshop 3 on linear SVMs, work through this now: Section 9.6.1 in James on page 330.

Exercise 2 Non-linear SVMs: using different kernels

Section 9.6.1 looks at non linear boundaries. When you get to the `svm` command using the radial kernel function, fit SVM models with the following kernels:

- Linear kernel

  ```
  svmfit=svm(y~., data=dat[train,], kernel="linear",   cost=1)
  plot(svmfit, dat[train,])
  ```

  No hyperplane is found, and all the red points are misclassified. Varying the cost does not help.

- Polynomial kernel with degree 2

  ```
  svmfit=svm(y~., data=dat[train,], kernel="polynomial",  degree=2,
     gamma=1, cost=0.1)
  plot(svmfit, dat[train,])
  ```

  With a quadratic polynomial kernel two distinct borders are possible. Try increasing the cost using a few values between 1 and 10. The boundary now becomes an ellipse.

  Note that the parameter cost is the reverse of the parameter C in the lecture notes. A high value for `cost` penalises heavily each support vector. In the notes the parameter was an allowance and a high value allowed more support vectors.

- Try increasing the degree. With $d = 3$ no sensible boundary is found. With $d = 4$ there is a reasonable fit but with an unrealistic boundary.

Using a radial kernel the boundary can become more irregular.

Continue to work through James until the end of section 9.6.4.

Exercise 3 Using SVMs on a practical data set

In Section 9.6.5 you will analyse the Khan *gene expression* data (in the ISLR package). These data have few observations (63 in the training set) but many variables measurements of how much 2308 genes are expressed. Data sets where $p \gg n$ are often difficult or impossible to fit with classical statistical modelling methods.

When you have finished section 9.6.5 fit an `rpart` classification tree to the Khan data. Obtain a classification matrix for the training and for the test data. Compare the results with the SVM result in the last section.

Hint: the syntax for `predict` for an `rpart` object is a liitle different as for an `svm` object:

```
predict(Khan.tree,newdata=dat.te,type="class")
```