N. Bertschinger
M. Kaschube
V. Ramesh

Machine Learning I
Exercise Sheet 8
Due on Mon, January 9, 2017, 12:15

Turn in your solutions via email, please send to:
kamyshanska@fias.uni-frankfurt.de
Include [ML1-1617] in the subject line, include your name in the file name.

Exercise 1. Competition: *In this exercise you can win 20 points plus some extra points by scoring high in a small regression competition. Here are the instructions:*

- *Download the training data from the course web site (these are text files):*
  - comp_trainX.dat*: 5-dimensional input, 250 samples*
  - comp_trainY.dat*: target output, 250 samples*

  Your program must be executable *without arguments and do the following:*
  - *Read test inputs from file comp_testX.dat (same format as training data, i.e. 5 numeric values per line) located in the working directory, i.e. where your program is run*
  - *Write predictions to file comp_testY.dat (same format as training data, i.e. one numeric value per line) in the working directory*

  Your program is allowed to use additional files, e.g. containing the weights of your model ... just hand them in along with your program. Your executable can assume that the files can be found in the working directory.
- Scoring*:*
  - *You get* **5 points** *for participation. Your program should run without error though.*
  - *The performance of your program is evaluated on testing data — which are different from your training data — in terms of its R^2:*

    $$R^2 = 1 - \frac{\sum_n (t_n - y_n)^2}{\sum_n (t_n - \mu)^2}$$

    where t_n denotes the target output for the n-th testing sample and y_n is your prediction. $\mu = \frac{1}{N} \sum_n t_n$ is the mean of the target outputs.

    Thus, R^2 normalizes the mean-squared error:

- ∗ $R^2 = 1$: *Perfect prediction*
- ∗ $R^2 \leq 0$: *Prediction is not better than predicting* $t_n \equiv \mu$

Here, you get $1\frac{1}{2}$ *points per 5% (rounded up) of* R^2*, e.g. for an* R^2 *of 42% you would get* $13\frac{1}{2}$ *points.*

Hints*:*

- *Start with a linear model using 5 basis functions just copying the 5 input dimensions, i.e.* $\phi_i(x_1, \ldots, x_5) = x_i$.
- *Try to add other basis functions, e.g. polynomial, sigmoid* $\phi(\boldsymbol{x}) = \tanh(\alpha \boldsymbol{x} - \beta)$ *or exponential kernel* $\phi(x) = e^{-\frac{1}{2}\frac{(\boldsymbol{x}-\boldsymbol{\mu})^2}{\sigma^2}}$.
- *Adjust all hyperparameters, i.e. regularization parameters and all additional parameters that you have introduced in your basis functions (e.g.* α, β *above).*

  Depending on your liking, use cross-validation or the Bayesian evidence for this purpose.

If you have any further questions, don't hesitate to contact me.
Good luck ☺