Homework I: Gaussians and linear regression

Solutions to this exercise sheet are to be handed in before the tutorial session on Friday, 26.10.12. Code (.m-file) and output (.mat-file, figures as .pdf-files) for the matlab-questions should be submitted by email to `nicolas.ludolph@student.uni-tuebingen.de`. Use comments to explain your code. All code and plots should also be handed in as hardcopy.

1. (15 points) **Mathematical preliminaries: Manipulating Gaussian densities** Consider the function $f(x) = \frac{1}{Z}\exp(ax^2 + bx + c)$, with $a < 0$, and the Gaussian density $g(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$
   (a) Find the maximum of $\log(f(x))$ as well its curvature (i.e. second derivative) at the maximum.
   (b) Show that, for appropriate choices of a, b, c, and Z, $f(x)$ is equivalent to $g(x)$. [Hint: Multiply out the square term in the exponent of $g(x)$, and equate the coefficients of $f(x)$ and $g(x)$.] Express μ and σ in terms of a, b, and c.

2. (25 points) **Linear regression** [**MATLAB**]. Suppose you are given training-inputs $D = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$, where each $\mathbf{x}_i$ is an $M \times 1$-dimensional vector, and outputs $\{y_1, \ldots, y_N\}$ You want to fit a function of the form $f(\mathbf{x}, \omega) = \omega^\top \mathbf{x} = \sum_{i=1}^{M} w_i x_i$ to this data by minimising the cost function

$$L(\omega, D) = \sum_{n=1}^{N} (f(\mathbf{x_n}, \omega) - y_n)^2 + \lambda\left(\omega^\top \omega\right)$$

   (a) By taking the derivatives of L with respect to each ω_i and setting them to 0, show that this cost function is minimised for $\omega_\lambda = \left(\sum_n \mathbf{x_n}\mathbf{x_n}^\top + \lambda \mathbf{I_M}\right)^{-1}\left(\sum_n \mathbf{x}_n y_n\right)$, where $\mathbf{I}$ is an identify matrix of size M.
   (b) Load the variables in the file `Homework1.mat` and fit the parameters to the data `xTrain` and `yTrain` (where each row of `xTrain` corresponds to one data-point), using $\lambda = 0$, i.e. no regularisation. Plot the vector ω_o that you obtain.
   (c) Train multiple versions of your model on the training-set, using values $\lambda = 1, 5, 10, 25, 50, 75, 100, 250, 500, 750, 1000$. Make a plot that shows how the least-squares errors of your model on the training and the test set changes as a function of λ. [Note: Please make clearly labelled plots with meaningful axes.]
   (d) Using this plot, decide which value of λ you expect to give you the best generalization performance, and report the value. Use this model to predict the y-values for the data `xValidation`, save your predictions as a variable named `yValidationPredicted` in a file `HomeWork1YourName.mat` and submit this file.
   (e) Your friendly instructors included a column of 'ones' for the input data (i.e. the first entry of each $\mathbf{x}$ is 1). Explain why (for this model), including a constant term in the input-data can be useful and results in a more flexibel regression model.
   (f) [optional] By inspecting the vector ω obtained for the best model, make a guess as to which of the dimensions of x are important for predicting y, and which are irrelevant.