# FMAN-45 Machine Learning, Fall 2016
# Assignment 1

*Solve the problems and write down the solutions. If the assignment involves programming, download the code we provide and do the additional, required programming. Write a detailed report. All solutions, plots and figures should be in* one *pdf. It should be possible to understand all material presented in the report without running any code.* Submit your solutions and code using your individual Moodle account *as two files (a pdf and a single archive with all the code) at* `http://moodle.maths.lth.se/course/` *by the deadline (note that there will be no extensions).*

## 1 Probabilities (50 points)

Suppose that we have three coloured boxes r (red), b (blue), and g (green). Box r contains 3 apples, 4 oranges, and 3 limes, box b contains 1 apple, 1 orange, and 0 limes, and box g contains 3 apples, 3 oranges, and 4 limes.

a) If a box is chosen at random with probabilities p(r) = 0.2, p(b) = 0.2, p(g) = 0.6, and a piece of fruit is selected from the box (with equal probability of selecting any of the items in the box), then what is the probability of selecting an apple? (25 points)

b) If we observe that the selected fruit is in fact an orange, what is the probability that it came from the green box? (25 points)

## 2 Non-linear Change of Variable in Probability Densities (50 points)

Consider a probability density $p_x(x)$ defined over a continuous variable $x \in \mathbb{R}$, and suppose that we make a nonlinear change of variable using $x = g(y)$, so that the density transforms according to:

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right| = p_x(g(y))|g'(y)| \tag{1}$$

By differentiating the above, show that the location $\hat{y}$ of the maximum of the density in $y$ is not in general related to the location $\hat{x}$ of the maximum of the density over $x$ by the simple functional relation $\hat{x} = g(\hat{y})$ as a consequence of the Jacobian factor. This shows that the maximum of a probability density (in contrast to a simple function) is dependent on the choice of variable. Verify that, in the case of a linear transformation, the location of the maximum transforms in the same way as the variable itself.

## 3 Regression (100 points)

In this problem, we explore the behavior of polynomial regression methods when only a small amount of training data is available. We use polynomial regression models of the form

$$y = w_0 + w_1 x + w_2 x^2 + \dots w_m x^m + \epsilon = \mathbf{x}^\top \mathbf{w} + \epsilon \tag{2}$$

where $\epsilon \sim N(0, \sigma^2)$ (zero mean Gaussian noise) and $\mathbf{x} = [1\, x\, x^2 \dots x^m]^\top$. In a matrix form for all the training outputs, the model can be written as:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e} \tag{3}$$

where $\mathbf{y} = [y_1, \dots, y_n]^\top, \mathbf{X} = [\mathbf{x}_1^\top; \dots; \mathbf{x}_n^\top]$ depends on the polynomial order, and $\mathbf{e} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. In other words, the outputs $\mathbf{y}$ are normally distributed with mean vector $\mathbf{Xw}$ and covariance matrix $\sigma^2\mathbf{I}$. The likelihood of the outputs, given the inputs, can therefore be expressed as

$$p(\mathbf{y}|\mathbf{Xw}, \sigma^2) = N(\mathbf{y}; \mathbf{Xw}, \sigma^2\mathbf{I}) \tag{4}$$

where $N(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multi-variate (here nvariate) Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

We will begin by using a maximum likelihood estimation criterion for the parameters w that reduces to least squares fitting.

1. Consider a 1D regression problem. The data in `housing.data` provides information of how 13 different factors affect house price in a residential area. (Each column of data represents a different factor, and is described in brief in the file housing.names.) To simplify matters (and make the problem easier to visualise), we consider predicting the house price (the 14th column) from the LSTAT feature (the 13th column). We split the data set into two parts (in `testLinear.m`), train on the first part and test on the second. You are provided with the necessary MATLAB code for training and testing a polynomial regression model. Simply edit the script (`ps1_part2.m`) to generate the variations discussed below.

   (a) (10 points) Use `ps1_part2.m` to calculate and plot training and test errors for polynomial regression models as a function of the polynomial order (from 1 to 7). Use 250 training examples (set `numtrain=250`).

   (b) (15 points) Briefly explain the qualitative behavior of the errors. Which of the regression models are over-fitting to the data? Provide a brief justification.

   (c) (15 points) Rerun `ps1_part2.m` with only 50 training examples (set `numtrain=50`). Briefly explain key differences between the resulting plot and the one from part a). Which of the models are over-fitting this time?

   There are many ways of trying to avoid over-fitting. One way is to use a maximum a posteriori (MAP) estimation criterion rather than maximum likelihood. MAP criterion allows us to penalize parameter choices that we would not expect to lead to good generalization. For example, very large parameter values in linear regression make predictions very sensitive to slight variations in the inputs. We can express a preference against such large parameter values by assigning a prior distribution over the parameters such as simple Gaussian

   $$p(\mathbf{w}; \alpha^2) = N(\mathbf{0}, \alpha^2\mathbf{I}) \tag{5}$$

   This prior decreases rapidly as the parameters deviate from zero. The single variance (hyper-parameter) $\alpha^2$ controls the extent to which we penalize large parameter values. This prior needs to be combined with the likelihood to get the MAP criterion. The MAP parameter estimate maximizes

$$\log(p(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2)p(w; \alpha^2)) = \log p(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2) + \log p(\mathbf{w}; \alpha^2) \tag{6}$$

The resulting parameter estimates are biased towards zero due to the prior. We can find these estimates as before by setting the derivatives to zero.

2. (15 points) Show that[1]

$$\mathbf{w}_{MAP} = \left(\mathbf{X}^\top \mathbf{X} + \frac{\sigma^2}{\alpha^2}\mathbf{I}\right)^{-1} \mathbf{X}^\top \mathbf{y} \tag{7}$$

3. (10 points). In the above solution, show that in the limit of infinitely large $\alpha$, the MAP estimate is equal to the ML estimate, and explain why this happens

4. Let us see how the MAP estimate changes our solution in the housing-price estimation problem. The MATLAB code you used above actually contains a variable corresponding to the variance ratio `var_ratio` $= \frac{\sigma^2}{\alpha^2}$ for the MAP estimator. This has been set to a default value of zero to simulate the ML estimator discussed in class. In this part, you should vary this value from 1e-8 to 1e-4 in multiples of 10 (i.e. 1e-8, 1e-7, ..., 1e-4). A larger ratio corresponds to a stronger prior (smaller values of $\alpha^2$ constrain the parameters $\mathbf{w}$ to lie closer to origin).

   (35 points) Plot the training and test errors as a function of the polynomial order using the above 5 MAP estimators and 250 and 50 training points. Describe how the prior affects the estimation results.

[1] See the Matrix Cookbook `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf` for an introduction to matrix calculus.