Machine Learning, Spring 2018: Exercise Sheet 9

Problem 1. Consider the polynomial curve fitting example from Section 7.2.1 in the LNs. In the LNs I say that polynomial curve fitting can be done by calling a ready-made function *polyfit* that Matlab offers you. In fact, *polyfit* is just a special case of linear regression. Find out for yourself how *polyfit* can be programmed (without peeking at the Matlab documentation). More specifically, consider the following optimization problem. Given: a sample $(x_i, y_i)_{i = 1, \ldots, N}$ of argument-value training points, where x_i, y_i are real numbers. Also given: a maximal polynomial degree m. Design an algorithm that finds the optimal weight vector $W = (w_0, \ldots, w_m)'$ defined by the loss function

$$L(W) = \frac{1}{N} \sum_{i=1}^{N} (p(x_i) - y_i)^2$$

All you have to do is to find a way to translate this problem into an instance of the linear regression problem specified in Equation (17) in the LNs.

Problem 2. Let $\gamma[\mu, \sigma^2]$ denote the pdf of a normal distribution with expectation μ and variance σ^2. A real-valued random variable X has a *mixture of Gaussians* (MoG) pdf $p(x) = \sum_{i = 1, \ldots, k} \alpha_i \gamma[\mu_i, \sigma_i^2] (x)$, where the k *mixture coefficients* α_i are non-negative and sum to 1. What is $E[X]$? – *Note*: Mixture of Gaussians representations of distributions are very popular in ML because they combine simplicity with flexibility. The graphics in Figure 23 in the LN actually show a MoG example.

Problem 3. Given a size-N training sample $(x_i, y_i)_{i = 1, \ldots, N}$ of argument-value training points, setting up an L-fold cross validation scheme for model optimization requires one to decide on the number L of folds in the first place. The two extreme cases are $L = 2$ (smallest possible number of folds) and $L = N$ (largest possible number of folds, "leave-one-out cross-validation"). Following the routines of the cross-validation procedure, in each of these cases one will determine an optimal model capacity. Here is a claim:

"When one uses 2-fold cross-validation, the optimal model capacity found is likely to be too low, that is, one will settle on a model capacity $m_{opt\ 2\text{-}fold}$ which is smaller than the true optimal model capacity m_{opt} and hence will give underfitting models in the end; whereas if one uses leave-one-out cross-validation, one will determine a model capacity $m_{opt\ N\text{-}fold}$ which is greater than m_{opt}, hence ultimately leading to overfitting."

Is this claim true? What is, in effect, the best choice for L? Give your reasoning in plain English.