Practical Exercise B

In this assignment you will implement a Bayesian binary classifier. For this, you will apply the Laplace approximation to the Logistic Classification model from the coursework exercise, evaluating the results obtained using several metrics, such as the average test log-likelihood and the confusion matrices that were already described in the coursework exercise (see the attached notes on Bayesian classification and sections 4.4 and 4.5 from Bishop's book for a description of the Laplace approximation, note that there is a typo in Bishop's book, in equation (4.143), where S_N on the left-hand-side should be S_N^{-1}). You will also investigate the performance of selecting hyper-parameter values by optimizing the model evidence (the normalization constant in Bayes rule). Your answers should contain an explanation of what you do and where code is asked for you need only include the central commands (complete listings are unnecessary). You must also give an interpretation of what the numerical values and plots you provide mean. Why are the results the way they are?

a) Describe how to apply the Laplace approximation to the Logistic Classification model from the coursework exercise. Assume a Gaussian prior over the model parameters with zero mean and variance σ_0^2: $p(\beta_m) = \frac{1}{Z}\exp(-\frac{1}{2\sigma_0^2}\beta_m^2)$ for $m = 0 \ldots N$. To approximate the predictive distribution, use the method described in section 4.5.2 of Bishop's book. Describe the approximation of the model evidence (normalization constant in Bayes rule) given by the Laplace approximation.

b) Write python code for computing the Laplace approximation. To obtain the MAP solution, use the python function `scipy.optimize.fmin_l_bfgs_b`, which performs gradient-based MINIMIZATION (not maximization). Compute the gradients as in the coursework exercise, but including now an additional term for the prior.

c) Run your code on the data from the coursework exercise after expanding the inputs with radial basis functions, as you already did in the lab, and fixing $\sigma_0^2 = 1$ and $l = 0.1$. Visualise the predictions by adding probability contours to the plots made in exercise (c) from the coursework assignment. Visualize also the predictions of the MAP solution obtained with `scipy.optimize.fmin_l_bfgs_b`. How do the results of the full Bayesian approach differ from those of the MAP solution?

d) Report the final training and test log-likelihoods per datapoint and the 2×2 confusion matrices for both the Laplace approximation and the MAP solution methods. Explain your findings.

e) Write code to tune σ_0^2 and l by optimizing the approximation of the model evidence given by the Laplace approximation. Use a grid search approach for the optimization process: create a grid of size 10×10 representing all possible combinations of 10 different values for σ_0^2 and 10 different values for l and then compute the approximation of the model evidence for each point in this grid. What 10 different points did you choose for σ_0^2 and l?

   A heat map plot is a graphical representation of data where the individual values contained in a matrix are represented as colors. Visualise a heat map plot with the approximation of the model evidence obtained for each value of σ_0^2 and l.

   What are the best values for σ_0^2 and l?

f) Report the visualisation of the predictions, the average training and test-loglikelihoods and the 2×2 confusion matrices obtained after tuning σ_0^2 and l. How do these results compare with the ones obtained with $\sigma_0^2 = 1$ and $l = 0.1$?