STOCKHOLM UNIVERSITY
Department of Statistics
Matias Quiroz

FINAL EXAM, MACHINE LEARNING ST5401
2022-11-30

Time for examination: 08.00-13.00

Allowed tools:
Course book (supplied as pdf): Lindholm, A., Wahlström, N., Lindsten, F. and Schön, T. *Machine Learning - A First Course for Engineers and Scientists*.
RStudio: The required packages are loaded in the supplied `R markdown` template `Template_exam_solutions.Rmd`. You may use any additional packages that are already installed.

Instructions: Submit your solutions as a html file generated by `R markdown`, including relevant code, figures, calculations and discussions. Detailed instructions on how to submit your exam are are found on `tenta.stat.su.se/tenta`. Paper sheets are handed out in case you prefer to provide handwritten solutions/discussions. It is recommended to submit your code as a separate markdown (.Rmd) (or .R file if you do not use markdown), to be used as backup. A markdown template is provided (`Template_exam_solutions.Rmd`), but feel free to use your own.

You are not allowed to use the internet during the exam. You are allowed to copy code from your own (possibly polished) computer lab solutions. Copying text from the textbook is strictly forbidden.

Exam format and grades: The exam consists of 4 problems worth a total of 40 marks. *The problems are not sorted by their degree of difficulty.*
Grades (out of 40): A: 36, B: 32, C: 28, D: 24, E: 20, Fx and F: < 20.

`Good luck!`

Problem 1. (10 marks)

Provide a well-motivated answer for each sub-question below. No need to carry out any computations or simulations on your computer. No need for lengthy discussions, I just want you to show that you understand your answer.

(a.) Are unbiased estimators always desirable? Discuss a situation in which you would prefer an unbiased estimator and another in which you would prefer a biased estimator.

(b.) Why is a decaying learning rate needed for a stochastic gradient descent algorithm to converge?

(c.) Describe the label switching phenomenon when estimating unsupervised Gaussian mixture models. What are the implications for the parameter estimates obtained via the EM-algorithm?

(d.) What is the purpose of bagging models (as opposed to just fitting a single model on the given dataset)?

(e.) Suppose that you fit an additive linear regression model

$$y = \beta_0 + \beta_1 x_1 + x_2\beta_2 + \varepsilon,$$

and that the true data generating process is

$$y = \beta_0 + \beta_1 x_1 + x_2\beta_2 + \beta_3 x_1 x_2 + \varepsilon.$$

Are you overfitting or underfitting the training data? Can a Gaussian process regression model mitigate this?

Problem 2. (10 marks)
Consider daily temperatures (in Celsius degrees) at some location in Japan over the course of a year (365 days). The output variable is the temperature y_i on day $i = 1, \ldots, 365$, and the covariate is

$$x_i = \frac{\text{number of days since the beginning of the year}}{365}.$$

The observations have been randomly split into a training set with 300 observations (`japan_temperatures_training.RData`) and a test set with 65 observations (`japan_temperatures_test.RData`). Note that the data are not considered as a time series, hence the random splitting.

(a.) Your task is to find the best fitting (see criteria below) polynomial (including an intercept), out of polynomials of order $k = 1, \ldots, 8$, fitted with a squared loss function and an L2 penalty (ridge regression). Use the function `cv.glmnet` to perform cross-validation to determine the optimal λ parameter for each polynomial. Consider only $\lambda \in [0, 1]$ by supplying `lambda=seq(0, 1, length.out=100)` as an argument to `cv.glmnet`. Choose the polynomial that minimises the root mean squared error (RMSE) on the test data.

(b.) Discuss the role of the parameter λ and its implication for the bias-variance tradeoff.

(c.) Fit the polynomial of choice from (a.) to the full training set (for example with `glmnet`) using the corresponding optimal λ found in (a.). What are the estimated polynomial coefficients?

(d.) Plot the fit of the polynomial in (c.) on a denser grid of scaled days (in the interval 0-1) together with the training and test data. From a visual inspection, is the model underfitting or overfitting the training data, or is it just about right? If any of the first two, what is the problem with the model? Suggest possible improvements that may mitigate this problem (no need for an implementation).

Problem 3. (10 marks)
The dataset `spam_training.RData` consists of $n = 3000$ spam ($y = 1$) and ham (no spam, $y = 0$) emails with corresponding 15 features[1]. Most of the features are continuous real variables in the interval $[0, 100]$, with values corresponding to the percentage of occurrence of a specific word or character in the email. There are also a few features that capture the tendency to use many capital letters. To evaluate the classifiers, the dataset `spam_test.RData` contains $n = 1601$ spam and ham emails (with the same 15 features as the training data). You should standardise the features to have zero mean and unit variance. If `X` is a matrix of features, this is achieved by `X<-scale(X)` in `R`. Note that if the model requires an intercept, it cannot be included in `X` when calling the `scale` function; include it after the standardisation.

(a.) Fit a vanilla (no penalty) logistic regression (classifier) model to the spam data using a package of your choice. Include an intercept in the model. Compute the confusion matrix (for example using the `caret` package) for the test data with a threshold of 0.5. Compute the precision and recall of the classifier. Interpret the meaning of the precision and recall in the context of the spam filter.

(b.) Fit a classification tree to the spam data using the `tree` package using default settings. Compute the confusion matrix (for example using the `caret` package) for the test data with a threshold of 0.5. Compute the precision and recall of the classifier. According to these measures together with the accuracy, do you prefer this classifier or the one in (a.)?

(c.) You sold your preferred classifier to a customer who launches a complaint. The complaint is that more spam emails are reaching their inbox (i.e. are not classified as spam) than desirable. The customer asks you to improve the classifier in this aspect, however, they have no resources to pay for a new model and they cannot give you more training data. Suggest what can be done in this setting and discuss any potential drawbacks.

[1]Source: https://archive.ics.uci.edu/ml/datasets/spambase. The dataset used here includes a subset of the 57 original features.

Problem 4. (10 marks)
Consider again the spam dataset in Problem 3. Suppose that we model the data as a two layer dense neural network,

$$\begin{aligned} \boldsymbol{q}^{(1)} &= h\left(\boldsymbol{W}^{(1)}\mathbf{x} + \boldsymbol{b}^{(1)}\right) \\ \boldsymbol{q}^{(2)} &= h\left(\boldsymbol{W}^{(2)}\boldsymbol{q}^{(1)} + \boldsymbol{b}^{(2)}\right) \\ z &= \boldsymbol{W}^{(3)}\boldsymbol{q}^{(2)} + b^{(3)} \\ \Pr(y = 1|\mathbf{x}) &= g(z), \end{aligned}$$

with activation functions ReLU h and sigmoid g. Suppose further that the first layer has 12 hidden units and the second layer has 6 hidden units. The `keras` package is not available in the exam room, so no code is required to solve this problem.

(a.) What is the dimension of the input $\mathbf{x}$ and the output $\Pr(y = 1|\mathbf{x})$?

(b.) What is the number of parameters for each of the equations above ($\mathbf{x}, \boldsymbol{q}$ and z are not parameters)?

(c.) What are the dimensions of $\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}, \boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, b^{(3)}, \boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{W}^{(3)}$ and z?

(d.) Discuss the role of the number of hidden units and its implication for the bias-variance tradeoff.

(e.) Suppose we instead consider the softmax function for two classes in the output layer in place of the sigmoid function for one class. Does this change the number of parameters in the network? If so, update the dimensions of the quantities in (c.) and state the new total number of parameters in the network.