Machine Learning B

2024-2025

Home Assignment 6

Yevgeny Seldin and Amartya Sanyal
Department of Computer Science
University of Copenhagen

The deadline for this assignment is **3 June 2025, 17:00**. You must submit your *individual* solution electronically via the Absalon home page.

A solution consists of:

- A PDF file with detailed answers to the questions, which may include graphs and tables if needed. Do *not* include your full source code in the PDF file, only selected lines if you are asked to do so.
- A .zip file with all your solution source code with comments about the major steps involved in each question (see below). Source code must be submitted in the original file format, not as PDF. The programming language of the course is Python.
- IMPORTANT: Do NOT zip the PDF file, since zipped files cannot be opened in *SpeedGrader*. Zipped PDF submissions will not be graded.
- Your PDF report should be self-sufficient. I.e., it should be possible to grade it without opening the .zip file. We do not guarantee opening the .zip file when grading.
- Your code should be structured such that there is one main file (or one main file per question) that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions, classes, etc.
- Handwritten solutions will not be accepted. Please use the provided latex template to write your report.

1 Analysing AdaBoost (35 points) [Amartya]

Assume you are given a training set $T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subset \mathbb{R}^d \times \{-1, +1\}$ of labelled instances.

Question 1 (20 points)**.** Prove, by induction over b, that the weight updates

$$w_i^{(b+1)} = \frac{w_i^{(b)} \exp(-\alpha_b y_i h_b(\mathbf{x}_i))}{\sum_{j=1}^n w_j^{(b)} \exp(-\alpha_b y_j h_b(\mathbf{x}_j))}$$

introduced in Adaboost can be written as

$$w_i^{(b+1)} = \frac{\exp(-y_i f_b(\mathbf{x}_i))}{\sum_{j=1}^n \exp(-y_j f_b(\mathbf{x}_j))},$$

where h_b is the weak learner fitted in boosting round b, α_b the importance of h_b, and $f_b = \sum_{p \leq b} \alpha_p h_p$.

Question 2 (15 points)**.** Recall the algorithm for Adaboost we saw in class where we had a weak learner which satisfies the guarantee that $\varepsilon_b \leq \frac{1}{2} - \gamma$ for some $\gamma > 0$. In this question, you are told that your weak learner satisfies the guarantee $\varepsilon_b \leq \frac{1}{2} - \gamma$ with probability $1 - \delta'$ (where the probability is over the randomness of selecting the subsampled training set and the randomness of the weak learner) where $\delta' \in (0, 1)$.

Then, find $B* \in \mathbb{Z}_+, \delta* \in (0, 1)$ such that when Adaboost runs for B^* iterations with the above weak learner, the error of the hypothesis h output by Adaboost is zero i.e.

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(h(x_i) \neq y_i) = 0,$$

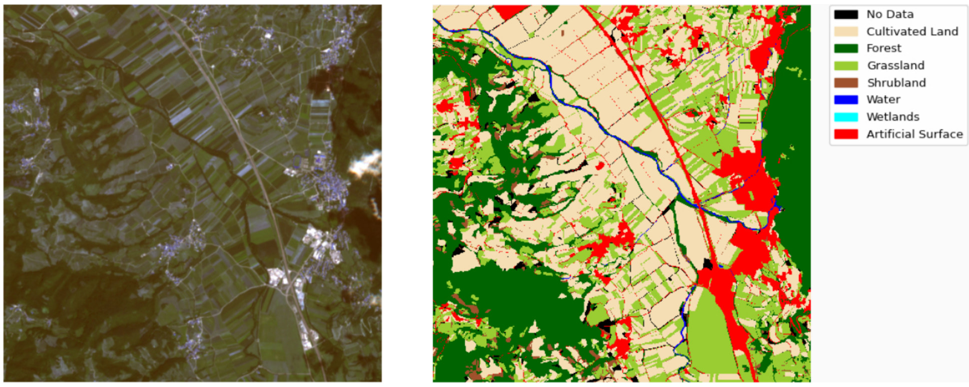
with probability at least $1 - \delta^*$.

Hint: $B*$ and δ^* can depend on γ and n.

2 Landcover Classification (35 points) [Amartya]

An important problem in remote sensing is landcover classification, i.e., the pixel-wise classification of satellite image data, see Figure 1. Often, multiple input images are given for a location, which are acquired over time. For instance, the popular Sentinel satellites[1] have a revisit time of 5 days, i.e., they take one image

[1] `https://sentinel.esa.int/web/sentinel/missions`



(a) Image Data

(b) Landcover Classification

Figure 1: The left image shows some input data. Here, only an RGB image is shown, but typically, more than three "bands" are available (e.g., RGB + near-infrared). The right image shows the classification of each pixel into one of the classes ("no data" corresponds to missing data). Such a plot is called landcover map.

every 5 days for each location that is monitored. Manually labelling billions of such pixels is too time-consuming and machine learning models are often used in this context to (automatically) obtain such landcover classifications.

In this exercise, you are supposed to apply AdaBoost to such a data set. As input data, you are given two files, `train.npz` and `test.npz`, containing some training and some test instances.[2] Each instance is composed of images of size 13 × 13 pixels from 6 "bands" (i.e., channels) and 12 timestamps, along with a class label corresponding to the (landcover) class corresponding to the central pixel. A simple way to obtain a decent landcover classification model is to consider these image data as simple feature vectors and to make use of tree ensemble models such as boosted trees (in fact, this has been the standard for many years before the deep learning wave). The Jupyter notebook `LandcoverClassification.ipynb` already contains some code to load the two datasets and to visualize the data. Extend this notebook and conduct the following steps:

1. Considering all the pixels as input features can quickly become time-consuming for AdaBoost. Select only the central pixels per image as features (the resulting feature vectors should consist of 72 values). Argue why this is a decent choice to accelerate the fitting process for the task at hand. Also argue why one might "loose" something by doing so.

2. Conduct 2-fold cross validation on the training data to select the best-

[2] You can download both files as well as a Jupyter notebook from: `https://sid.erda.dk/cgi-sid/ls.py?share_id=c9SgMJSGik`

performing AdaBoost model. As weak learners/models, consider simple decision tree classifiers with the Gini index as impurity measure. For AdaBoost, make use of the 'SAMME' discrete boosting approach that you know from the lecture. As grid, consider:

- *Depth of decision trees:* [1,2,3]
- *Number of boosted trees:* [50,100,200]

After having performed the grid search, select the best model, refit the model to the entire training data, and compute the test accuracy. Also generate a confusion matrix on the test data to visualize the errors made by the model. What is the accuracy of the model on the test set?

3. Optional: Can you improve the quality of the model by considering other features and/or other assignments for the parameters?

Hint: Make use of Scikit-Learn and the `GridSearchCV` class. You can provide a base estimator to the AdaBoost class. Here, figure out how to provide the tree depth as parameter to the grid search.

3 Majority Vote (15 points) [Yevgeny]

Solve Exercise 3.9 in Yevgeny's lecture notes.

4 Occam's kl-razor vs. PAC-Bayes-kl (15 points) [Yevgeny]

Solve Exercise 3.7 in Yevgeny's lecture notes.